

UNIVERSIDADE FEDERAL DO PARANÁ

ROBISON CRIS BRITO

UM MODELO DE OTIMIZAÇÃO PARA PLANEJAMENTO DINÂMICO DE VOO PARA  
GRUPOS DE DRONES POR MEIO DE SISTEMA MULTIAGENTE E LEILÕES  
RECURSIVOS

CURITIBA PR

2020

ROBISON CRIS BRITO

UM MODELO DE OTIMIZAÇÃO PARA PLANEJAMENTO DINÂMICO DE VOO PARA  
GRUPOS DE DRONES POR MEIO DE SISTEMA MULTIAGENTE E LEILÕES  
RECURSIVOS

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Eduardo Todt.

CURITIBA PR

2020

CATALOGAÇÃO NA FONTE – SIBI/UFPR

---

B862m

Brito, Robison Cris

Um modelo de otimização para planejamento dinâmico de voo para grupos de drones por meio de sistema multiagente e leilões recursivos [recurso eletrônico]/ Robison Cris Brito, 2020.

Tese (Doutorado) - Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná. Área de concentração: Ciência da Computação.

Orientador: Eduardo Todt.

1. Planejamento de rota. 2. Veículo aéreo não tripulado. I. Todt, Eduardo. II. Universidade Federal do Paraná. III. Título.

CDD 629.1

---

Bibliotecária: Vilma Machado CRB9/1563



## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **ROBISON CRIS BRITO** intitulada: **Um Modelo de Otimização para Planejamento Dinâmico de Voo para Grupos de Drones por Meio de Sistema Multiagente e Leilões Recursivos**, sob orientação do Prof. Dr. EDUARDO TODT, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 03 de Julho de 2020.

Assinatura Eletrônica  
04/07/2020 00:21:13.0  
EDUARDO TODT

Presidente da Banca Examinadora

Assinatura Eletrônica  
06/07/2020 15:40:15.0

JOSÉ RICARDO DA ROCHA CAMPOS

Avaliador Externo (UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ)

Assinatura Eletrônica  
03/07/2020 21:36:47.0  
ROBERTO PEREIRA

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica  
07/07/2020 20:46:36.0

FERNANDO SANTOS OSORIO

Avaliador Externo (UNIVERSIDADE DE SÃO PAULO)

## RESUMO

Este trabalho apresenta um modelo aplicado de cooperação para otimizar voos de veículos aéreos não tripulados do tipo quadricóptero, também conhecidos como Drones, com aplicação na agricultura de precisão. O modelo utiliza Sistema Multiagente para permitir a abertura, que é a propriedade de inserir e retirar elementos do modelo a qualquer momento. Para garantir a dinamicidade, que é a característica que o modelo tem de se recuperar de eventos adversos ou falhas, agentes cognitivos com BDI foram utilizados. Para garantir a troca de mensagens independente da quantidade de elementos no modelo, foi utilizado o protocolo FIPA Contract-NET. Um algoritmo distribuído de otimização utilizando leilões recursivos também foi desenvolvido, o qual visa otimizar o tempo de voo, assim como o uso da bateria dos Drones, sendo a bateria a grande limitação destes e inibindo sua utilização na agricultura de precisão. Esse algoritmo foi testado em seu modelo original e, posteriormente, refinado a partir de heurísticas e metodologias visando diminuir o número de leilões recursivos, assim como o tempo de processamento, em comparação ao modelo original. Este modelo, após aplicação das heurísticas e metodologias, foi testado. Em cenários contendo múltiplos Drones, o desempenho foi 30% superior ao algoritmo dinâmico encontrado na literatura que também pode ser aplicado em ambientes dinâmicos. Do ponto de vista de abertura e dinamicidade, o modelo foi testado no simulador MultiDrone Simulator, permitindo gerar novos planos de voo, mesmo com eventos adversos. Os resultados dos testes em simulação realizados sustentam que o modelo proposto apresenta comportamento como esperado, mostrando-se como uma plataforma promissora de pesquisa para uso de Drones em cenários da agricultura de precisão, uma vez que este modelo permite a utilização de múltiplos Drones em ambientes dinâmicos e abertos, garantindo a otimização do tempo de voo, o que garante economia da bateria dos Drones.

Palavras-chave: Drones, Sistema Multiagente, BDI, Leilão Recursivo

## ABSTRACT

This work presents an applied model of cooperation to optimize flights of unmanned aerial vehicles like quadcopters, also known as Drones, involved in precision agriculture. This model uses a Multiagent System to allow up the opening, which is the property of inserting and removing elements from the model at any time. To allow dynamism, which is the characteristic that the model has to recover from adverse events or failures, cognitive agents with BDI structure were used. To guarantee the exchange of messages in dynamic number of elements, the FIPA Contract-NET protocol were used. A distributed optimization algorithm using recursive auctions was also developed, which aims to optimize the number of points covered by Drones. This model aims to optimize the flight time, which directly reflects the optimization of the Drone's battery use. This is a great limitation of this kind of aerial vehicle and which inhibits its use in precision agriculture. This algorithm was tested as original proposed and, later, refined from heuristics and methodologies in order to decrease the number of auctions, as well as the processing time. This model, after applying the heuristics and methodologies, was tested, and in scenarios containing multiple Drones, the performance was 30 % higher than the dynamic algorithm found in the literature that can also be applied in dynamic environments. From the point of view of openness and dynamics, the model was tested in the MultiDrone Simulator, allowing to generate new flight plans, even with the simulated adverse events. The results of the simulation tests carried out maintain that the proposed model behaves as expected, showing itself as a promising research platform for the use of drones in precision agriculture scenarios, since this model allows the use of multiple Drones in environments dynamic and open, guaranteeing the flight optimization, which ensures battery saving for Drones.

Keywords: Drones, Multiagent System, BDI, Recursive Auctions

## LISTA DE FIGURAS

2.1	Visão geral de um modelo aplicado de otimização de voos para grupo de Drones.	22
2.2	Analogia do modelo baseado em leilões recursivos em relação a um leilão tradicional . . . . .	31
2.3	A esquerda, cenário com um Drone, uma base de recarga e quatro pontos de passagem. A direita, o grafo das possibilidades de plano de voo para o cenário .	33
3.1	Tipos de organizações de agentes (Hübner e Sichman, 2003) . . . . .	38
3.2	Componentes da arquitetura BDI (de Oliveira Zamberlam e Giraffa, 2001) . . . .	40
3.3	a) Classes de problemas existente, com os problemas P contidos em NP, que por sua vez estão contidos em P-Space. b) Gráfico da complexidade dos algoritmos, quanto mais próximo do eixo y, mais custoso é o processamento . . . . .	44
3.4	Exemplo de um circuito Hamiltoniano (Wikipedia, 2019b) . . . . .	45
4.1	Visão geral do modelo proposto com as ferramentas sugeridas para o desenvolvimento . . . . .	49
4.2	Protocolo de interação da FIPA Contract-Net (FIPA, 2002) . . . . .	51
4.3	Tela do MultiDrone Simulator. A esquerda, simulação de um voo com quatro Drones. A direita, a alocação de recursos em um gráfico de Gantt . . . . .	52
5.1	Contribuição do capítulo fundamentação teórica para definição das tecnologias e para o modelo proposto . . . . .	54
5.2	Definição dos grupos e papéis dos agentes no modelo e as restrições da comunicação . . . . .	55
5.3	Especificação Funcional para o Sistema Multiagente . . . . .	56
5.4	Estrutura dos agentes codificados em BDI, com as crenças em preto, desejos em verde e as intenções em azul . . . . .	57
5.5	Diagrama de sequência de leilões usando FIPA Contract-Net. . . . .	58
5.6	Agentes do modelo dispostos para o processo de otimização por meio do leilão recursivo - passo a passo . . . . .	59
5.7	Grafo com as possibilidades de deslocamento de um Drone considerando a posição inicial o vértice 11, $k = 2$ e autonomia do Drone igual a 5 . . . . .	60
6.1	Cenário com 4 pontos, com $k$ variando de 1 a 3 . . . . .	63
6.2	a) Cenário com 4 pontos. b) Cenário com 9 pontos. c) Cenário com 16 pontos .	63
6.3	Grafo com 120 possibilidades leilões, autonomia igual a 5, $k = 2$ , e todos os pontos sendo testados como inicial . . . . .	64
6.4	Primeiro resultado obtido para o cenário com 4 pontos (a), 9 pontos (b) e 16 pontos (c), em todos, a base está no Ponto 1 e o voo acontece no sentido horário .	65

6.5	Modelo de leilões representado em um grafo após aplicação de h1 . . . . .	65
6.6	Grafo de poda após a aplicação da heurística h2 . . . . .	66
6.7	Resultados após h2, observando que o cenário com 9 pontos não possui todos os pontos sobrevoados.. . . . .	67
6.8	Resultado para o cenário com 9 pontos. a) Utilizando $k = 2$ b) utilizando $k = 5$ .	67
6.9	Grafo de poda após a aplicação da heurística h3 . . . . .	68
6.10	Representação gráfica da quantidade ideal de vizinhos dada a posição de um ponto no plano cartesiano . . . . .	69
6.11	Proporção de pontos internos (maior quantidade de vizinhos) e pontos externos (menor quantidade de vizinhos) para diferentes formatos de área . . . . .	70
6.12	Fluxo de leilões realizados em um grafo onde a busca em profundidade acontece	70
6.13	Representação gráfica da metodologia de escolha do vizinho mais próximo para um cenário de 4x4 . . . . .	71
6.14	Comparação dos percursos obtidos para circuito Hamiltoniano (a) e não Hamiltoniano (b) . . . . .	71
6.15	Representação gráfica da metodologia da divisão de área de voo para um cenário de 8x8 com quatro Drones. . . . .	72
7.1	a) Resultado obtido pelo Leilão Recursivo no cenário com 64 Pontos. b) Resultado obtido pela Otimização Gulosa no cenário com 64 Pontos. . . . .	75
7.2	a) Resultado obtido pelo Leilão Recursivo no cenário com 121 Pontos. b) Resultado obtido pela Otimização Gulosa no cenário com 121 Pontos . . . . .	75
7.3	a) Resultado obtido pela otimização por leilões recursivos no cenário com 1.156 pontos e quatro Drones. b) Resultado obtido pela Otimização Gulosa no cenário com 1.156 pontos e quatro Drones . . . . .	77
7.4	Gráfico de Gantt para o cenário com 1156 pontos, 4 Drones, autonomia de 121 pontos e otimização realizada por meio de leilões recursivos. . . . .	77
7.5	Gráfico de Gantt para o cenário com 1156 pontos, 4 Drones, autonomia de 121 pontos e otimização realizada por meio de algoritmo guloso. . . . .	78
7.6	Telas do aplicativo Android para otimização de voo de grupo de Drones . . . . .	79
7.7	Tela para configuração de eventos externos ao ambiente, para teste de abertura e dinamicidade . . . . .	80
7.8	a) Plano de voo original com quatro Drones. b) Plano de voo após a retirada do Drone 4 (em amarelo) que aconteceu após o pouso do primeiro voo . . . . .	81
7.9	a) Plano de voo original com três Drones. b) Plano de voo atualizado com a inclusão de um quarto Drone no canto inferior direito. . . . .	81
7.10	Tela de configuração para mudança da direção/velocidade do vento. . . . .	82
7.11	a) Plano de voo original com quatro Drones. b) Plano de voo adaptado a nova direção/velocidade do vento. . . . .	83
7.12	Plano de voo gerado após o retorno precoce do Drone 1, em azul, após sobrevoar 13 pontos. . . . .	84



7.13	Grupo de publicações Operacional, apresentando uma visão geral do MAOVGD, já com as publicações específicas. . . . .	86
7.14	Visão geral das fases de pesquisa, já com os artigos do grupo Modelo de Otimização publicados/submetidos. . . . .	87

## LISTA DE TABELAS

2.1	Palavras-chave e sinônimos para a construção da String de busca.. . . . .	26
6.1	Resultados da execução do algoritmo de otimização com leilões recursivos para cenário com 4, 9 e 16 pontos . . . . .	64
6.2	Quantidade de leilões antes e após a aplicação da heurística h1, que define o ponto inicial para base de recarga . . . . .	65
6.3	Quantidade de leilões antes e após a aplicação da heurística h2, que não permite que um mesmo ponto seja visitado mais de uma vez. . . . .	66
6.4	Quantidade de leilões antes e após a aplicação da heurística h3, que possui um fator de término de processamento . . . . .	68
6.5	Quantidade de leilões após h3 com k=4 e, após a aplicação da metodologia de diferentes valores para k . . . . .	69
6.6	Quantidade de leilões antes e após a aplicação da metodologia de escolha do vizinho mais próximo. . . . .	71
6.7	Quantidade de leilões para circuito Hamiltoniano e não Hamiltoniano . . . . .	72
6.8	Quantidade de leilões antes e após a aplicação da metodologia de divisão de área de atuação por Drones . . . . .	72
7.1	Análise comparativa entre Leilões Recursivo x Otimização Gulosa para um cenário com 64 pontos . . . . .	74
7.2	Análise comparativa entre Leilões Recursivo x Otimização Gulosa para um cenário com 121 pontos. . . . .	76
7.3	Análise comparativa entre Leilões Recursivo x Otimização Gulosa para um cenário com 1 ponto e 4 Drones . . . . .	76
7.4	Características dos trabalhos que mais contribuíram para o modelo aplicado proposto. Na tabela, WP significa se permite ponto de passagem, FP se trata pontos proibidos, 3D se permite traçar rotas considerando a altitude de voo, Tipo de UAV qual o tipo de veículo aéreo foi utilizado nos testes, em Téc. Otim. qual o algoritmo de otimização foi utilizado, em abertura se o modelo permite a inclusão de novos elementos durante a sua execução, e por fim, dinamicidade se elementos como velocidade e direção do vento são considerados pelo modelo. . . . .	84

## LISTA DE ACRÔNIMOS

AG	Algoritmo Genético
ANAC	Agência Nacional de Aviação Civil
BDI	Belief, Desire and Intention
CARTAgO	Common ARTifact Infrastructure for AGents Open Environments
DLR	Deutsches Zentrum für Luft- und Raumfahrt
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
ED	Especificação Deontica
EE	Especificação Estrutural
EF	Especificação Funcional
EO	Especificação Organizacional
FP	Forbidding Point
FIPA - ACL	Found for Intelligent Physical Agents - Agent Communication Language
GRASP	Greedy Randomized Adaptive Search Procedure
ha	Hectare
IBGE	Instituto Brasileiro de Geografia e Estatística
IHC	Interface Homem Computador
INPE	Instituto Nacional de Pesquisas Espaciais
JAVA SE	Java Standard Edition
JSON	JavaScript Object Notation
KQML	Knowledge Query and Manipulation Language
LEACH	Low Energy Adaptive Clustering Heirarchy
MOISE+	Model of Organization for multi-agent SystEm
OACI	Organização da Aviação Civil Internacional
MAOVGD	Modelo Aplicado de Otimização de Voo para Grupo de Drones
MDE	Modelo Digital de Elevação
NASA	National Aeronautics and Space Administration
NIMA	National Imagery and Mapping Agency
PICO	Population Intervention Comparison Outcome
PLIM	Programação Linear Inteira Mista
PSO	Particle Swarm Optimization
QP	Questão de Pesquisa
RPAS	Remotely Piloted Aircraft Systems
RTK	Real Time Kinematic
RTX	Real-Time eXtended
SIR-C/X-SAR	Spaceborne Imaging Radar-C/X-Band Synthetic Aperture Radar

SRTM	Shuttle Radar Topography Mission
TSP	Travelling Salesperson Problem
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicles
WP	Way Point
XML	eXtensible Markup Language

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	CONSIDERAÇÕES INICIAIS	14
1.2	OBJETIVOS	16
1.2.1	Objetivo Geral.	16
1.2.2	Objetivos Específicos	16
1.3	JUSTIFICATIVA	17
1.4	ORGANIZAÇÃO DA TESE	18
<b>2</b>	<b>PROBLEMA DE PESQUISA.</b>	<b>20</b>
2.1	CONTEXTO DA PESQUISA	20
2.2	MAPEAMENTO SISTEMÁTICO SOBRE COOPERAÇÃO DE VEÍCULOS AUTÔNOMOS	24
2.3	TRABALHOS CORRELATOS.	28
2.4	MODELO APLICADO DE OTIMIZAÇÃO DE VOO PARA GRUPOS DE DRONES	30
2.5	APLICABILIDADE DO MODELO EM UM PROBLEMA DO CENÁRIO AGRÍCOLA	33
2.6	CONCLUSÃO	35
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA.</b>	<b>36</b>
3.1	SISTEMAS MULTIAGENTES.	36
3.1.1	Modelos Organizacionais de Agentes.	37
3.2	AGENTES DE SOFTWARE COGNITIVO	39
3.2.1	Linguagem de Programação BDI - AgentSpeak(L).	41
3.3	NEGOCIAÇÃO ENTRE AGENTES UTILIZANDO TÉCNICA DE LEILÃO	42
3.4	COMPLEXIDADE COMPUTACIONAL DE ALGORITMOS DE OTIMIZAÇÃO	43
3.5	CONCLUSÃO	46
<b>4</b>	<b>TECNOLOGIAS E MATERIAIS UTILIZADOS.</b>	<b>48</b>
4.1	MOISE+ PARA MODELAGEM DO SISTEMA MULTIAGENTE	48
4.1.1	Especificação Estrutural	48
4.1.2	Especificação Funcional	48
4.1.3	Especificação Deôntica	49
4.2	JASON PARA O DESENVOLVIMENTO DE AGENTES COGNITIVOS	49
4.3	FIPA CONTRACT-NET PARA NEGOCIAÇÃO POR LEILÕES	50
4.4	MULTIDRONE SIMULADOR PARA TESTES DA OTIMIZAÇÃO BASEADA EM LEILÕES RECURSIVOS	51
4.5	MATERIAIS UTILIZADOS	53

4.6	CONCLUSÃO . . . . .	53
<b>5</b>	<b>MODELO PROPOSTO . . . . .</b>	<b>54</b>
5.1	ORGANIZAÇÃO DOS AGENTES PARA O MODELO PROPOSTO . . . . .	54
5.2	PROPOSTA DE AGENTES COGNITIVOS . . . . .	56
5.3	FIPA CONTRACT-NET APLICADO AO MODELO DE OTIMIZAÇÃO DE VOOS DE DRONES . . . . .	57
5.4	LEILÕES RECURSIVOS . . . . .	58
5.5	CONCLUSÃO . . . . .	61
<b>6</b>	<b>HEURÍSTICAS E METODOLOGIAS APLICADAS AO ALGORITMO DE OTIMIZAÇÃO BASEADO EM LEILÕES RECURSIVOS . . . . .</b>	<b>62</b>
6.1	CENÁRIO E METODOLOGIA PARA A REALIZAÇÃO DOS TESTES . . . . .	62
6.2	OTIMIZAÇÃO DO LEILÃO RECURSIVO POR MEIO DE HEURÍSTICAS . . . . .	63
6.2.1	Testes com o Modelo Original . . . . .	64
6.2.2	Testes do Modelo com Heurística h1 - Definição do ponto inicial . . . . .	65
6.2.3	Testes do Modelo com Heurística h2 - um ponto não pode ser visitado duas vezes . . . . .	66
6.2.4	Testes do Modelo com Heurística h3 - Parâmetro de parada de processamento após o primeiro ótimo ser encontrado . . . . .	67
6.3	METODOLOGIA DE OTIMIZAÇÃO APLICADA A CENÁRIOS REAIS . . . . .	68
6.3.1	Utilização de Diferentes Valores de k em um Mesmo Cenário. . . . .	69
6.3.2	Identificação do Vizinho mais Próximo . . . . .	70
6.3.3	Definição da Áreas de Atuação . . . . .	71
6.4	CONCLUSÃO . . . . .	72
<b>7</b>	<b>RESULTADOS E DISCUSSÕES . . . . .</b>	<b>74</b>
7.1	TESTES DE OTIMIZAÇÃO DO MODELO APLICADO DE VOO DE GRUPO DE DRONES . . . . .	74
7.1.1	Testes em Cenário com 64 Pontos - Circuito Hamiltoniano . . . . .	74
7.1.2	Testes em Cenário com 121 pontos - Circuito Não Hamiltoniano . . . . .	75
7.1.3	Testes em Cenário com 1.156 Pontos - Quatro Drones . . . . .	76
7.2	APLICAÇÃO DO MODELO DE OTIMIZAÇÃO DE VOO DE GRUPO DE DRONES EM UM DISPOSITIVO MÓVEL . . . . .	78
7.3	AVALIAÇÃO DA ABERTURA E DINAMICIDADE DO MODELO. . . . .	79
7.3.1	Testes de Abertura do Modelo . . . . .	80
7.3.2	Testes de Dinamicidade do Modelo. . . . .	82
7.4	ANÁLISE QUALITATIVA DO MODELO APLICADO DE OTIMIZAÇÃO DE VOO DE GRUPO DE DRONE . . . . .	83
7.5	PUBLICAÇÕES . . . . .	85
7.6	CONCLUSÃO . . . . .	90

<b>8</b>	<b>CONCLUSÃO</b> . . . . .	<b>92</b>
8.1	CONSIDERAÇÕES FINAIS . . . . .	92
8.2	ENCAMINHAMENTOS . . . . .	93
	<b>REFERÊNCIAS</b> . . . . .	<b>95</b>
	<b>APÊNDICE A – ARTIGOS SUBMETIDOS/PUBLICADOS</b> . . . . .	<b>100</b>
A.1	ICOMP’17 - SOCIOTHECNICAL APPROACH. . . . .	101
A.2	LARS’17 - SYSTEMATIC MAPPING. . . . .	108
A.3	LARS’18 - PID VERSUS FUZZY DRONE CONTROL . . . . .	114
A.4	MEDITEC’18 - WEB SYSTEM TO FLIGHT MANAGEMENT . . . . .	120
A.5	COTB’19 - AUTONOMOUS LANDING/TAKING-OFF . . . . .	126
A.6	COMPSAC’19 - FIXED WINGS VERSUS ROTATIVE WINGS . . . . .	132
A.7	COMPSAC’19 - AREA DIVISION . . . . .	138
A.8	COMPSAC’19 - QUALIFICATION PAPER. . . . .	142
A.9	B-SIDES’20 - BEST K-NEIGHBOR PARAMETER CONSTANT . . . . .	148
A.10	ICICT’20 - MOBILE OPTIMIZATION ALGORITHM . . . . .	153
A.11	WPSN’20 - BASES (SUBMETIDO) COMMUNICATION . . . . .	158
A.12	CEA’20 - ALGORITHM EVALUATION (SUBMETIDO) . . . . .	164
A.13	SWARMNET’20 - GREEDY OPTIMIZATION (SUBMETIDO) . . . . .	173
A.14	WIDROID’20 - SIMULATOR (SUBMETIDO) . . . . .	180
A.15	B-SIDES’21 - NEAREST NEIGHBOR AND AREA DIVISION (A SUBMETER) . . . . .	185
A.16	WESAAC’21 - MULTIAGENT SYSTEM (SUBMETIDO) . . . . .	189
	<b>APÊNDICE B – ANEXOS</b> . . . . .	<b>202</b>
B.1	REGRAS DA ANAC PARA USO DE DRONES. . . . .	203

# 1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais do presente trabalho, os seus objetivos geral e específicos e a justificativa, por fim, a estrutura dos capítulos subsequentes.

## 1.1 CONSIDERAÇÕES INICIAIS

Drone, quadricóptero, Veículo Aéreo Não Tripulado (VANT), *Remotely Piloted Aircraft Systems* (RPAS) e *Unmanned Aerial Vehicle* (UAV) são algumas das variações de nomes para os veículos aéreos não tripulados controlados a distância e/ou autônomos. Os UAVs, para generalizar as diversas denominações atribuídas aos veículos aéreos não tripulados, foram inicialmente concebidos para fins bélicos, mas com a evolução e a acessibilidade de novas tecnologias suas aplicações vêm crescendo nas mais diversas áreas. Hoje existem dois tipos de UAVs: os modelos de asa fixas, que possuem aparência semelhante a um pequeno avião, e os modelos com asa rotativa, que são compostos por pares de hélices.

Para Bastianelli et al. (2012) existe uma demanda muito grande para uso de UAVs. No meio militar, eles podem ser aplicados no reconhecimento de lugares, vigilância, avaliação de danos após desastres naturais ou como mecanismo de contingência para a comunicação. No meio civil, são usados na monitoração ambiental (poluição, florestas, clima ou aplicações científicas), acompanhamento de incêndios, controle de fronteiras, combate ao tráfego de drogas, vigilância aérea, mapeamento, monitoramento de tráfego, ajuda humanitária, buscas e salvamentos e na agricultura de precisão.

Apesar de muito útil, um UAV é um equipamento que precisa ser manuseado com cuidado, fato que dificultou por anos sua homologação para uso comercial no Brasil. Isso devido as suas hélices que giram em alta velocidade e ao seu peso, que pode variar de alguns gramas à vários quilogramas.

Inicialmente, em 2015, o Ministério da Defesa Comando da Aeronáutica lançou o ICA 100-40, que é a regulamentação para Sistema de Aeronaves Remotamente Pilotadas e o Acesso ao Espaço Aéreo Brasileiro (AERONÁTICA, 2015). Tal regulamento previa voos de UAVs no espaço brasileiro, porém, sem muitos detalhes e com várias limitações.

Em 2 de maio de 2017 a ANAC tornou pública a nova legislação sobre uso de UAVs que promete impulsionar e trazer maior segurança ao seu uso comercial no Brasil. Alguns limites estabelecidos no novo regulamento seguem definições de outras autoridades de aviação civil pelo mundo, como *Federal Aviation Administration* (FAA), *Civil Aviation Safety Authority* (CASA) e *European Aviation Safety Agency* (EASA), reguladores dos Estados Unidos, Austrália e da União Europeia, respectivamente. O Anexo B.1 apresenta um resumo e pontua alguns detalhes da nova norma, chamada de Regulamento Brasileiro de Aviação Civil Especial – RBAC–E nº 94, publicado no Diário Oficial da União do dia 03 de maio de 2017.

Em síntese, para homologação de uso, os UAVs foram divididos em três categorias: Classe 1, dos UAVs com peso superior a 150kg; Classe 2, para UAVs de 25 a 150kg; e Classe 3, para veículos com peso inferior a 25kg.

A maioria dos UAVs utilizados para recreação pertencem a classe 3, denominados na regulamentação de aeromodelo e podem voar sem um registro prévio, porém respeitando algumas regras, como estar a 30 metros de distância na horizontal de pessoas e não ultrapassar 120 metros de altura. Para voar com o aeromodelo próximo de pessoas é necessário que elas concordem



previamente com a operação, ou seja, as pessoas precisam saber e concordar com o voo daquele equipamento nas proximidades onde elas se encontram (ANAC, 2017).

Os aeromodelos da classe 3 costumam ter baixo custo e apresentam uma boa relação custo x benefício, uma vez que possuem menor burocracia para o voo e são facilmente adquiridos no mercado nacional. Dentre os principais modelos estão o Phantom da empresa DJI e o AR. Drone, da empresa Parrot.

Ainda assim, a legislação que trata do uso do espaço aéreo brasileiro por aeronaves remotamente pilotadas deve passar por constante revisão e adequação, dada a natureza dinâmica da atividade e dos avanços tecnológicos recorrentes. Dentre as lacunas presente neste regulamento está a ausência da regulamentação para voos autônomos, ficando seus voos limitados às áreas não povoadas. Essa limitação não inibe estudos e pesquisa, podendo essas serem realizadas por universidades e empresas em propriedades privadas e não habitadas, respeitando a distância de 30 metros de pessoas.

Devido a pouca e frágil regulamentação, um dos poucos cenários nos quais pode ser aplicado o uso do voo autônomo de UAVs é na agricultura de precisão, já que comumente os voos acontecem em áreas de terra não habitada. A vantagem de utilizar voo autônomo neste cenário é que não existe a limitação da distância entre o UAV e seu controlador. Em voos controlados, muitas vezes o controle ocorre via radiofrequência, ficando limitado à potência do transmissor. Outra desvantagem do voo controlado é que parte da bateria do UAV é comprometida com a comunicação. Assim, em voos autônomos é possível abranger uma área maior, fazendo com que a busca por informações utilizando sensores ou câmeras acopladas ao UAV seja mais eficiente do que em voos controlados.

A importância dos UAVs na agricultura de precisão é destacada por autores como, George et al. (2013) que registram que os UAVs fornecem a melhor plataforma para avaliar a produção agrícola, sendo rápidos e eficientes se comparados às técnicas terrestres tradicionais, permitindo a leitura de dados em praticamente todos os tipos de terreno. Brandão et al. (2015) complementam, ressaltando que os UAVs são dispositivos versáteis que podem ser usados nas mais variadas aplicações agrícolas, tais como monitoramento automático de plantios e de canais de água e irrigação, detecção e controle de pragas agrícolas e na pecuária, na contagem de animais no pasto, por exemplo.

Ainda, segundo Brandão et al. (2015) é nos voos autônomos que os UAVs se destacam no meio agrícola, já que frequentemente são usados para monitoramentos e inspeções de grandes áreas de plantio. Anthony et al. (2014), cujo trabalho propõe o uso de UAVs na agricultura para monitorar o crescimento das plantas, também ressalta o uso de veículos aéreos operando de forma autônoma, já que este pode fazer leituras dos dados e, por meio de algoritmos específicos, classificá-los sem necessidade de interação de um usuário especialista. Com voos autônomos e o processamento no próprio aparelho, a quantidade de dados processados e a área analisada podem ser muito maiores.

Porém, o uso de UAV da classe 3, em especial de Drones, no monitoramento da agricultura possui algumas limitações e desafios, em especial, para o monitoramento de uma grande extensão de terra. Como uma das grandes limitações dos UAVs de baixo custo é a baixa autonomia, voos relativamente longos tornam-se inviáveis, assim como o monitoramento de uma grande área utilizando um único UAV.

Uma solução para este tipo de situação é utilizar um grupo de Drones que colaborem entre si para otimizar o monitoramento de uma grande área, garantindo rapidez e criando meios de contingência. Dessa forma, caso ocorram problemas com um dos Drones durante o monitoramento, os outros podem assumir as tarefas desse.

Outra vantagem está na otimização do uso de recursos, neste caso, o tempo de voo e, conseqüentemente, o uso da bateria. Utilizando vários Drones classe 3 para a realização de uma missão, eles podem otimizar o tempo de voo se comparado ao desenvolvimento desta missão por um único UAV do tipo asa fixa, que é comumente utilizado na agricultura de precisão. Para facilitar o controle, existe a possibilidade de o Drone pousar, recarregar sua bateria e levantar voo de forma autônoma. Assim, a execução da missão pode ser realizada de forma rápida e sem interação humana.

Ainda, a utilização de um algoritmo de otimização de voo de um grupo de Drones pode possibilitar menor desgaste dos Drones, uma vez que estes permanecerão o menor tempo possível em voo, além da rapidez no desenvolvimento de uma missão.

Considerando este contexto, a presente tese apresenta um modelo aplicado para otimização de voo para grupos de Drones classe 3. Para simplificar a referência a este modelo ao longo do texto, será utilizada a sigla MAOVGD (Modelo Aplicado de Otimização de Voo para Grupo de Drones). O modelo e um algoritmo de otimização dinâmico e aberto aplicado a este modelo, ambos propostos, serão testados. O MAOVGD considera a realização de missões cooperadas entre os elementos contidos no modelo (Drones, Bases de Recargas e Pontos de Passagem), garantindo:

- **Abertura:** propriedade que garante a inclusão e a exclusão de recursos dentro do modelo, tais como UAVs e pontos a serem visitados, durante sua execução;
- **Dinamicidade:** propriedade que garante a execução do modelo, independente dos eventos adversos ou falhas;
- **Otimização:** o modelo proposto contará com um algoritmo de otimização dinâmica que auxiliará na criação dos planos de voos dos múltiplos Drones.

O modelo é composto por um sistema Multiagente, cuja organização dos agentes garante a flexibilidade e a abertura do ambiente. Para garantir a dinamicidade, os agentes de software serão codificados como agentes cognitivos e, por fim, a otimização do plano de voo acontecerá por meio de um algoritmo baseado em leilões recursivos, que garantem bons resultados em um pequeno tempo de processamento, assim, esta otimização não comprometerá a dinamicidade do ambiente.

## 1.2 OBJETIVOS

A seguir serão apresentados os objetivos geral e os objetivos específicos da tese.

### 1.2.1 Objetivo Geral

Propor e testar um modelo aplicado para otimização do tempo de voo de um grupo de Drones, garantindo sua abertura, dinamicidade e otimização do tempo de voo.

### 1.2.2 Objetivos Específicos

Dentre os objetivos específicos, destacam-se:

- Propor um Sistema Multiagente que assegure a abertura e a dinamicidade do modelo proposto;

- Apresentar e avaliar um algoritmo de otimização que possa ser aplicado a um modelo aberto e dinâmico, garantindo a otimização do tempo de voo dos múltiplos Drones;
- Aprimorar o algoritmo de otimização, por meio de heurísticas, permitindo que ele tenha bom desempenho em diferentes cenários.

### 1.3 JUSTIFICATIVA

O MAOVDG propõe um modelo para impulsionar o uso de Drones classe 3 para o monitoramento de grandes áreas de terra, no qual comumente se insere a agricultura de precisão. Hoje, este tipo de voo costuma acontecer por meio de UAV do tipo Asa Fixa, que são mais caros que os Drones e possuem algumas limitações.

Do ponto de vista econômico, se comparado o Asa Fixa Wingo, que possui autonomia de 10 horas, com um Drone Phantom, com autonomia de 30 minutos, em teoria, um conjunto de 20 Phantom consegue permanecer no ar o mesmo tempo que um Wingo, porém, um Wingo custa em torno R\$ 600.000,00 e um DJI Phantom custa aproximadamente R\$ 2.000,00.

Ainda, durante o voo de um asa fixa, se houver qualquer variação em seu alinhamento, o que é muito comum acontecer, haverá erros na leitura dos sensores ou distorções nas imagens capturadas, o que dificulta a identificação dos pontos de união e, conseqüentemente, a geração das ortofotos. Isso não ocorre com Drones, pois eles voam em baixa velocidade e podem permanecer imóveis no ar até que uma leitura sem interferências seja realizada.

Hoje existem empresas que já utilizam Drones da classe 3 para fazerem algum tipo de monitoramento aplicado à agricultura de precisão, mas nesta situação, são necessários vários voos do Drone para cobrir uma área de terra relativamente pequena, sendo comum o uso de um conjunto de 5 ou 6 baterias, havendo a interação humana para pouso, troca de bateria e decolagem, justificando a necessidade de um modelo de voo autônomo e otimizado para um grupo de Drones.

Dada a natureza dinâmica do ambiente no qual os voos acontecem, existe a necessidade do modelo proposto ser aberto, o que significa que Drones poderão ser incluídos e excluídos no/do modelo durante a execução de uma missão, que pode levar de alguns minutos a várias horas. Esta abertura será provida pela modelagem do problema como um Sistema Multiagente.

Referente à dinamicidade, o modelo deve se recuperar de eventos adversos, tais como o término precoce da bateria de algum Drone ou eventos naturais, como mudança da direção ou da velocidade do vento. A característica da dinamicidade do modelo será garantida pelo uso de agentes cognitivos.

Em relação ao algoritmo de otimização proposto, ele deve gerar planos de voos otimizados rapidamente após detectar eventos adversos, uma vez que em ambiente com vários Drones pousando e decolando a todo momento, o tempo entre o pouso de um Drone e a decolagem de outro pode ser de poucos segundos. Assim, com base nas informações recuperadas pelos Drones que pousaram, um novo processo de otimização ocorrerá e novos planos de voos serão enviados a todos os Drones que ainda estão nas bases.

Do ponto de vista computacional, um problema como esse, que é de natureza dinâmica, formado por vários pontos de passagem, com muitos Drones e bases de recarga, com alocações de tarefas é classificado como um problema de complexidade P-Space.

De acordo com Sipser (2007), P-Space é o conjunto de todos os problemas de decisão que podem ser resolvidos por uma máquina de Turing usando uma quantidade polinomial de espaço em memória. Embora os problemas do tipo NP precisam apenas de uma quantidade polinomial de recurso de processamento, nos problemas do tipo P-Space, é necessário preocupar-se, além do processamento, em haver uma quantidade polinomial de memória para obter soluções.

Pode-se fazer uma analogia desse problema com o clássico Problema do Caixeiro Viajante (TSP - *Travelling Salesmen Problem*), porém, no problema da otimização do voo de um grupo de Drones, é necessário considerar que são centenas de pontos de passagem, com vários Drones, bases de recarga e seu uso precisa ser agendado, tornando-se um problema de *Scheduling*. Há que se considerar, ainda, que o ambiente é dinâmico, que o custo entre os pontos de passagem podem mudar de acordo com as condições climáticas e que elementos como Drones podem ser incluídos ou removidos dinamicamente no ambiente. Estas características classificam esse problema como P-Space, tendo complexidade computacional superior a resolução de um TSP clássico.

Para os testes foi utilizado um ambiente de simulação desenvolvido pelo autor desta tese para validar o modelo e o funcionamento do Sistemas Multiagentes, visando garantir a abertura do modelo e a codificação destes utilizando agentes cognitivos assegurando a dinamicidade. Também foi possível testar no ambiente de simulação o algoritmo de otimização de voos baseado em leilões recursivos, o que permite que o processamento deste seja distribuído em diferentes agentes de software.

Ao final da tese, o MAOVDG foi abstraído em um modelo genérico para aplicação em outros cenários da Internet das Coisas que necessitam de otimização distribuída.

#### 1.4 ORGANIZAÇÃO DA TESE

A presente Tese está dividido em capítulos, sendo este o primeiro que apresenta o contexto da tese, os objetivos e a motivação para o seu desenvolvimento.

O Capítulo 2 apresenta na seção 2.1 o contexto de pesquisa, focando nas vantagens e desvantagens da utilização de um conjunto de Drones aplicado ao cenário da agricultura de precisão, sendo que para tal contextualização foram utilizadas algumas técnicas de Interface Homem Computador (IHC). A seção 2.2 apresenta um mapeamento sistemático sobre cooperação de veículos autônomos, com detalhes de como os artigos foram selecionados para leitura/estudo. Na seção 2.3 estão os trabalhos correlatos com as publicações consideradas mais relevantes para esta Tese. Em seguida é apresentado o problema do cenário agrícola.

O Capítulo 3 apresenta os conceitos fundamentais para o desenvolvimento da tese. O referencial teórico sobre desenvolvimento baseado em sistemas multiagentes está na Seção 3.1, em 3.2 sobre agentes de software cognitivo, em 3.3 sobre negociação entre agentes utilizando técnica de leilão e em 3.4 sobre complexidade computacional de algoritmos de otimização.

O Capítulo 4 apresenta as ferramentas utilizadas. Em 4.1 MOISE+ para organização do Sistema Multiagente, 4.2 JASON para o desenvolvimento dos agentes cognitivos, 4.3 FIPA Contract-Net como protocolo de leilão e 4.4 Otimização Utilizando Leilões Recursivos como método de otimização do modelo proposto.

O Capítulo 5 apresenta o modelo aplicado proposto. Em 5.1 Organização dos Agentes para o Modelo Proposto, 5.2 Codificação dos Agentes em Linguagem JASON, 5.3 FIPA Contract-Net aplicado ao Modelo de Otimização de Voos de Drones e 5.4 Leilões Recursivos.

O Capítulo 6 detalha as heurísticas e as metodologias utilizadas para a otimização dos resultados do algoritmo proposto, sendo que na seção 6.1 são apresentados os cenários utilizados para realizações dos testes, na seção 6.2 são apresentadas as heurísticas utilizadas, na seção 6.3 são apresentadas algumas metodologias que permitem encontrar resultados de maneira mais rápida.

O Capítulo 7 apresenta os resultados obtidos, sendo que em 7.1 são apresentados os resultados obtidos em um ambiente de simulação para cenários formados por um ou mais Drones, com dezenas, centenas e milhares de pontos. Em 7.2 a execução modelo modelo é realizado em

dispositivos com processamento limitado, mostrando a viabilidade deste. Em 7.3 são apresentados os testes de Abertura e Dinamicidade do modelo, a fim de testar o Sistema Multiagente proposto, assim como a codificação utilizando agentes cognitivos. Em 7.4, é realizada uma comparação do modelo proposto com outros trabalhos encontrados na literatura. Na seção 7.5 é apresentando um segundo cenário que pode ser beneficiado pelo modelo, e por fim, em 7.6, são apresentadas as publicações realizadas ao longo do desenvolvimento desta tese.

Ao final tem-se o Capítulo 8 com a conclusão, assim como com as dificuldades encontradas e as sugestões de trabalhos futuros.

Nos Apêndices são apresentados 16 artigos publicados/submetidos sobre estudos e técnicas que conduziram ao longo dos últimos quatro anos o desenvolvimento desta Tese. No Anexo é apresentando um resumo da nova regulamentação de voo de Drone no Brasil.

## 2 PROBLEMA DE PESQUISA

Este capítulo inicia apresentando o contexto da aplicação de otimização do tempo de voo para grupo de Drones.

Para verificar a relevância do trabalho, assim como as lacunas de pesquisa, foi realizado um mapeamento sistemático, no qual os principais trabalhos foram identificados e estudados, subsidiando a proposta de um modelo de otimização dinâmica.

Por fim, é apresentado um modelo aplicado para otimização de voo de grupo de Drones, assim como detalhado um problema real no qual o modelo proposto será aplicado.

### 2.1 CONTEXTO DA PESQUISA

O termo Drone será utilizado ao longo desta tese para fazer referência a todo Veículo Aéreo Não tripulado do tipo asa rotativa, que pode possuir 4 hélices (quadricóptero), 6 hélices (hexacópteros), 8 hélices (octacópteros) e, assim, sucessivamente.

Os Drones possuem características que os tornam excelentes para aplicação na agricultura de precisão. Hoje, é comum utilizar neste cenário veículos do tipo Asa Fixa, que possuem maior autonomia, mas que também possuem uma série de limitações. Um estudo comparativo entre UAV do tipo Asa Fixa (pequenos aviões) e UAV do tipo Asa Rotativa (Drones) aplicado a Agricultura de Precisão foi publicado por Brito et al. (2019a), o texto completo está no Apêndice A.6. Em resumo, destacam-se como principais vantagens dos Drones:

1. **Precisão na leitura dos dados:** como os Drones possuem um sistema de estabilidade formado por pares de hélices que os permite voar em baixa velocidade, é possível fazer uma leitura detalhada das informações vindas dos sensores ou das câmeras neles presentes. Outra vantagem em relação aos Asas Fixas está na dinamicidade da altitude do voo. Os Drones podem voar a alturas mais baixas para a leitura detalhada dos dados, ou, se necessário, subir em alta velocidade vertical a alturas maiores, para uma visão mais ampla da área;
2. **Facilidade de manobra:** manobras acentuadas podem ser realizadas com Drones, ao contrário dos Asas Fixas, que costumam fazer grandes curvas para mudarem de direção. A inclinação acentuada de um Asa Fixa durante uma curva também é uma desvantagem, já que neste momento, ele não pode obter fotos nem fazer leitura com seus sensores, pois os dados estariam distorcidos. Os Drones, por sua vez, costumam voar sem a necessidade de inclinações acentuadas, permitindo leituras mais precisas o tempo todo;
3. **Não necessita da interação humana na decolagem e aterrissagem:** como as decolagens e aterrissagens dos Drones acontecem na vertical, elas podem ocorrer sem a interação humana, sendo necessário apenas uma base plana para que esses procedimentos sejam realizados. Já nos Asas Fixas, a natureza de tais operações é na horizontal, sendo necessário um operador humano para lançá-los ao voo e resgatá-los após o pouso;
4. **Custo:** outra vantagem do Drone está no seu custo que costuma ser muito inferior ao do Asa Fixa. Em alguns casos, com o custo de um único Asa Fixa é possível comprar um conjunto de Drones.

Até mesmo a velocidade superior de deslocamento, uma das vantagens do Asa Fixa em relação ao Drone, acaba se transformando em desvantagem em cenários de monitoramento agrícola, uma vez que a alta velocidade pode ocasionar o arrastamento das imagens capturadas pelas câmeras, assim como uma baixa precisão dos dados coletados a partir dos seus sensores.

Após a definição de que seriam utilizados Drones na tese, foi realizado um estudo para a definição do escopo do modelo. Este foi o primeiro trabalho publicado, em 2017, no qual foi identificada a viabilidade e a aplicabilidade da pesquisa, uma vez que existe uma série de limitações legais para voos de Drones, além de problemas da interação dos Drones com o meio no qual eles se inserem. O resultado foi publicado em Brito et al. (2017b) (artigo completo no Apêndice A.1). Neste estudo, as pessoas e as empresas que interagem com o grupo de Drones foram chamados de *stakeholders*.

Em resumo, de acordo com o estudo realizado nesse artigo, as interações dos Drones com os *stakeholders* foram definidas por meio de uma abordagem exploratória das técnicas utilizadas no cenário de Interface Homem Computador (IHC), tais como *Problem Solving* (Oulasvirta e Hornbæk, 2016), *Stakeholder Diagram* (Liu, 2000), *Evaluation Frame* (Baranauskas et al., 2000), *Value Pie* (Pereira e Baranauskas, 2014) e *Building Blocks of Culture* (Hall, 1959).

Cada técnica trouxe importantes contribuições para a definição do escopo do modelo proposto como resultado desta tese, conforme segue:

1. ***Problem Solving***: foi utilizado o *Problem Solving* Construtivo, que reforça a necessidade de tornar a pesquisa mais aplicada possível, a fim de resolver um problema pertinente. No lugar da construção de modelos genéricos, esta técnica orienta a construção de modelos aplicados na solução de problemas existentes. Usando essa técnica foi proposta a solução para o uso de Drones em cenários de monitoramento de precisão, tais como na agricultura, em florestas, áreas de fronteiras e busca e resgate. Embora possa ser utilizado em vários cenários, nesta tese o uso de Drones será direcionado para a agricultura de precisão;
2. ***Stakeholder Diagram e Evaluation Frame***: após o uso destas técnicas foi constatada a necessidade de aplicar o uso de Drones em cenários não urbanos, apontando uma grande oportunidade para a agricultura de precisão. Outra contribuição foi a necessidade de tornar a interação desses Drones com o usuário o mais simples possível, dada a complexidade de controlar múltiplos veículos simultaneamente, sendo que os Drones precisam ser autônomos no pouso, na decolagem, no carregamento da bateria e no voo. Outra contribuição após o uso destas técnicas é a verificação da possibilidade de utilizar veículos terrestres móveis para realizar o recarregamento da bateria dos Drones;
3. ***Value Pie e Building Blocks of Culture***: contribuíram nos aspectos de autonomia e inteligência para tomada de decisão, sendo que a solução proposta, assim como seu operador, devem ser especialistas, minimizando riscos de acidentes com os Drones. Outra importante contribuição destas técnicas diz respeito a encontrar mecanismos para minimizar o tempo de voo dos Drones, além de facilitar a inclusão/exclusão de elementos do modelo dinamicamente e de recuperar-se automaticamente dos eventos adversos e falhas.

Com base nos estudos realizados, foi possível definir o escopo do MAOVGD, conforme Figura 2.1. Esta apresenta um grupo de Drones, os quais foram programados para fazerem pouso e decolagem autônoma. As bases de recarga são fixas, mas poderiam ser veículos *off-road* autônomos, reposicionando-se no terreno a ser monitorado enquanto os Drones encontram-se

recarregando. A comunicação no modelo acontecem apenas entre as bases de recarga, já que a sua energia não é limitada, como acontecem nos Drones. Desta forma, o modelo é orientado a evento, pois depende do pouso do Drone para alimentar o modelo com novas informações, que resultará na otimização de novos planos de voo. Cada base possui um mecanismo que permite a comunicação umas com as outras, garantindo a execução distribuída do algoritmo de otimização proposto.

Para o gerenciamento da missão por um usuário, haverá um sistema web de gerenciamento deste, por meio do qual Drones são atribuídos a missão, assim como as bases de recarga. É, ainda, definido o perímetro do voo e a posição inicial das bases de recargas e a distância de realização entre leituras.

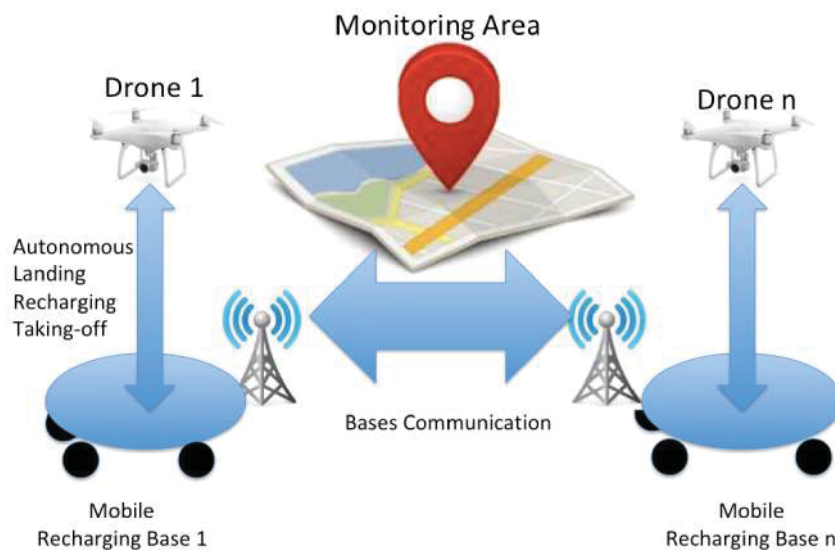


Figura 2.1: Visão geral de um modelo aplicado de otimização de voos para grupo de Drones

A seguir serão apresentados os desafios para permitir que o MAOVGD possa ser aplicado em um cenário real, assim como os trabalhos já desenvolvidos ou auxiliados pelo autor desta Tese para ser possível tal aplicação.

1. **Pouso/Recarregamento/Decolagem Autônoma:** esse assunto já foi explorado por outros autores, inclusive, o autor desta Tese orientou um trabalho de TCC, publicado em Loureiro et al. (2019) (artigo completo no Apêndice A.5). Em resumo, este apresenta um modelo de pouso, recarregamento e decolagem autônomos de um AR. Drone 2.0, da empresa francesa Parrot. Quando a bateria alcança um nível crítico pré-definido pelo usuário, o Drone retorna para a base de recarga utilizando um módulo GPS presente nele e, em seguida, por meio de processamento de imagens, o Drone inicia o processo de aterrissagem. Enquanto o Drone se encontra em solo, ele recebe as novas instruções de voo, estando após a recarga, apto a realizar uma nova decolagem. Outro trabalho realizado, vindo o pouco na base de um Drone montando pela universidade, motivou um trabalho de comparação para fazer o pouso usando o controle PID, no lugar de lógica Fuzzy. O resultado deste trabalho foi publicado em Favarim et al. (2018) (artigo completo no Apêndice A.3);
2. **Base de Recarga:** embora não seja obrigatório, as bases de recarga em ambientes rurais podem ser móveis, como um veículo *off-road* alimentado por placas de energia solar. Assim, enquanto o Drone pouso e inicia o processo de recarregamento, que pode



demorar vários minutos, a base movimentar-se para uma nova posição estratégica a fim de otimizar o tempo de voo em decorrência do deslocamento realizado pelo veículo. Um trabalho inicial do projeto deste veículo foi realizado no mesmo laboratório onde o autor desta Tese trabalha (Puhl et al., 2018) . Referente ao reposicionamento do veículo autônomo de recarga, existem técnicas para escolha da melhor posição para este, sendo um dos trabalhos devolvido por um acadêmico, na orientação do autor desta tese, e publicado em Saque et al. (2019) (artigo completo no Apêndice A.7);

3. **Comunicação entre as Bases de Recarga:** a otimização da autonomia do Drone é um fator importante para o modelo como um todo, já que esta é a grande limitação do seu uso na agricultura de precisão. Assim, sugere-se que os Drones não se comuniquem entre si, havendo a comunicação apenas com as bases, uma vez que elas possuem mecanismos de carregamento solar, sem limitação de bateria. Desta forma, ao retornar para a base de recarga, o Drone atualiza o modelo como um todo, compartilhando a área que ele explorou, atualizando as informações como os dados dos sensores e da câmera e os dados como velocidade/direção do vento. O modelo processará tais informações e definirá um novo plano de voo para todos os Drones em solo. Este tipo de comunicação é chamado de *half-line* e é baseado em eventos. Para o modelo proposto será utilizado o evento de pouso do Drone para iniciar novas etapas de otimização. Um trabalho referente a comunicação *half-line* entre as bases de recarga foi desenvolvido na co-autoria do autor desta Tese e submetido para publicação em Netto et al. (2020) (artigo completo no Apêndice A.11);
4. **Área a ser monitorada:** para a execução do modelo, a área a ser monitorada deve ser conhecida e cadastrada em um sistema de gerenciamento de missão. Neste sistema, todos os pontos de visita devem ser informados, assim como os pontos proibidos, quantidade de Drones necessários para executar a missão, quantidades de bases de recarga disponíveis, posição inicial dessas bases, distância entre os pontos para coleta de dados, entre outros. Como o sistema para o gerenciamento envolve o controle de múltiplos Drones, este deve ser o mais simples possível para o usuário, de modo que informe apenas os dados necessários para início da missão e após isso, o controle dos Drones seja realizado de forma automática por um sistema inteligente. O sistema web para gerenciamento de voo de múltiplos Drone foi desenvolvido por um acadêmico, na orientação do autor desta Tese, e publicado em Filho et al. (2018) (artigo completo no Apêndice A.4). Este não é um sistema inteligente, mas uma interface gráfica que permite o cadastramento de informações passo a passo.

O projeto como um todo é muito extenso para ser abrangido por uma única Tese de doutorado, entretanto, foi verificada sua viabilidade em estudos paralelos que envolvem: pouso/recarga/decolagem autônoma, desenvolvimento de uma estação de recarga móvel *off-road*, comunicação entre as estações de recarga, desenvolvimento de um software para gerenciamento de missão, assim como o desenvolvimento de um algoritmo para escolha da melhor posição dentro de uma área para iniciar a nova missão. Embora sejam trabalhos paralelos, a presente Tese não foca nestas questões.

Sabendo que todos os itens anteriores são possíveis de aplicação, a presente Tese propõe e testa a inteligência do MAOVDG, que envolve a proposta de um modelo de otimização aberto e dinâmico para o voo de grupo de Drones. Esta proposta é composta de um algoritmo de otimização baseado em leilões recursivos que retorna respostas rápidas e um modelo que aceita inclusão e exclusão de elementos em tempo de execução e trata as mudanças do custo de voo

entre um ponto e outro. Para a abertura e dinamicidade, foi proposto um modelo baseado em Sistemas Multiagente e agentes cognitivos.

## 2.2 MAPEAMENTO SISTEMÁTICO SOBRE COOPERAÇÃO DE VEÍCULOS AUTÔNOMOS

Após a definição do cenário de pesquisa, foi realizado um mapeamento sistemático para identificar os possíveis trabalhos relacionados à voos de grupos de UAVs. O resultado deste foi publicado em Brito et al. (2017a) (artigo completo no Apêndice A.2). Embora o mapeamento sistemático tenha ocorrido em 2017, ele foi atualizado durante os anos de 2018, 2019 e 2020.

Para a presente tese, optou-se por selecionar para estudo todos os artigos que envolvem qualquer tipo de veículos não urbanos que cooperassem entre si. A opção de utilizar veículos não urbanos visa reduzir o número de artigos retornados na pesquisa, uma vez que existem muitos trabalhos envolvendo automóveis inteligentes. Assim, foram utilizados apenas os trabalhos que envolvem veículos rurais, UAVs, veículos aquáticos, entre outros não urbanos.

Um mapeamento sistemático consiste na realização de busca em largura, explorando um determinado cenário de forma mais ampla antes de refinar a pesquisa para um tema específico por meio de uma busca em profundidade. As etapas para a realização deste mapeamento foram definidos segundo Petersen et al. (2015), que orienta que um mapeamento sistemático seja conduzido seguindo cinco etapas, aplicadas de forma sequencial:

- Definição das Questões de Pesquisa (QP<sub>i</sub>);
- Busca por estudos preliminares de relevância;
- Seleção dos estudos primários;
- Categorização dos artigos;
- Mapeamento e extração das informações que respondam às questões previamente estabelecidas.

A hipótese de pesquisa a ser respondida por esta tese é: é possível realizar o monitoramento de grandes quantidades de terra utilizando voo de múltiplos Drones que cooperam a fim de otimizar seus voos e permitindo a abertura e dinamicidade do modelo? A partir deste modelo é possível abstrair para um modelo genérico de otimização distribuída que possa ser adaptado para outros cenários da Internet das Coisas?

Para formular as Questões de Pesquisa, a metodologia *Population, Intervention, Comparison and Outcomes* (PICO), sugerida por Kitchenham e Charters (2007), foi utilizada para criar as *Strings* de busca submetidas às bases de dados de publicações científicas.

Para a presente tese, foram definidas quatro Questões de Pesquisa que delimitam o espaço de busca por trabalhos científicos:

- **QP01:** Quais são os principais estudos que envolvam múltiplos veículos autônomos que colaboram entre si para atingirem um objetivo específico;
- **QP02:** Quais os principais estudos que resultaram em modelos ou *frameworks* de cooperação entre veículos autônomos;

- **QP03:** Como abstrair desses estudos um modelo genérico que possa ser aplicado não só a um cenário específico, como por exemplo, de veículos aéreos não tripulados, mas para qualquer conjunto de veículos autônomos não urbanos (aéreo, terrestre ou aquáticos) e que se adapte a ambientes dinâmicos com frequentes entradas e saídas de veículos, assim como se recuperem de eventos adversos, tais como o término precoce da bateria/combustível de algum desses veículos, rajadas de vento, chuva, congestionamentos, entre outros;
- **QP04:** Qual o melhor cenário real para testar a arquitetura proposta pela QP03.

Com base nas questões de pesquisa, foram definidas as palavras-chave, juntamente com seus sinônimos, para a construção das *Strings* de busca nas bases de publicações existentes na Web. As seguintes bases foram utilizadas para realizar a busca: IEEE Xplore (<http://ieeexplore.ieee.org>), ACM (<https://dl.acm.org/>), Scopus (<https://www.scopus.com>), Science Direct (<https://www.scopus.com>) e Compendex (<http://www.engineeringvillage.com/>). Esses repositórios foram selecionados tendo como base a experiência reportada por Dyba et al. (2007). Também foram ouvidos professores e pesquisadores que sugeriam essas bases para pesquisa de artigos científicos.

Para a formulação da *String* de busca, foram definidos três grandes eixos para pesquisa:

- Trabalhos que apresentam o uso de veículos autônomos de qualquer natureza (aéreo, terrestre ou aquático);
- Trabalhos que fazem uso de algum mecanismo de cooperação entre os veículos, a fim de atingirem um objetivo comum;
- Trabalhos que resultaram em algum modelo para cooperação entre veículos, podendo ser *framework*, um Sistema Multiagente, um ambiente de simulação, ou algo semelhante.

Observou-se que, desta forma, as *Strings* de busca seriam bastante genéricas e uma grande quantidade de artigos seriam selecionadas, que é o objetivo do presente mapeamento. Recuperando uma grande quantidade de artigos, ampliam-se as chances de encontrar outros trabalhos que possam contribuir direta ou indiretamente na resposta das Questões de Pesquisa.

Para a definição das palavras-chave foram utilizados termos em inglês, uma vez que a maioria dos trabalhos publicados estão nesse idioma. Também foram definidos três grandes eixos de palavras-chave, que acredita-se retornar a grande maioria dos trabalhos relacionados a sistemas para cooperação entre veículos autônomos. São elas: *Autonomous Vehicle*, *Cooperation* e *Framework*, assim como seus sinônimos. Para os sinônimos foi utilizado o ambiente online [www.thesaurus.com](http://www.thesaurus.com) que permite consultar sinônimos para palavras no idioma inglês.

A partir da identificação das palavras-chave, apresentadas na Tabela 2.1, as *Strings* foram concatenadas utilizando o conectivo lógico OR quando pertenciam a um mesmo conjunto e o conectivo AND para conjuntos diferentes.

Essas pesquisas foram realizadas em agosto de 2017, sendo atualizado em maio de 2020. Foram recuperados 1575 artigos, divididos da seguinte maneira: IEEE ([ieeexplore.ieee.org](http://ieeexplore.ieee.org)) - 194 artigos, ACM ([dl.acm.org](https://dl.acm.org/)) - 549 artigos, Scopus ([www.scopus.com](http://www.scopus.com)) - 448 artigos, Science Direct ([www.sciencedirect.com](http://www.sciencedirect.com)) - 61 artigos, Compendex ([www.engineeringvillage.com](http://www.engineeringvillage.com/)) - 323 artigos. Outros artigos mais recentes também foram lidos para complementar este mapeamento.

Após a exclusão dos artigos repetidos, restaram 1076 artigos para serem analisados. Com os artigos únicos, iniciou-se o trabalho de refinamento, com os artigos sendo selecionados para a categorização.

Tabela 2.1: Palavras-chave e sinônimos para a construção da String de busca.

Palavra-Chave	Sinônimo
autonomous vehicle	unmanned aerial vehicle, UAV, autonomous aerial vehicle, self-governing aerial vehicle, uncontrolled aerial vehicle, unmanned drone, autonomous drone, self-governing drone, uncontrolled drone,,unmanned car, autonomous car, self-governing car, uncontrolled car, unmanned vehicle, unmanned ground vehicle autonomous vehicle, self-governing vehicle, uncontrolled vehicle, unmanned boat, autonomous boat, self-governing boat, uncontrolled boat,,unmanned submarine, autonomous submarine, self-governing submarine, uncontrolled submarine.
cooperation	association, teamwork, participation, alliance, combination, partnership, negotiation, colaboration
<i>Framework</i>	environment, system, simulation environment, multi agent system, MAS.

Para o processo de refinamento foram utilizados critérios de inclusão e de exclusão, detalhados no Apêndice A.2.

Aplicando os critérios de inclusão e exclusão, que ocorreram por meio da análise do título, resumo e palavras-chave, foram selecionados para categorização 545 artigos.

Após a seleção dos artigos foi realizada a categorização. A ferramenta Mendeley Desktop foi utilizada para a criação de pastas e subpastas dentro do ambiente, nas quais os artigos eram copiados após a leitura dos dados mais significativos. Também foi desenvolvido um programa em Java SE para recuperar dados do banco de dados SQLite do Mendeley Desktop visando facilitar algumas categorizações.

O mapeamento permitiu identificar que os tipos de veículos mais propícios para o desenvolvimento da presente tese são os veículos aéreos não tripulados. Há uma grande quantidade de estudos com o uso desses veículos de forma cooperativa, sendo que 67,70% dos artigos investigam UAVs nos cenários com multi-veículos, assim como 12,92% fazem uso de UAV integrando com outros tipos de veículos (terrestres ou aquáticos), perfazendo um total de 80,62% dos artigos que tratam diretamente de UAVs.

Chen et al. (2014) resume as vantagens de pesquisar cenários envolvendo UAVs com múltiplos veículos. Segundo o autor, grupos de UAVs podem formar uma rede de veículos para a realização de missões complexas, como resgates, buscas, patrulhamento e mapeamento. Nesses cenários várias áreas de pesquisas surgem, como definição da infraestrutura de rede de comunicação entre os veículos aéreos, técnicas de cooperação e coordenação entre os veículos, modelos de otimização, entre outros.

Chirwa e Lauf (2014) destaca que um único Veículo Aéreo Não Tripulado não é eficiente para realizar tarefas como busca e resgate, reconhecimento e vigilância, uma vez que é necessário realizar vários voos de ida e volta dentro de uma área específica. A solução para este problema é

usar um grupo de UAVs cooperantes, formando um Sistema Aéreo Não Tripulado (*Unmanned Aerial System - UAS*).

O modelo para um ambiente tão dinâmico quanto o cenário com múltiplos UAV justifica a utilização de Sistema Multiagente, no qual cada veículo e recurso do ambiente - tais como plataforma de recarregamento da bateria do veículo, pontos de passagens, restrição física no ambiente como lagos ou penhascos, entre outros - é modelado como um agente de software. Esses agentes cooperam e/ou negociam entre si para atingirem um objetivo comum.

Segundo Chen et al. (2015), as áreas de pesquisas envolvendo Veículos Aéreos Não Tripulados cooperativos estão em amplo crescimento. O principal motivo são as vantagens proporcionadas pelo deslocamento aéreo. Segundo o autor, para obter uma gestão eficaz de cenários com múltiplos UAVs e ambientes dinâmicos, um modelo baseado em um Sistema Multiagente deve ser considerado.

Além do trabalho citado no parágrafo anterior, outros 33 artigos categorizados apresentaram solução para o cenário de múltiplos UAVs baseados em Sistemas Multiagentes, o que reforça o argumento de que esta é uma arquitetura potencial para a proposta de um modelo para cooperação entre veículos.

Em um cenário dinâmico formado por veículos e recursos, é necessária a escolha de uma técnica rápida e flexível de negociação entre os agentes, uma vez que eles devem cooperar para atingirem o objetivo da missão. Dos artigos estudados, 8 tratam especificamente de técnicas de negociação.

Uma técnica de negociação utilizada em alguns artigos é a negociação baseada em leilões. Segundo Pippin e Christensen (2012), métodos de negociação baseados em leilão são frequentemente usados para executar atribuição de tarefas distribuídas em equipes com elementos dinâmicos. Ao todo, foram contabilizados 12 trabalhos que usam técnicas de negociação baseadas em leilões.

Referente ao mapeamento realizado, em relação à questão de pesquisa QP01, dos 523 artigos categorizados, 180 tratavam de algum tipo de cooperação entre veículos autônomos, o que mostra que este tema está sendo bastante explorado e possui potencial para novas pesquisas. Destes, a maioria trata de cooperação em cenários bem específicos, como exploração, busca e resgate, coordenação de movimentos, alocação de tarefas e recursos, entre outros.

Considerando o modelo de cooperação de veículo de uma forma mais genérica, foram selecionados para a leitura na íntegra 14 artigos que tratam de métodos de cooperação utilizando veículos autônomos, assim como 4 artigos que tratam a cooperação entre agentes em um sistema de simulação, já que os algoritmos propostos nos artigos poderiam ser aproveitados para a presente tese.

Referente a QP02, 28 artigos foram selecionados para leitura na íntegra por integrarem o grupo de artigos que tratam de *frameworks* que objetivam a cooperação de veículos autônomos.

Para explorar de forma mais aprofundada o tema de pesquisa, assim como responder a QP03, foi necessária a realização de uma Revisão Sistemática, com os artigos selecionados lidos e analisados. Essa leitura possibilitou identificar as tecnologias utilizadas, a metodologia de trabalho e a análise dos resultados, a fim de propor um tema inédito de pesquisa. Com a leitura desses artigos, também foi possível identificar trabalhos similares e definir uma metodologia para comparação do que já existe com o que está sendo proposto.

Para a QP04, identificou-se que cenários envolvendo UAVs são mais explorados para testes de cooperação entre veículos, uma vez que 262 artigos trataram o uso de UAVs e 50 artigos realizaram a cooperação entre UAVs e outros veículos. Assim, no total foram 312 trabalhos que exploraram este cenário, correspondendo a 80,62% dos artigos categorizados.

### 2.3 TRABALHOS CORRELATOS

Após o mapeamento sistemático, alguns trabalhos foram selecionados para a leitura na íntegra, iniciando assim um trabalho de Revisão Sistemática.

De todos os trabalhos lidos, foram separados os 8 mais relevantes, apresentado a seguir. A Revisão Sistemática foi importante para ressaltar o cenário no qual se aplica o presente estudo e auxiliou na escolha da arquitetura e das ferramentas a serem utilizadas.

O primeiro artigo lido foi escrito por Avellar et al. (2015). Esse é o trabalho mais próximo ao modelo proposto, uma vez que trabalha com otimização de voo de múltiplos UAVs aplicados na agricultura de precisão. Além deste ser, dentre os selecionados para a leitura na íntegra, o trabalho mais citado na literatura. Por este motivo, além do artigo, a Tese que resultou este trabalho também foi lida e analisada.

Avellar et al. (2015) trabalha com a cooperação de dois Asas Fixas aplicados na agricultura de precisão. O autor divide o problema em duas partes: primeiro, ele usa o princípio de fluxo de partículas para fazer os Asas Fixas serem atraídos por pontos pelos quais eles precisam passar. Em seguida, ele utiliza um algoritmo de otimização desenvolvido em Programação Linear Inteira Mista (PLIM) para realizar a otimização de voo dos dois Asas Fixas, conhecendo os pontos de passagem e o local de pouso e de decolagem. Dentre as limitações do trabalho está o fato de ser desenvolvido para apenas dois Asas Fixas, além de ser um modelo estático. Antes do voo, conhecendo os pontos de passagem é executado o algoritmo em PLIM que gera os planos de voos, que posteriormente serão executados pelos Asas Fixas.

Segundo o autor, o número de dois Asas Fixas é o ideal para voos cooperativos, pois quanto maior o número de Asas Fixas, maior a complexidade e o custo total do sistema. Voos com Asas Fixas tem um custo alto para *setup* da aeronave, além da necessidade da intervenção para decolagem e aterrissagem. Aumentando o número de Asa Fixa aumenta o número de operadores do sistema.

Segundo o próprio autor, em pesquisas realizadas para o desenvolvimento do seu trabalho, a justificativa de utilizar Asas Fixas em voos cooperativos é porque a maioria dos trabalhos científicos desenvolvidos entre 2005 e 2015 utilizou esse tipo de veículo.

Apesar do autor utilizar Asas Fixas, ele destaca limitações, como as restrições relacionadas à velocidade mínima da aeronave que não pode ser abaixo de determinado limite para não correr o risco de cair e o não detalhamento na leitura dos dados pela alta velocidade de voo.

Já do ponto de vista do algoritmo de otimização, o mais próximo à proposta deste trabalho, por poder ser executado em ambientes abertos e dinâmicos é proposto por Khaledi et al. (2018). Neste, a autora apresenta um algoritmo de otimização para grupo de Drones, porém, a diferença é que o algoritmo não foi desenvolvido para otimização de voo, mas sim, para roteamento de comunicação em redes de Drones, também conhecidos como MANETs.

O algoritmo apresentado permite grande dinamicidade por ser um algoritmo guloso, assim, ele é de ordem linear e sua complexidade aumenta a medida que novos Drones são inseridos na rede.

O objetivo é que um pacote passe por todos os Drones, coletando informações deste, assim, ele possui algumas simplificações que o modelo proposto não tem, como por exemplo, a limitação da autonomia. No artigo, o objetivo é otimizar o caminho da mensagem que deve passar por todos os Drones, já na presente Tese, o objetivo é otimizar o caminho para passar em todos os pontos, porém, considerando autonomia, podendo ser necessário mais de voo.

Para a escolha do ponto mais próximo, o algoritmo para cálculo de distância euclidiana é utilizado, sendo calculada a distância do nó atual com todos os outros nós e é escolhido o nó de menor distância euclidiana para o roteamento.

Para as comparações e testes deste trabalho, a autora de Khaledi et al. (2018) aplicou o algoritmo em um modelo de otimização de voo de grupo de Drones aplicado na agricultura de precisão, e o trabalho foi submetido em Afghah et al. (2020).

Outro trabalho de grande contribuição foi publicado por Ergezer e Leblebicioğlu (2013) que apresenta um modelo de voo para múltiplos Asas Fixas, testado apenas em ambiente de simulação. Embora simulado, este trabalho considera questões como ângulos e velocidade mínima para que o Asa Fixa consiga realizar manobras.

Assim como o trabalho anterior, Ergezer e Leblebicioğlu (2013) também define os pontos que devem ser visitados pelo Asa Fixa, porém, inclui o parâmetro altura de voo, além de introduzir o conceito de pontos proibidos, que são pontos pelos quais o Asa Fixa não pode passar. O resultado é um plano de voo tridimensional. O trabalho de Ergezer e Leblebicioğlu (2013) visa encontrar um resultado factível, no qual todos os pontos informados devem ser visitados. Para definir o plano de voo foi utilizado Algoritmo Genético (AG), sendo que o sistema é treinado e otimizado antes dos voos das aeronaves, não sendo considerado neste modelo a dinamicidade existente nos modelos reais, como rajadas de vento ou término precoce da bateria.

Como limitação, o autor informa que o trabalho apresentado não é capaz de encontrar a melhor solução possível para o cenário, identificando apenas uma solução factível dada a complexidade computacional envolvida.

Ho et al. (2013) apresenta um modelo para otimizar voos de Asas Fixas, aplicado à coleta de dados de redes de sensores sem fio. O cenário de aplicação é interessante, pois, em redes sem fio um dos maiores custos é a transmissão de dados entre os nós da rede. Para minimizar esse custo, é proposto um sistema para otimização de voo de um Asa Fixa, utilizando a topologia da rede dos sensores sem fio sendo que os sensores são distribuídos em *cluster*, com os sensores próximos enviando todas as informações para um nó chamado de *head* do *cluster*. Esse nó centraliza todas as informações e as encaminha para o Asa Fixa, assim que ele sobrevoar o referido ponto (nó *head* do *cluster*). Os gastos de bateria para troca de informações dentro da rede são comparados com a abordagem tradicional, chamada de *Low Energy Adaptive Clustering Heirarchy* (LEACH) e o modelo utilizando Asa Fixa apresentou economia de bateria do sistema como um todo.

Com o objetivo de otimizar o voo do Asa Fixa que faz a coleta de informação, Ho et al. (2013) utilizou o *Particle Swarm Optimization* (PSO), aplicando um algoritmo clássico utilizado para resolver o problema de *Traveling Salesman Problem* (TSP), o que não trata da dinamicidade, nem mesmo múltiplos Asas Fixas. Porém, o autor destaca na sua conclusão que um dos trabalhos futuros é apresentar técnicas para tratar a coleta de dados com múltiplos Asas Fixas.

Já um trabalho que chamou muito a atenção pela organização de ambientes dinâmicos de voos, sem tratar necessariamente de otimização, foi o publicado por Menegol et al. (2018), no qual o autor embarcou em um Asa Fixa uma infraestrutura para que ele se comportasse como um agente inteligente em um Sistema Multiagente. O autor do trabalho considera que os Asas Fixas estejam conectados entre si o tempo todo e codificam a lógica do Asa Fixa como sendo um agente cognitivo utilizando para isso o *Belief, Desire and Intention* (BDI).

Assume-se que esses Asas Rotativas fazem parte de um Sistema Multiagente dinâmico e aberto, permitindo que novos Asas Fixas sejam inseridos ou retirados dinamicamente. Para garantir tal possibilidade é utilizado o modelo organizacional de agente chamado *Model of Organization for multi-agent SystEm* (MOISE+). Já para o compartilhamento de informações entre os Asas Fixas é utilizada a ferramenta *Common ARTifact Infrastructure for AGents Open Environments* (CArtAgO), que funciona como um repositório de artefatos. Nesse repositório, as crenças, os desejos e as intenções do Sistema Multiagente são compartilhados entre todos os agentes.

Neste artigo, o objetivo principal foi verificar como a plataforma utilizando BDI, MOISE+ e CArTAgO se comporta do ponto de vista de eficiência quando embarcado em um Asa Fixa. Os testes realizados foram de consumo de memória, CPU e velocidade, com os dados sendo comparados com o modelo imperativo tradicional de programação para o Asa Fixa. Os resultados foram muito bons, mostrando que mesmo com o Sistema Multiagente embarcado, os Asas Fixas tiveram um desempenho similar ao modelo tradicional de programação imperativa.

Outro trabalho que também fez uso de Sistemas Multiagente para otimizar o voo de grupo de UAV foi Ernest et al. (2013). O autor propõe um sistema de simulação para otimizar o voo de grupos de UAVs. Para o sistema, o autor aborda o problema como sendo um *Travelling Salesman Problem* (TSP), considera o voo de múltiplos UAVs, embora não é considerado que os UAVs podem retornar para a base para recarregar sua bateria. Para a otimização, o autor criou um algoritmo híbrido a partir da combinação de heurísticas com técnicas estocásticas.

Roberge et al. (2014) apresentam um modelo de otimização de voo de grupo de Asas Fixas a partir de um algoritmo híbrido de PSO e AG. Este trabalho considera a otimização do percursos de Asas Fixas em um cenário tridimensional. Uma das contribuições deste trabalho é que o processamento é executado de forma paralela otimizando o uso dos processadores *multi-core*. Outra contribuição é o uso de função multi-objetivo, permitindo otimizar mais de um parâmetro do sistema. Este trabalho também foi testado apenas em ambiente de simulação.

Por fim, Caska e Gayretli (2015) apresenta um sistema de cooperação entre Drone e *Unmanned Ground Vehicles* (UGV) com o objetivo de calcular a melhor rota de voo dos Asas Rotativas e identificar o número ideal de UGV para fornecer suporte aos veículos aéreos. Neste, considera-se que após o voo, o Drone pode pousar sobre o UGV e este se desloca até outro ponto estratégico, enquanto o Drone recarregue sua bateria. No modelo proposto, um UGV pode carregar até dois Asas Rotativas ao mesmo tempo. Entretanto, o sistema não é *online*, assim, o algoritmo de otimização é executado antes do início da missão, identificando o número ideal de Asas Rotativas e UGV para completar a missão e definindo os planos de movimentação dos veículos envolvidos. A partir deste ponto, o sistema perde sua dinamicidade, sendo executado apenas o que foi previamente calculado.

## 2.4 MODELO APLICADO DE OTIMIZAÇÃO DE VOO PARA GRUPOS DE DRONES

Embora muitos estudos preliminares foram desenvolvidos e mostraram a viabilidade do MAOVD, um tema de pesquisa merece destaque: a otimização do tempo de voo de uma área, considerando também a abertura e dinamicidade do modelo, sendo este tema em específico trabalhado nesta Tese.

Para garantir a abertura do modelo, foi optado por concebê-lo utilizando a arquitetura de Sistema Multiagente. Para garantir a dinamicidade, os agentes foram codificados como agentes cognitivos. Para comunicação e negociação entre os elementos do modelo, foi utilizada técnica de leilão. Já para garantir a otimização dinâmica, foi desenvolvido um modelo de otimização baseado em leilões recursivos.

A escolha do Sistema Multiagente ocorreu por este se comportar muito bem em ambientes abertos, com a inclusão e exclusão de elementos durante a execução do modelo. No modelo proposto, cada elemento é codificado como agente de software. Os agentes negociam entre si visando atingir um objetivo global, que é a otimização do tempo de voo, garantindo com isso a redução coletiva do consumo de bateria dos Drones.

Para garantir que o modelo funcione bem em ambientes dinâmicos, foi utilizada a codificação dos agentes como sendo agentes cognitivos, com crenças, desejos e intenções. O modelo é orientado a eventos, já que crenças são alimentadas a cada pouso de Drone. Dentre as



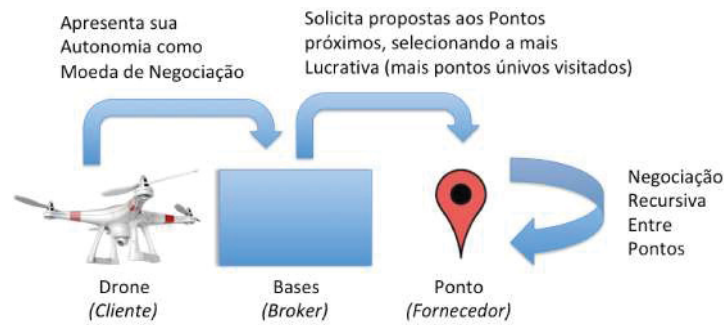


Figura 2.2: Analogia do modelo baseado em leilões recursivos em relação a um leilão tradicional

crenças dos agentes estão informações como velocidade e direção do vento e se um ponto já foi visitado ou não. Já o desejo dos agentes é basicamente a função objetivo do modelo e, neste caso, o que se deseja otimizar é o tempo coletivo de voo. Por fim, as intenções são as ações que devem ser realizado pelos agentes, que é do que seguir o fluxo de leilões até que este seja concluído.

O método de troca de mensagens entre os agentes escolhido foi o leilão. Este método é relativamente simples de ser aplicado em sistemas computacionais a partir do protocolo FIPA Contract-Net e permite que os agentes se comuniquem dentro do modelo independentemente da quantidade de elementos, que pode variar ao longo do tempo.

Para a otimização, estes leilões acontecem de forma recursiva dentro do modelo. A Figura 2.2 apresenta o modelo proposto. Neste, o Drone é cliente e possui como moeda de negociação sua autonomia. Ele solicita propostas de plano de voo ao leiloeiro, neste ambiente representado pela base de recarga, que por sua vez solicita propostas de visita aos seus pontos vizinhos. Estes pontos vizinhos, por fim, solicitam novas propostas, de forma recursiva, a todos os seus vizinhos e, assim, o leilão continua até que o Drone não possua mais dinheiro, ou seja, não tenha mais autonomia suficiente para continuar o voo, sendo escolhido pela base a proposta mais lucrativa dada a função objetivo.

Como função objetivo do algoritmo de otimização, definiu-se a otimização do tempo de voo dos Drones, o que reflete diretamente no uso da bateria. Assim, a proposta que permite o voo por um número maior de pontos, com o menor consumo da bateria do Drone, será escolhida. A ordem de passagem nos pontos, velocidade mínima ou máxima do Drone, altitude necessária, entre outras informações não são restrições do modelo e não estão definidas em sua função objetivo. Assim, o algoritmo tem liberdade para escolha dos valores de tais parâmetros ao longo do tempo.

O algoritmo de otimização utilizado deve permitir bons resultados de forma rápida, uma vez que o cenário de aplicação é dinâmico. Técnicas de otimização matemática que permitem resultados ótimos, mas que levam muito tempo para processamento, tais como PLIM, e métodos que necessitam de treinamento prévio, como Redes Neurais, foram descartados.

Outra característica do modelo proposto está na comunicação entre agentes. Com o objetivo de maximizar o tempo de voo os Drones não se comunicam entre si, destinando, assim, toda a energia das baterias para o voo. Desta forma, este modelo é baseado em eventos, sendo que as comunicações só acontecem entre as bases de recarga, que não possuem bateria limitada. Ao pousar o Drone alimentará o modelo com as informações capturadas durante o voo, iniciará um novo processo de leilão e um novo plano de voo será gerado.

A abertura, a dinamicidade e a otimização do MAOVGD serão testadas apenas em ambiente de simulação, uma vez que testes reais, o que pretende-se realizar em um futuro próximo, dependem de outras partes do modelo, tais como os pouso/rearregamento/decolagem autônoma e comunicação entre as bases de recarga. O ambiente de simulação escolhido, o

MultiDrone Simulator, permite realizar os testes, assim como uma análise quantitativa/qualitativa dos dados gerados.

Para os testes do MAOVGD, algumas simplificações foram adotadas para que pudessem ser executado em ambiente de simulação, tais como:

- Considera-se que os Drones tenham a capacidade de decolar, pousar e recarregar sua bateria de forma autônoma;
- Considera-se que as bases recarreguem a bateria dos Drones sem a intervenção humana e que estas possuam comunicação entre elas;
- Considera-se que o modelo tenha processamento central, podendo ser um *device* portátil, no qual, o usuário poderá, via interface gráfica, delimitar o perímetro de voo, definir a precisão da leitura, ou seja, a distância entre um ponto de passagem e outro e definir onde estarão as bases de recarga;
- A informação que será coletada pelos Drones, tais como fotos ou dados de sensores, não interferirá no MAOVGD, assim, o único objetivo para testes é fazer com que os Drones sobrevoem todos os pontos do sistema;
- Para os testes do MAOVGD, considera-se que as bases de recargas sejam estáticas, ou seja, nesta tese elas não serão tratados como veículos. Também considera-se que o modelo tenha uma base de recarga por Drone, assim, estes não precisam competir pelas bases;
- Não será tratada a colisão entre Drones, uma vez que Drones comerciais, como o Phantom 4, possuem um modo de operação no qual ele identifica possíveis colisões e as evita, pairando no ar ou mudando de altitude por alguns segundos.

Embora exista um tempo considerável entre o pouso de um Drone, o carregamento de sua bateria e sua decolagem para uma nova missão, o processamento e a otimização devem ser o mais rápido possíveis, dado a dinâmica do ambiente, pois, mesmo com um Drone aterrizado, novos eventos de pouso e decolagem podem acontecer devido aos múltiplos Drones presentes no modelo, gerando novas informações na base de crenças e havendo a necessidade de novas otimizações, assim, é necessário haver rapidez no processamento. O algoritmo de otimização não precisa encontrar a melhor solução, mas uma solução factível em um curto espaço de tempo, dado a dinamicidade do ambiente.

Para um melhor entendimento da dinâmica de otimização, tem-se a Figura 2.3. Nesta, um Drone envia um pedido de proposta para a base, que por sua vez, inicia um leilão com cada um de seus pontos próximos que continua de forma recursiva até que a autonomia do Drone seja esgotada. Neste exemplo, consideram-se 5 pontos de autonomia do Drone e que cada ponto tenha dois vizinhos próximos.

No grafo apresentado, o ponto inicial 1 é a raiz do grafo, pois nele se encontra a base de recarga. Do ponto 1, o Drone pode ir para o ponto 2 ou ponto 3, sendo possível observar que esta árvore tem um fator de ramificação igual a 2, sendo este o número de vizinhos. A altura da árvore é a autonomia do Drone, neste caso, 5. Cada aresta do grafo é um leilão que acontece e neste conjunto de 20 possibilidades, a base deve escolher a proposta que percorra o maior número de pontos e que permita retornar para a base.

Neste grafo, como os leilões acontecem de forma sequencial, é realizado uma busca em profundidade da melhor solução, assim, cada caminho é explorado de cima para baixo e da esquerda para direita.

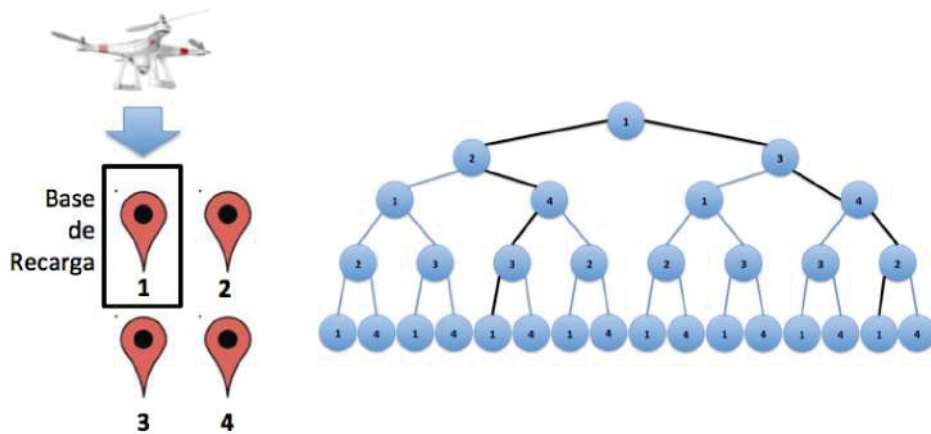


Figura 2.3: A esquerda, cenário com um Drone, uma base de recarga e quatro pontos de passagem. A direita, o grafo das possibilidades de plano de voo para o cenário

A validação deste modelo foi realizada em ambiente de simulação, sendo testado o algoritmo de otimização, assim como a abertura e a dinamicidade do modelo. Dadas suas características, não existe *dataset* para comparação de resultado. Para testar o desempenho do algoritmo de otimização, ele é comparado com Khaledi et al. (2018), este algoritmo de otimização foi desenvolvido na Northern Arizona University que possui um núcleo de pesquisa de Drones aplicados na agricultura.

Após uma análise qualitativa do desempenho do algoritmo proposto, foram realizados os testes de abertura e dinamicidade e, por fim, o modelo foi comparado qualitativamente com outros trabalhos presentes na literatura.

A seguir é apresentado um cenário que será beneficiado com o modelo proposto por esta Tese: o desenvolvimento de um Modelo Digital de Elevação.

## 2.5 APLICABILIDADE DO MODELO EM UM PROBLEMA DO CENÁRIO AGRÍCOLA

Alguns cenários agrícolas merecem destaque para uso de Drones se comparado aos UAVs de asa fixa. Um deles é o desenvolvimento de Modelos Digital de Elevação (MDE), que é uma representação quantitativa da superfície terrestre que fornece informações básicas sobre o relevo do terreno. Os MDE podem ser produzidos a partir de imagens *Shuttle Radar Topography Mission* (SRTM), TOPODATA ou, ainda, a partir de levantamentos topográficos realizados em campo.

As imagens SRTM são fruto da missão SRTM, resultado da cooperação entre a *National Aeronautics and Space Administration* (NASA) e a *National Imagery and Mapping Agency* (NIMA), do Departamento de Defesa dos Estados Unidos e das Agências Espaciais da Alemanha e da Itália - *Deutsches Zentrum für Luft- und Raumfahrt* (DLR). As imagens foram obtidas em fevereiro de 2000, entre as latitudes 60°N e 56°S, por meio do radar *SIR-C/X-SAR* (*Spaceborne Imaging Radar-C/X-Band Synthetic Aperture Radar*) a bordo do ônibus espacial Endeavour, com resolução espacial de 30 metros para os Estados Unidos e de 90 metros para os demais países do mundo. A resolução de 90 metros, muitas vezes, não é precisa devido aos problemas de leitura (Valeriano, 2005).

Segundo Miranda (2005), a EMBRAPA (Empresa Brasileira de Pesquisa Agropecuária), todavia, utilizou as imagens SRTM do território brasileiro disponibilizadas e realizou um tratamento matemático de modo que foi possível obter medidas precisas para altitudes distantes umas das outras em 90 metros para o território brasileiro.

Já as imagens Topodata, por sua vez, resultam de um projeto iniciado pelo Instituto Nacional de Pesquisas Espaciais (INPE) sendo utilizadas imagens SRTM disponibilizada que tiveram os dados processados, inspecionados, revisados e, posteriormente, aprimorados e corrigidos por profissionais do INPE em 2008 e 2009, a fim de obter uma resolução espacial de 30 metros. Visando uma expansão do projeto, em 2011 foi desenvolvida uma metodologia que capacitava a sua aplicação em qualquer dado SRTM pelo processo de interpolação por krigagem (Valeriano, 2005).

Entretanto, para ambos os casos, o uso destas imagens não se torna interessante para pequenos terrenos ou terrenos muito acidentados, que comumente encontram-se na região sul do país. Nesses casos, para um melhor detalhamento, é necessária a interpolação de dados, sendo calculada a diferença de altitude entre dois pontos conhecidos, traçado uma reta entre elas, permitindo estimar a altitude de outros pontos, o que é muito pouco preciso.

Desta forma, a maneira mais precisa de se ter um MDE de terrenos muito acidentados é realizando o levantamento *in loco*. Em pequenas propriedades é possível realizar esse levantamento com uso de instrumentos, com um teodolito. Já em grandes propriedades, o uso de UAVs voando em baixa velocidade facilitaria a obtenção de tais informações.

De posse de um MDE, é possível obter informações de uma área de terra, tais como:

- Mapas de fluxo - identifica o sentido e a intensidade de escoamento da água após uma chuva;
- Mapas de fluxo acumulado - identifica os pontos nos quais se acumulam água após a chuva, assim como a quantidade acumulada;
- Hipsométrico - representação da elevação de um terreno por meio de cores. As cores utilizadas possuem equivalência com a cota do terreno;
- Mapa de orientação de vertente: documento cartográfico que indica a declividade das encostas.

Dada a importância da criação de um MDE, iniciativas do governo incentivam pesquisas nessas áreas, tais como o PronaSolo do Governo Federal (Embrapa, 2016).

Na proposta do PronaSolo, o Instituto Brasileiro de Geografia Estatística (IBGE) criará uma base de dados para acesso da população e dos gestores sobre a elevação do solo no território nacional. A execução do projeto está prevista para um horizonte de 30 anos com um investimento do Governo Federal de R\$ 80 milhões para todo o Brasil (Embrapa, 2016). Uma das propostas do PronaSolo é criar dados que subsidiarão técnicas que permitam eliminar as curvas de níveis, as quais dificultam a operacionalização do solo.

Segundo o site do Governo Federal, os mapas atuais não possuem escala viável para o manejo de solo em nível de propriedade e muito menos para planejamento de microbacias. A Embrapa estima um custo benefício de R\$ 185,00/ha para cada real investido no projeto. Hoje é possível a confecção de mapas menos precisos em escalas maiores. Segundo Embrapa (2016), o Brasil perde cerca de \$ 5 bi/ano em função do processo erosivos.

A utilização de Drones para o mapeamento do solo é uma opção muito eficiente, já que estes veículos aéreos podem possuir, além das tradicionais câmeras, sensores ativos. Os sensores ativos, tais como infravermelho e ultrassom permitem uma maior qualidade dos dados lidos, uma vez que dados obtidos por meio de câmeras podem vir distorcidos, já que algumas áreas de solo podem estar obstruídas por florestas, edificações, vegetações com cortes mais rasteiros, entre outros, o que dificulta a leitura do solo.

Outra vantagem do uso de Drones é que estes podem estar dotados com sistemas modernos e de baixo consumo para geolocalização, um sistema que merece destaque é o *Real Time Kinematic*, ou RTK, o qual permite uma geolocalização baseado em estação rádio-base, por meio de radiofrequência, conseguindo uma precisão maior com um maior consumo de bateria, o que é muito importante para os Drones, que possuem limitação da bateria.

Entretanto, para ambientes muito acidentados, o RTK pode possuir algumas limitações, uma vez que o Drone pode ficar fora da área de visão da estação rádio-base, possuindo, nesta situação, a opção de utilizar o sistema *real-time eXtended*, ou RTX, que faz um híbrido entre o sistema baseado em radiofrequência e satélites.

## 2.6 CONCLUSÃO

Este capítulo apresentou uma contextualização do problema de pesquisa, iniciando com os motivos que levaram a escolha de Drones para o monitoramento agrícola, assim como a abordagem exploratória do cenário de Interface Homem Computador (IHC) que permitiram a definição do cenário de pesquisa, que em conjunto com o mapeamento sistemático realizado forneceram informação suficiente para a definição do Modelo Aplicado de Otimização de Voo de Grupo de Drones.

Dos trabalhos selecionados para a leitura após o mapeamento, 8 foram escolhidos para serem utilizados como parâmetros de comparação, um deles envolvendo um algoritmo de otimização para ambientes dinâmicos e os outros sete para uma comparação qualitativa com o presente modelo.

Em seguida, o capítulo apresentou com detalhes o Modelo Aplicado de Otimização de Voo para Grupo de Drones, como ele proverá a abertura, a dinamicidade e a otimização e, ainda, algumas simplificações que serão consideradas para os testes e como será realizada a validação do modelo.

Ao final, este capítulo apresentou um cenário que poderia ser beneficiado com o voo de grupo de Drones, aplicando na estimativa de MDE do terreno.

### 3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta conceitos fundamentais sobre as principais técnicas e modelos utilizados no desenvolvimento do presente trabalho. Inicialmente são apresentados os conceitos básicos sobre Sistemas Multiagentes e Organização. Na sequência é apresentado o modelo BDI, que permite o desenvolvimento de agentes cognitivos. Após é apresentada a técnica de negociação baseada em leilões e, por fim, uma seção sobre Complexidade Computacional de Algoritmos de Otimização, que caracteriza o problema de otimização a resolvido por esta tese.

#### 3.1 SISTEMAS MULTIAGENTES

Os Sistemas Multiagentes são entidades compostas por agentes de software, que possuem as características de cooperarem entre si para a resolução de problemas que fogem das capacidades individuais dos agentes. Para o desenvolvimento dessas entidades virtuais é necessário utilizar o conhecimento de outras áreas para auxiliar na concepção desse contexto computacional.

Assim como a orientação a objetos recebeu uma grande contribuição da matemática, a representação do conhecimento e raciocínio receberam contribuições da psicologia, da lógica e da biologia concebendo-se, assim, as redes neurais, PSO, AG e outras técnicas de otimização. A área de Sistemas Multiagentes é influenciada pela sociologia, vislumbrando uma concepção de sistema com propriedades que até então somente sociedades reais possuíam (Hübner e Sichman, 2003).

Os Sistemas Multiagentes são formados por quatro aspectos que são:

1. Os agentes;
2. A interação entre os agentes;
3. O ambiente de execução dos agentes;
4. A organização social dos agentes.

Agentes são softwares autônomos e essa autonomia significa que um agente existe independente dos demais e do problema a ser solucionado. Nessa autonomia de existência, um agente não precisa de outro para existir, mesmo que para alcançar seus objetivos ele eventualmente precise da ajuda dos demais.

A interação entre os agentes é obtida por troca de mensagens, como em muitas situações os agentes se localizam em diferentes máquinas da rede, é necessário que tais mensagens sejam compatíveis com o protocolo de comunicação da rede. Além disso, é necessário que as mensagens possuam um protocolo próprio, para que os diferentes agentes, que eventualmente se encontram em plataformas diferentes, consigam manter coerência na comunicação, além de facilitar a inclusão de novos agentes ao sistema. Entre os protocolos de comunicação de agentes destacam-se o *Foundation for Intelligent Physical Agents - Agent Communication Language* (FIPA ACL) (FIPA, 2012) e o *Knowledge Query and Manipulation Language* (KQML) (Finin et al., 1994).

O ambiente no qual os agentes são executados trata-se de um local perceptivo, que interage com os agentes, provendo informações necessárias para a tomada de decisão. Esse ambiente também pode ter seu estado modificado pelas ações executadas pelos agentes.

A organização social estabelece restrições aos comportamentos dos agentes, para que o sistema possua um comportamento grupal coeso. Essa organização está implicitamente definida no sistema. São raras as vezes nas quais exista descrição explícita da organização que possa ser lida pelos próprios agentes, servindo de fonte nos seus processos cognitivos. A organização social dos agentes define como eles interagem, hierarquias, grupos e aptidões (Hübner e Sichman, 2003).

Com base nos quatro aspectos de um Sistema Multiagente, pode-se definir seu ciclo de vida em duas etapas: a concepção e a resolução (Hübner e Sichman, 2003). Na concepção são definidos os modelos de propósitos gerais para os agentes, para suas interações e para as formas de organização. Na resolução, um grupo de agentes adota esses modelos para resolver os problemas que lhe são apresentados no ambiente de execução. Diferentes tipos de problemas demandam dos agentes diferentes escolhas de modelos. A principal característica é a independência entre a concepção dos modelos e o problema, isto é, os modelos não são desenvolvidos para solucionar um problema específico.

Levando em consideração o ponto de vista de desenvolvimento de modelo, um Sistema Multiagente pode ser definido por modelos genéricos de agentes, interação e organização, que possam ser instanciados dinamicamente em um ambiente de execução para solucionar os problemas que surgem.

A seguir estão quatro características para o desenvolvimento de Sistemas Multiagentes (Alvares e Sichman, 1997):

1. Os agentes são concebidos independentemente de um problema particular;
2. A interação entre os agentes não é projetada anteriormente, busca-se definir protocolos que possam ser utilizados em situações genéricas;
3. A decomposição de tarefas para solucionar um dado problema pode ser feita pelos próprios agentes;
4. Não existe um controle centralizado da resolução do problema.

### 3.1.1 Modelos Organizacionais de Agentes

Pode-se definir organização de agentes como a maneira em que estes se organizam para executar ações com o objetivo de levar o ambiente a um estado satisfatório a todos. Lemaître e Excelente (1998) relatam a existência de duas abordagens para a compreensão de como é a organização de um grupo de agentes: centrada nos agentes e centrada na própria organização.

Nas organizações centradas nos agentes, os agentes não possuem uma representação explícita da organização. Essa representação está distribuída no código dos agentes e um observador externo (ou outro agente) pode apenas inferir de forma subjetiva a organização por meio de observação do comportamento dos agentes. Esta descrição é chamada de organização observada. Pode-se citar como exemplo, a organização de formigas em um formigueiro. Cada indivíduo que observar um formigueiro formará uma opinião diferente sobre a organização, já que não existe uma descrição única da sua real organização, pois está distribuída e implícita no DNA das várias formigas do formigueiro.

Na abordagem centrada na organização, há uma representação explícita da organização e o observador ou um agente pode obter informação formal de como o sistema está organizado. Esta descrição é chamada de sociedade institucionalizada. Pode-se utilizar como exemplo um time de futebol, que recebe as instruções vindas do técnico e os jogadores devem cooperar mutuamente de

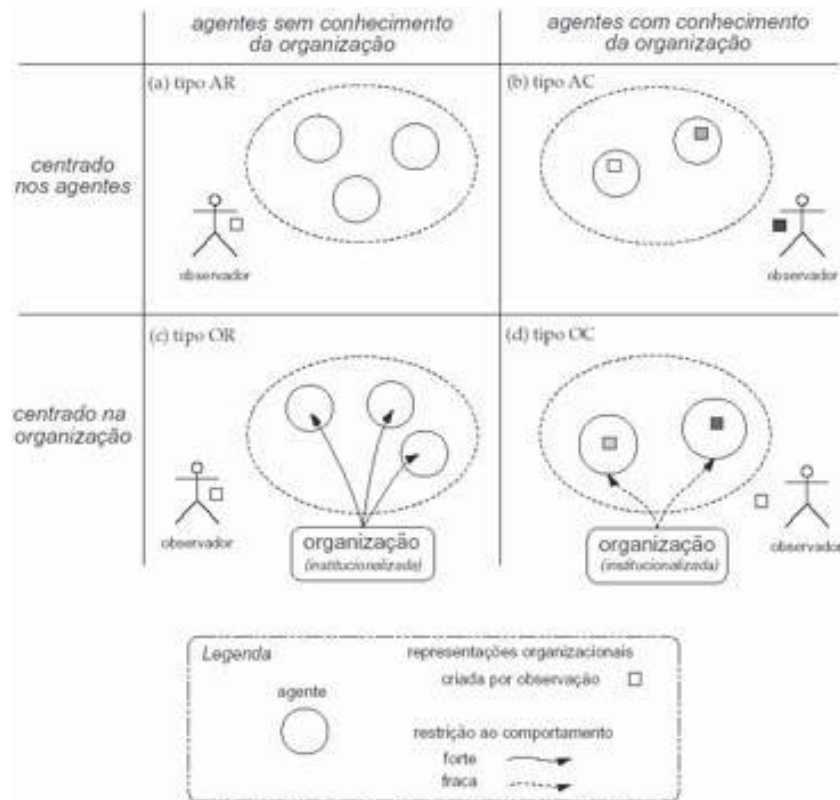


Figura 3.1: Tipos de organizações de agentes (Hübner e Sichman, 2003)

acordo com as instruções. Neste caso, os observadores conseguem representá-la explicitamente, sem precisar observar seu comportamento ou considerar os agentes que a compõem.

Em Hübner e Sichman (2003), é proposta uma sub-divisão nas duas abordagens citadas anteriormente. Em uma delas, os agentes são capazes de representar subjetivamente a organização na qual eles estão inseridos e na outra os agentes não são capazes de tal representação. Assim, são duas abordagens centradas no agente e duas centradas na organização.

A Figura 3.1 apresenta as quatro abordagens segundo Hübner e Sichman (2003), sendo que em (a) apenas o observador consegue modelar o comportamento do sistema de forma subjetiva, em (b) a organização é construída pelos próprios agentes de forma subjetiva, cada agente pode construir uma representação organizacional diferente (na figura, os quadrados preenchidos ilustram as diferentes representações subjetivas). Em (c), embora o sistema tenha uma organização, os agentes são incapazes de conhecê-la. Já em (d), os agentes podem conhecer a organização tanto de forma subjetiva (cada agente terá uma visão da organização) ou adquirindo informações formais da organização (institucionalizada).

A seguir, uma descrição detalhada de cada abordagem:

- **Tipo AR:** as organizações tipo AR são concebidas com uma visão centrada nos agentes. Os agentes não representam e não raciocinam sobre a organização observada. Eles são incapazes de gerar uma representação interna de sua organização. Nesse ambiente os agentes possuem comportamentos elementares e a partir deles, por meio da interação entre os agentes e o ambiente, podem surgir comportamentos complexos. Evidentemente, nesse ambiente os agentes são incapazes de alterar sua organização, já que não a conhecem. Como exemplo pode-se citar a organização de um formigueiro;



- **Tipo AC:** as organizações tipo AC também são concebidas com uma visão centrada nos agente e nesse modelo os agentes representam e raciocinam sobre a organização observada a partir de sua percepção ou comunicação, embora não exista uma representação explícita. Nesse ambiente, os agentes são incapazes de alterar sua organização. Como exemplo, citam-se algumas empresas, que embora a organização não tenha uma representação explícita, os funcionários conseguem de maneira subjetiva raciocinar sobre a organização;
- **Tipo OR:** são concebidas com uma visão centrada na organização e os agentes não são capazes de representar internamente a sua organização de modo explícito, apesar da sociedade possuir uma descrição. Esse tipo de organização estabelece uma restrição forte sobre os agentes, já que eles não têm permissão para agir de forma diferenciada ao comportamento estabelecido pela organização, nem mesmo tirar proveito desse conhecimento para a realização das tarefas. Nesse modelo, os agentes não podem mudar a organização. Cita-se como exemplo algumas redes de espionagem, nas quais, embora haja uma organização, os espões são incapazes de representá-las;
- **Tipo OC:** concebidas com uma visão centrada na organização, os agentes têm capacidade de representar internamente a organização institucionalizada, possuem seu comportamento parcialmente determinado pela organização, percebem como esta interfere internamente no seu comportamento e no comportamento dos demais agentes do sistema. Os agentes são capazes também de utilizar essas informações para melhorar o seu funcionamento. O tipo de restrição imposta por esse tipo de organização é chamada de restrição fraca, pois como os agentes conhecem a organização institucionalizada podem fugir do comportamento que a organização estabelece para promover os objetivos. Por exemplo, em um time de futebol, o zagueiro pode sair de sua posição original e participar de uma jogada no ataque.

Deve-se também levar em consideração as características de autonomia dos agentes dentro das organizações, sendo chamada autonomia organizacional a capacidade do agente de agir de forma diferente ao pré-estabelecido. Para organizações do tipo OC, a autonomia é uma possibilidade. Para organizações do tipo OR não existe autonomia organizacional, já que os agentes são inteiramente dependentes do comportamento estabelecido pela organização e não conseguem modificá-la. Já nas organizações centradas nos agentes (AR e AC) não pode ser definida autonomia organizacional, já que não existem restrições organizacionais.

O modelo MOISE+ desenvolvido por Hübner e Sichman (2003), será utilizado para representar a organização de agentes na presente tese e é detalhado na Seção 4.1. Este é um modelo centrado na organização, distribuído, onde todos os agentes possuem explicitamente uma representação da organização.

### 3.2 AGENTES DE SOFTWARE COGNITIVO

Dentre os segmentos da Inteligência Artificial, alguns são dedicados ao estudo de arquiteturas de agentes cognitivos baseados em estados mentais. As arquiteturas BDI caracterizam-se por utilizarem os estados mentais de crenças, desejos e intenções.

Os estados mentais humanos possuem vínculo com o ambiente, estabelecendo uma relação de existência e significância. Para o agente, os estados mentais indicam uma propriedade de direcionamento do mundo para o agente e vice-versa. Como exemplo, a afirmação "a porta está fechada" é uma crença sobre a porta, do mundo para o agente e "entrar na sala" é um desejo, do

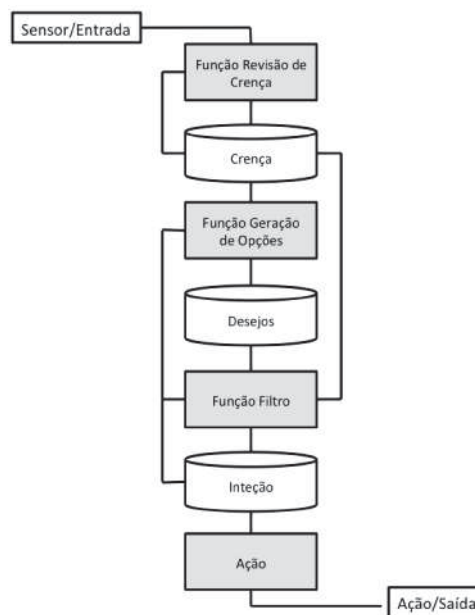


Figura 3.2: Componentes da arquitetura BDI (de Oliveira Zamberlam e Giraffa, 2001)

agente para o mundo. As intenções são um sub-conjunto de desejos, que podem ser consideradas as ações que o agente deve executar para satisfazer um desejo.

Os agentes tratam as crenças como verdades sobre o ambiente, utilizando-as para expressar suas expectativas sobre possíveis estados futuros. Um agente pode ter crenças sobre o mundo, sobre outros agentes, sobre interações com outros agentes e crenças sobre suas próprias crenças, inclusive, as crenças podem ser contraditórias.

Os desejos, ao contrário dos estados representados pelas crenças, representam estados desejáveis que o sistema pode apresentar.

As intenções são um subconjunto dos desejos. Se um agente decide seguir uma meta específica, então essa meta torna-se uma intenção. São as intenções que determinam o processo de raciocínio prático, definindo as ações que serão realizadas. Uma vez adotada uma intenção haverá um direcionamento no raciocínio prático futuro, ou seja, enquanto se tem uma intenção específica, somente serão consideradas as ações que são consistentes para a realização desta intenção.

Sendo assim, no modelo BDI as decisões são logicamente restritas pelas crenças dos agentes, os quais representam o ambiente, o estado mental dos outros agentes e suas capacidades. Com base nesses estados mentais, um agente pode possuir desejos. Esses desejos podem depender exclusivamente do agente em questão, de alguns ou de todos os agentes que fazem parte de uma sociedade. Um sub-conjunto de desejos, as intenções do agente, pode traçar uma meta para que o desejo seja satisfeito.

Ao projetar um agente baseando-se no modelo BDI, são especificadas suas crenças e seus desejos, mas a escolha das intenções fica sob responsabilidade do próprio agente, isto é, de uma auto-análise desses estados inicialmente disponíveis.

Em de Oliveira Zamberlam e Giraffa (2001) são definidos sete componentes importantes de uma arquitetura BDI. Os componentes são descritos na sequência e representados na Figura 3.2:

- Um conjunto de crenças atuais, que representam as informações que o agente tem sobre seu ambiente atual;

- Uma função de revisão de crenças, que a partir da entrada percebida e com as crenças atuais do agente, determina um novo conjunto de crenças;
- Uma função geradora de opções, que determina as opções disponíveis para o agente, ou seja, seus desejos, tendo como base suas crenças atuais sobre seu ambiente e suas intenções atuais;
- Um conjunto de desejos atuais, representando possíveis cursos de ações disponíveis para o agente;
- Uma função filtro, que representa o processo de deliberação do agente, e que determina as intenções dos agentes, tendo como base suas crenças, desejos e intenções atuais;
- Um conjunto de intenções atuais, representando o foco atual do agente;
- Uma função de seleção, que determina uma ação para executar, tendo como base as intenções atuais.

Dentre as linguagens de programação que utilizam a arquitetura BDI, uma merece atenção especial: a linguagem AgentSpeak(L), primeiramente apresentada por Rao (1996). Essa linguagem foi projetada para a programação de agentes BDI na forma de planejamento reativo (*reactive planning systems*). Esses sistemas são aplicativos em execução permanente, respondendo a eventos que ocorrem no ambiente que eles estão situados, por meio de execução de planos que se encontram em uma biblioteca. Para a presente tese foi utilizada a linguagem JASON, detalhada na Seção 4.2, que permite desenvolver agentes cognitivos baseados em BDI por meio da linguagem AgentSpeak.

### 3.2.1 Linguagem de Programação BDI - AgentSpeak(L)

A linguagem de programação AgentSpeak(L) é uma extensão natural da programação em lógica, porém focada na arquitetura de agentes BDI, representando um modelo abstrato para a programação de agentes (Wooldridge, 1997).

A linguagem possui uma sintaxe semelhante ao paradigma de programação em lógica, tal como a aplicada no ambiente PROLOG. Assim, um agente codificado em AgentSpeak(L) corresponde a um conjunto de crenças e planos que formarão a base de conhecimento inicial. Uma unidade de crença corresponde a um predicado de primeira ordem na programação em lógica e as literais de crenças são unidades de crenças ou suas negações.

Em Hübner et al. (2004) são apresentados dois tipos de objetivos para um agente em AgentSpeak(L): objetivos de realização (*achievement goals*) e objetivos de teste (*test goals*). Objetivos de realização e de teste são predicados da programação em lógica tradicional, tais como crenças, porém com operadores prefixados '!' e '?', respectivamente.

Objetivos de realização expressam que o agente quer alcançar um estado no ambiente, sendo o predicado associado ao objetivo verdadeiro. Na prática, esses objetivos iniciam a execução de sub-planos. Um objetivo de teste retorna a unificação do predicado de teste com uma crença do agente, ou falha caso não seja possível a unificação com nenhuma crença do agente. Um evento ativador (*triggering event*) define quais eventos podem iniciar a execução de um plano. Um evento pode ser interno, quando gerado pela execução de um plano em que um subobjetivo precisa ser alcançado, ou externo, quando gerado pelas atualizações de crenças que resultam da percepção do ambiente. Eventos ativadores são relacionados com a adição e a remoção de atitudes mentais (crenças ou objetivos) que são representadas pelos operadores prefixados ('+') e ('-') (Hübner et al., 2004).

Planos fazem referência a ações básicas que um agente é capaz de executar em seu ambiente. Em Hübner et al. (2004), essas ações são definidas por predicados com símbolos predicativos especiais (chamados símbolos de ação) usados para distinguir ações de outros predicados. Um plano é formado por um evento ativador (denotando o propósito do plano), seguido de uma conjunção de literais de crença representando um contexto. O contexto deve ser consequência lógica do conjunto de crenças do agente no momento em que o evento é selecionado pelo agente para o plano ser considerado aplicável. O restante é uma sequência de ações básicas ou subobjetivos que o agente deve atingir ou testar quando uma instância do plano é selecionada para execução.

Duas estruturas importantes para o interpretador são o conjunto de eventos e o conjunto de intenções. Intenções são ações com as quais um agente se compromete a executar quando um evento é chamado. Eventos causam o início da execução de planos que tem eventos ativadores correspondentes. Eles podem ser externos, quando originados da percepção do ambiente (exemplo: a adição ou a remoção de crenças); ou internos, quando chamados por meio da execução de planos do agente (um subobjetivo em um plano gera um evento do tipo "adição de objetivo de realização"). Eventos externos criam novas intenções representando diferentes focos de atenção na atuação do agente no ambiente (Hübner et al., 2004).

Como a linguagem AgentSpeak(L) possui uma grande quantidade de ações pré-definidas em sua biblioteca, conhecidas como ações internas, são identificadas por possuírem o caractere de ponto (.) no início de seu nome. Elas são executadas internamente pelos agentes e não afetam o ambiente como ocorre com as ações definidas pelos programadores da linguagem. Entre as ações internas destacam-se comandos para imprimir conteúdo na console e para troca de mensagens entre agentes.

### 3.3 NEGOCIAÇÃO ENTRE AGENTES UTILIZANDO TÉCNICA DE LEILÃO

Leilão é uma maneira simples de executar alocação de recursos em Sistemas Multiagentes. Em um leilão, um agente pode expressar quanto deseja pagar por um recurso específico por meio de um lance que, por sua vez, é processado pelo agente que oferece o recurso ou por um agente central, chamado de leiloeiro.

No primeiro caso, não existe um elemento centralizador no sistema, o que evita gargalos, entretanto o agente que necessita do recurso pode não conhecer a localização do agente que o ofereça, o que dificulta o processo de negociação.

No segundo caso, com o agente leiloeiro, todos os recursos disponíveis para leilão no sistema são conhecidos por esse agente, o que facilita sua localização, entretanto o sistema fica dependente desse elemento central.

Em ambos os casos, todos os lances são processados pelo agente leiloeiro ou o agente que oferece os recursos, e quem oferecer o maior lance terá o direito de utilizá-lo. O processo para escolher o melhor lance depende do tipo de leilão utilizado. A seguir, são apresentados os principais tipos de leilões (Vidal, 2006):

- **Inglês:** os participantes dão lances crescentes para um determinado item, até que nenhum participante esteja disposto a dar um lance maior que o atual. É provavelmente o tipo mais comum de leilão;
- **Holandês:** o leiloeiro fixa um preço inicial alto e vai progressivamente diminuindo tal preço, até que algum dos participantes esteja disposto a comprar o item por aquele preço. É assim chamado devido ao fato de ser utilizado por vendedores de flores na Holanda;

- **De Lance Fechado e Primeiro Preço:** os participantes entregam envelopes lacrados com seu respectivo lance e cada participante tem direito a apresentar apenas um lance. Os envelopes são abertos ao mesmo tempo, sendo que o maior lance é declarado vencedor;
- **Vickrey:** semelhante ao leilão de lances fechados e primeiro preço, com a diferença de que o valor pago pelo vencedor é o segundo maior preço dado no leilão. Por isso, também é chamado de leilão de segundo preço;
- **Reverso:** conhecido como *procurement auctions* ou *reverse auctions*, esse leilão é utilizado quando existe um único comprador e vários possíveis vendedores, sendo que nesse leilão cabe ao vendedor o trabalho de submeter lances. O leilão reverso não é formalmente diferente do leilão inglês, a diferença é que nesse vence o menor e não o maior preço. Por isso, qualquer estratégia definida para o leilão inglês vale também para o leilão reverso.

Quanto ao valor a ser pago por um recurso, deve ser levado em consideração a avaliação subjetiva do recurso pelo agente e ao recurso pode ser atribuído um valor privado, comum ou correlato.

Quando um agente necessita utilizar imediatamente um recurso, é provável que ele atribua a esse recurso um valor privado. Como exemplo, pode-se citar uma pessoa com fome em um restaurante, ele pagará por uma refeição um valor proporcional a sua fome e a quanto possui de dinheiro.

A um recurso também pode ser atribuído um valor comum, sendo esse o valor inicial. Um exemplo clássico são as lojas, elas compram mercadorias em grandes quantidades para revender e, nessa situação, não utilizarão o que estão adquirindo pois o produto será revendido. Logo, o valor que elas oferecem pelo produto no leilão é proporcional a quanto irão lucrar com o produto na venda.

Já o valor correlato, cita-se como exemplo a compra de uma casa. Nesses casos, na maioria das vezes, não se paga o preço de custo da casa e, sim, um valor agregado pela sua localização e o valor que poderia ser recebido em uma venda futura. Essa é uma situação muito comum no mundo real.

Para a implementação de leilões em sistemas computacionais, existe um protocolo definido pela FIPA (2002) chamado FIPA Contract-Net, apresentado em detalhes na seção 4.3. O tipo de leilão adotado neste modelo foi o De Lance Fechado e Primeiro Preço, o qual minimiza a quantidade de mensagens trocadas, uma vez que é tratado apenas o primeiro lance.

### 3.4 COMPLEXIDADE COMPUTACIONAL DE ALGORITMOS DE OTIMIZAÇÃO

Considerando os problemas computacionais que podem ser resolvidos por meio de algoritmos, existem os problemas chamados de polinomiais, ou problemas da classe P, que são aqueles que possuem um resultado exato e podem ser resolvidos em um tempo computacional aceitável. São exemplos deste problema a localização de dados em um vetor ou a ordenação dos seus elementos.

Porém, existe um conjunto muito maior de problemas que não estão contidos em P, chamados de problemas Não-Polinomiais, ou NP, que não possuem uma solução exata e a única maneira de resolver estes problemas é explorando todas as possibilidades possíveis de solução, fazendo um processamento exaustivo, o que é muito custoso do ponto de vista de processamento. Estes são problemas difíceis de serem resolvidos, em especial, em instâncias

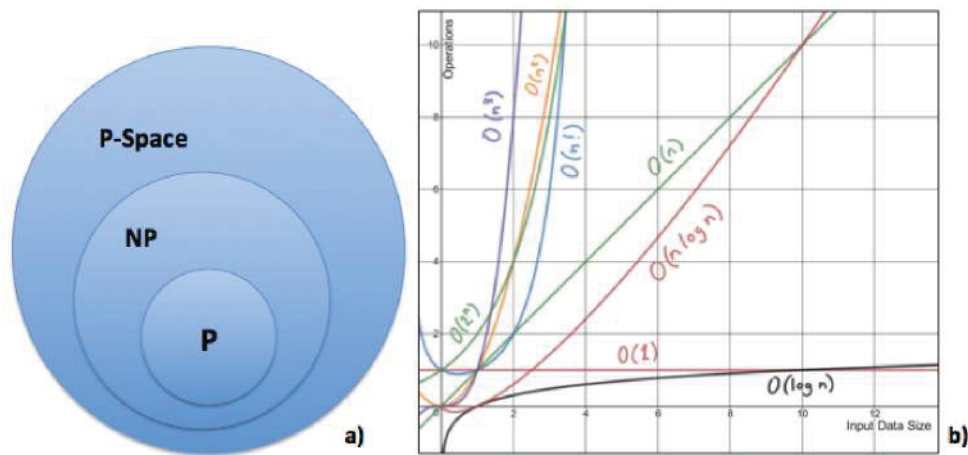


Figura 3.3: a) Classes de problemas existente, com os problemas P contidos em NP, que por sua vez estão contidos em P-Space. b) Gráfico da complexidade dos algoritmos, quanto mais próximo do eixo y, mais custoso é o processamento

com muitas informações. Como exemplo cita-se o tradicional problema do Caixeiro Viajante (Wikipedia, 2019c).

Mas, ainda mais complexo do que os problemas do tipo NP, são os problemas do tipo P-Space, no qual a presente tese se insere. Neste tipo de problema, além da complexidade no processamento para se encontrar uma solução ótima, tem-se também a limitação da memória para que esta solução seja encontrada.

Segundo Kleinberg e Tardos (2005), problemas de decisões do tipo P devem ser resolvidos em tempo polinomial. Já em P-Space, os problemas de decisões devem ser resolvidos em espaço polinomial. Como exemplo de P-Space tem-se o Problemas de Localização de Instalações, ou *Facility Location Problem* (Wikipedia, 2019a). Este estuda a alocação ideal de depósitos para minimizar os custos de transporte de produtos, considerando para isso fatores como evitar a colocação de materiais perigosos próximos a residências, ou depósitos próximos a concorrentes do produto.

Na Figura 3.3.a. é possível verificar as classes dos problemas existente, com os problemas do tipo P contidos na classes NP, assim como os NP contidos em P-Space, sendo este último o de maior complexidade de acordo com a representação da figura.

Para resolver estes problemas existem algoritmos que são avaliados por sua eficiência na resolução. Como é difícil fazer com que todos os algoritmos sejam executados em uma mesma máquina para uma comparação justa, sua eficiência é analisada de forma assintótica, ou seja, a eficiência para resolver problemas à medida que a instância de entrada aumenta, sendo esta a quantidade de informações a ser manipulada.

Por exemplo, um algoritmo de pesquisa sequencial, buscando por um elemento dentro de um vetor, possui uma complexidade  $O(n)$ , este presente no gráfico da Figura 3.3.b. Esta complexidade é linear e à medida que se acrescentam elementos em um vetor (eixo x), aumenta de forma proporcional o esforço computacional para executar as operações de pesquisa (eixo y). Assim, independentemente da máquina ou da instância em que este algoritmo é executado, sua complexidade será sempre a mesma. Tem-se outros exemplos, como a busca binária, com complexidade  $O(\log n)$ , o que significa que a medida que se aumenta a quantidade de elementos no vetor, o custo das operações não aumentam na mesma proporção. Já algoritmos de ordenação de vetores, por exemplo, possuem complexidade maior, como o algoritmo da bolha,

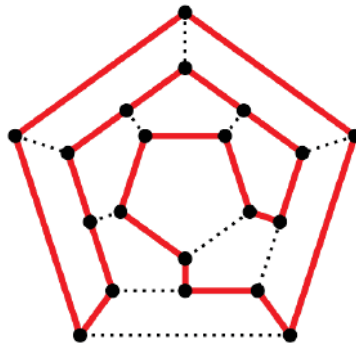


Figura 3.4: Exemplo de um circuito Hamiltoniano (Wikipedia, 2019b)

com complexidade  $O(n^2)$  e o algoritmo de ordenação Merge Sort que possui complexidade  $n \log n$ .

No gráfico da Figura 3.3.b, vê-se que os algoritmos mais eficientes tendem a se aproximar do eixo x, já os algoritmos menos eficientes, se aproximam do eixo y.

Todos os algoritmos apresentados nos parágrafos anteriores resolvem problema polinomiais, ou seja, problemas do tipo P, encontrando soluções exatas em um tempo computacional aceitável, o que é bom. Entretanto, os problemas da classe NP só possuem resultados ótimos se os algoritmos forem executados de forma exaustiva e estes algoritmos costumam ter um custo de processamento para encontrar uma solução muito próximos do eixo y.

Um exemplo de problema da classe NP é o de circuito Hamiltoniano (Wikipedia, 2019b). Este problema é caracterizado por um grafo e o objetivo é saber se é possível percorrer todos os pontos deste grafo, passando por cada ponto apenas uma vez e retornando ao ponto inicial - Figura 3.4. Para que isso seja possível, é necessário testar todas as combinações possíveis de caminhos, o que torna a complexidade deste algoritmo fatorial -  $O(n!)$ . Com um cenário pequeno com poucos pontos é possível encontrar soluções ótimas em um tempo computacional aceitável, já quando aumenta o número de pontos, a complexidade computacional aumenta, por isso, o custo computacional para encontrar soluções neste tipo de problema é muito próximo do eixo y, como pode ser observado na Figura 3.3.

Entretanto, este ainda não é o pior dos problemas computacionais em termos de complexidade. Se considerarmos que para resolver o problema do Caixeiro Viajante é necessário encontrar todos os circuitos Hamiltoniano do grafo, computar o custo de cada caminho, compará-los para só então saber o menor caminho, pode-se dizer que a complexidade computacional do Caixeiro Viajante é maior do que a complexidade computacional da resolução do problema do circuito Hamiltoniano.

E por fim, no problema do tipo P-Space, como o planejamento de voo para grupo de Drones de forma otimizada, que é a abordagem utilizada por esta Tese, a complexidade é ainda maior, pois existem muitos Drones, pontos de passagem, bases de recarga. Além disso, existe a dinamicidade do ambiente, como a mudança do custo para visitar cada ponto, que ocorre na mudança da velocidade ou direção do vento, além de eventos adversos, como o término precoce da bateria de um Drones. Pode-se considerar este como um problema complexo de alocação de tarefas.

Assim, apesar da complexidade e das características do presente problema, o algoritmo de otimização utilizado na proposta desta tese deve ser rápido e dinâmico, gerando planos de voos para os Drones de forma rápida, mesmo que estes não sejam os melhores trajetos possíveis, mas que sejam soluções próximas da ótima, uma vez que para cada mudança de ambiente novos planos de voos devem ser gerados e enviados para os Drones.

Ao contrário dos algoritmos com execução exaustiva utilizados para resolver o caixeiro viajante, por exemplo, que podem ser executados por muitas horas para instâncias não muito grande, aqui tem-se a necessidade de um algoritmo que seja executado em segundos mesmo para instâncias muito grandes, algo que é muito comum quando se precisa de otimização em cenários que envolvem Internet das coisas ou agricultura de precisão, por exemplo.

Para se obter resultados factíveis, não necessariamente o melhor de todos, mas aceitáveis com pequeno custo computacional é comum a utilização de heurísticas e meta-heurísticas para resolver problemas do tipo NP ou P-SPACE.

Um exemplo de heurística utilizada para resolver o problema do Caixeiro Viajante, com um custo computacional muito pequeno se comparado aos algoritmos tradicionais que encontram a solução ótima, é o Algoritmo Guloso (Cormen, 2009).

Esta heurística define que sempre que o caixeiro precisar escolher uma cidade, a decisão será a cidade com menor custo, sendo executada até que todas as cidades sejam visitadas. Esta heurística não garante a solução ótima, mas garante uma única solução factível em um tempo computacional aceitável, porém, se o algoritmo for executado inúmeras vezes, todas elas retornarão a mesma resposta, já que se trata de um algoritmo determinístico.

Visando melhorar os resultados obtidos por heurísticas determinísticas, como o algoritmo guloso, utiliza-se meta-heurísticas, as quais possuem parâmetros aleatórios, fazendo que a cada execução, por exemplo, tem-se resultados diferentes, sendo escolhido ao final o melhor resultado de todos. Nas meta-heurísticas, os algoritmos são executados até que uma condição de parada seja alcançada, podendo ser um número especificado de ciclos de execução ou um tempo limite.

Uma meta-heurística conhecida para resolver o problema do caixeiro viajante é o GRASP (*Greedy Randomized Adaptive Search Procedure*) (Feo e Resende, 1995). Ele é baseado no algoritmo guloso, porém, a cada cidade o algoritmo escolhe as  $n$  cidades mais próximas, sendo  $n$  definido pelo programador, sorteando entre elas (este é o parâmetro aleatório do algoritmo) uma das cidades mais próximas para visita. Se o algoritmo for executado várias vezes, provavelmente todos os resultados serão diferentes, sendo escolhido ao final o melhor resultado para ser apresentado.

Desta forma, a presente tese propõe um algoritmo para encontrar soluções rápidas para o problema de otimização de voo para grupos de Drones, que é um problema do tipo P-Space. Como não é um algoritmo guloso, o algoritmo proposto tem complexidade computacional alta (próximo do eixo  $y$ ) e sendo executado de forma exaustiva garante o encontro do ótimo global utilizando muito recursos como processamento e memória, o que torna desproporcional a execução exaustiva para cenários dinâmicos.

A este algoritmo podem ser aplicadas heurísticas e meta-heurísticas que permitem encontrar soluções factíveis em tempo de processamento aceitável, considerando que para o cenário de agricultura de precisão, no qual mudanças no cenário acontecem de forma muito rápida, é necessário um algoritmo que se adapte muito rápido a esta mudança, otimizando da mesma forma o voo, porém, não focando no resultado ótimo, mas em um resultado factível dado o tempo que se tem entre uma mudança no ambiente e outra.

### 3.5 CONCLUSÃO

Este capítulo fundamenta a escolha e a validação teórica das técnicas utilizadas para resolver o problema de abertura, dinamicidade e otimização do MAOVGD.

Em um primeiro momento, técnicas e tipos de modelagem utilizando Sistema Multiagentes são apresentadas, centrando na modelagem baseada nas organizações de agentes. Nela, todo o Sistema Multiagente é expresso por documentos e as informações são acessíveis por todos



os agentes. Assim, um agente inserido no sistema dinamicamente consegue entrar já assumindo um papel, o que garante a abertura necessária.

Para a dinamicidade é necessário que todos os agentes conheçam informações como crenças, desejos e intenções, que podem ser modificadas ao longo do tempo de processamento e por este motivo a utilização de agentes cognitivos baseados no modelo BDI atendem as necessidades do modelo proposto por esta Tese.

Para a troca de informações entre os agentes, independente da dinâmica do modelo, a técnica escolhida foi a técnica de leilão, já que o modelo proposto se assemelha à um leilão tradicional.

Por fim, foi apresentado um estudo sobre complexidade computacional, para identificar a inserção neste contexto do problema de otimização proposto e, assim, ter parâmetros para sua avaliação.

## 4 TECNOLOGIAS E MATERIAIS UTILIZADOS

Neste capítulo são apresentadas as ferramentas que serão utilizadas para o desenvolvimento do modelo proposto. Inicialmente o capítulo apresenta o MOISE+, este utilizado para organização dos agentes no Sistema Multiagente proposto. Em seguida é apresentado a linguagem JASON para a codificação dos agente cognitivos utilizando os princípios do BDI. Após, é detalhado o protocolo FIPA Contract-Net para o modelo de negociação entre os agentes baseado em leilões e, por fim, a técnica de leilão recursivo para otimização do modelo. Uma visão geral de onde cada ferramenta é utilizado é apresentada na Figura 4.1. Ao final do capítulo são apresentados os equipamentos para realização dos testes do modelo proposto.

### 4.1 MOISE+ PARA MODELAGEM DO SISTEMA MULTIAGENTE

O modelo MOISE+ apresenta uma visão centrada nas organizações, do tipo OC. Neste, a organização existe e é institucionalizada com uma representação explícita e os agentes têm capacidade de raciocinar sobre tal representação. Como consequência da visão centrada na organização, o modelo MOISE+ possui uma Especificação Organizacional (EO), que é um documento XML (*eXtensible Markup Language*) que deve ser respeitado por todos os agentes que pertencem à organização. A Especificação Organizacional é subdividida em três dimensões: Especificação Estrutural (EE); Especificação Funcional (EF) e Especificação Deontica (ED).

#### 4.1.1 Especificação Estrutural

A Especificação Estrutural se preocupa com os aspectos que devem ser respeitados pelos agentes na organização, tais como a estrutura da organização e as atividades que serão distribuídas para os agentes. Nessa especificação encontram-se as definições dos grupos, as atribuições de papéis aos grupos e as relações entre os papéis (autoridade, comunicação, conhecimento sobre outro papel e compatibilidade).

No modelo de organização proposto pelo MOISE+, o conceito de papel tem função primordial por ser o elo entre o agente e a organização. Os papéis somente são assumidos no nível coletivo, isto é, dentro de um grupo. Intuitivamente, um grupo representa um conjunto de agentes com afinidades maiores e objetivos mais próximos.

#### 4.1.2 Especificação Funcional

Na Especificação Funcional são definidas as metas globais do Sistema Multiagente, também chamadas de esquema. Um esquema é formado por um conjunto de metas locais.

Uma meta global representa um estado do mundo que é desejado pelo Sistema Multiagente. Uma meta global diferencia-se de uma meta local pelo fato desta última ser uma meta de um único agente, enquanto a primeira é uma meta de todo o Sistema Multiagente. Como as metas são organizadas em uma hierarquia, uma meta global (nível hierárquico maior) é decomposta em metas locais (nível hierárquico menor). As metas locais podem ser: sequenciais (uma é realizada após a outra e a meta global é satisfeita quando todas as metas locais forem realizadas), paralelas (podem ser realizadas ao mesmo tempo, mas a meta global só é satisfeita se todas as metas locais forem realizadas) ou escolha (idêntica a anterior, porém a meta global é satisfeita se pelo menos uma meta local for realizada).

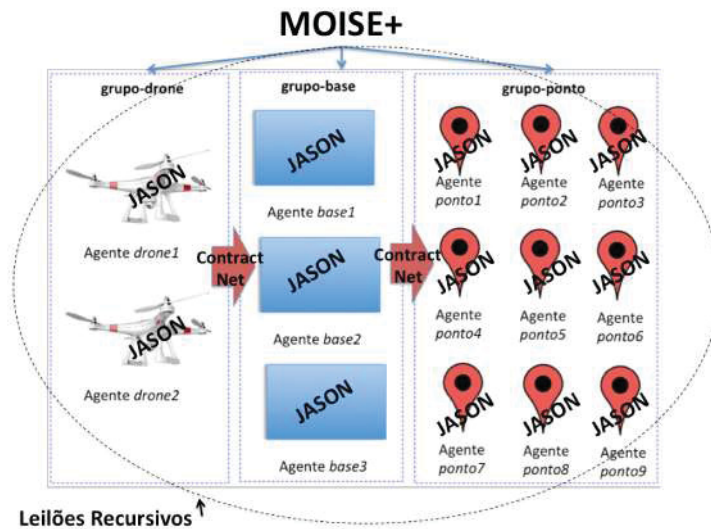


Figura 4.1: Visão geral do modelo proposto com as ferramentas sugeridas para o desenvolvimento

Uma meta possui um valor associado ao nível de satisfação, que indica se a meta já foi realizada (valor *satisfied*) ou não (valor *unsatisfied*), ou ainda se ela é impossível de ser iniciada em dado momento (valor *impossible*). Uma meta é impossível, por exemplo, quando em uma sequência de metas, a meta anterior a ela ainda não foi realizada.

#### 4.1.3 Especificação Deôntica

A relação entre a Especificação Estrutural e a Especificação Funcional é feita pela Especificação Deôntica em nível individual. Nela são especificadas quais metas locais um papel tem permissão ou obrigação de se comprometer.

Uma permissão determina que um agente com um papel pode se comprometer com um objetivo local específico. Restrições temporais são estabelecidas para a permissão, isto é, se estabelece um conjunto de períodos de tempo no qual a permissão é válida.

Uma obrigação estabelece que um agente com papel específico deve se comprometer com um objetivo local em um período de tempo pré-determinado.

Na Seção 5.1 é apresentada a organização do Sistema Multiagente proposto institucionalizada pela ferramenta MOISE+ para a presente tese.

## 4.2 JASON PARA O DESENVOLVIMENTO DE AGENTES COGNITIVOS

A ferramenta JASON é um interpretador para uma extensão da linguagem AgentSpeak(L), com algumas melhorias, incluindo a comunicação entre agentes baseada na teoria de atos de fala (Austin, 1975) (Searle e Vanderveken, 1985).

Utiliza-se o SACI (Hübner e Sichman, 2000) para a comunicação entre agentes remotos, sendo assim, um Sistema Multiagente desenvolvido com o JASON pode ser distribuído em uma rede de computadores sem muito esforço.

Uma característica importante do JASON em comparação aos outros sistemas BDI é que ele é implementado em Java e, com isso, é multi-plataforma, sendo disponível como *Open Source* sob a licença GNU LGPL (*Lesser General Public License*). É, ainda, disponibilizada uma interface padrão para o desenvolvimento de novas funcionalidades, sendo facilmente personalizado e expandido.

Além de interpretar a linguagem AgentSpeak(L) original, o JASON possui os seguintes recursos (Hübner et al., 2004):

- Tratamento de falhas em planos;
- Comunicação baseada em atos de fala (incluindo informações de fontes como anotações de crenças);
- Anotações em identificadores de planos, que podem ser utilizadas na elaboração de funções personalizadas para seleção de planos;
- Suporte para o desenvolvimento de ambientes, que normalmente não é programada em AgentSpeak(L). No JASON, o ambiente é programado em Java;
- A possibilidade de executar o Sistema Multiagente distribuídamente em uma rede;
- Possibilidade de especializar (em Java) as funções de seleção de planos, as funções de confiança e toda a arquitetura do agente (percepção, revisão de crenças, comunicação e atuação);
- Possuir uma biblioteca básica de ações internas;
- Possibilitar a extensão da biblioteca de ações internas.

Uma característica importante do JASON é a disponibilização de comandos básicos em uma biblioteca, sendo chamados de ações internas. As ações internas são métodos utilizados com muita frequência, como, por exemplo, o envio de mensagens para outros agentes, escrever uma mensagem na console do JASON (útil no processo de depuração), função para retornar o resultado da soma de dois números, entre outros. As funções internas podem ser utilizadas tanto no contexto como no corpo dos planos e são facilmente identificadas por possuírem um caractere de '.' (ponto) no início ou ao longo da definição do método.

Essas ações são chamadas de internas para distinguir das que são codificadas no corpo de um plano do agente, também chamadas de ações básicas, e que são invocadas ao longo do programa.

Para a definição e a execução de um Sistema Multiagente em JASON, deve-se definir um conjunto de agentes AgentSpeak(L) e um ambiente no qual todos estarão situados. Essas definições devem ser informadas em um arquivo texto com extensão .mas2j.

### 4.3 FIPA CONTRACT-NET PARA NEGOCIAÇÃO POR LEILÕES

O protocolo FIPA Contract-Net define uma sequência de passos para o desenrolar de um leilão em um ambiente computacional. A Figura 4.2 apresenta o fluxo de um leilão utilizando o protocolo FIPA Contract-Net.

O protocolo é simples e se divide em dois momentos. Primeiro, quando o agente Iniciador solicita aos demais agentes participantes propostas para o leilão e o segundo quando o agente Iniciador envia o resultado para o agente ganhador do leilão. O agente Iniciador pode ser um agente leiloeiro ou um agente que possua um recurso para leiloar.

Inicialmente, como pode ser observado no fluxo da Figura 4.2, o agente Iniciador envia uma solicitação de propostas (*call for proposal* - cfp) para os participantes do leilão. Cada participante elabora uma oferta (*propose*) ou envia uma mensagem de recusa (*refuse*).

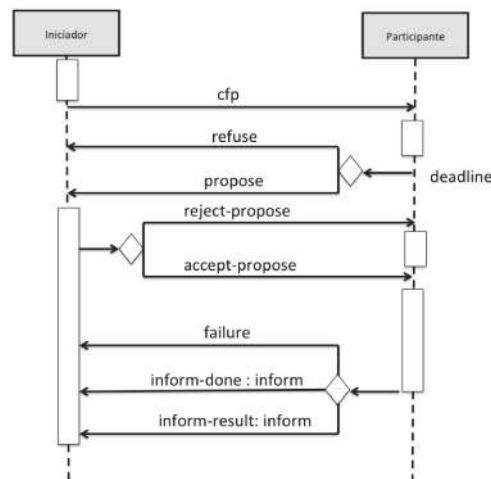


Figura 4.2: Protocolo de interação da FIPA Contract-Net (FIPA, 2002)

Após um período denominado *deadline*, o Iniciador analisará a proposta e selecionará uma para atender o *cfp*. Dessa maneira, o agente que tem a proposta selecionada receberá uma mensagem de *accept-proposal*, enquanto os demais receberão uma mensagem *reject-proposal*.

O agente que teve a proposta selecionada atenderá a solicitação do agente Iniciador e poderá respondê-lo com uma mensagem *failure*, caso ocorra algum erro no processamento da proposta; *inform-done*, para informar que a atividade já foi realizada ou *inform-result*, para enviar o resultado do processamento.

Cada uma dessas técnicas de negociação possui características específicas, permitindo sua utilização em diferentes cenários. Por exemplo, para o problema de alocação de tarefas para diferentes Drones, a técnica de negociação por meio de leilões é a que melhor se adapta, pois os Drones possuem recursos, que é seu tempo de voo, e querem fornecer as bases de decolagem, que por sua vez, tentam vendê-las aos pontos pelos quais o Drone deve passar. Assim, o objetivo desse leilão é fazer com que o Drone percorra a maior quantidade de pontos e que possa retornar para uma base para recarregar sua bateria. Para o problema desta tese, será utilizado o Leilão de Lance Fechado de Primeiro Preço.

#### 4.4 MULTIDRONE SIMULATOR PARA TESTES DA OTIMIZAÇÃO BASEADA EM LEILÕES RECURSIVOS

Para os testes do MAOVDG, foram estudados 8 ambientes de simulação, sendo eles: Paparazzi UAV, JAUS, AETOURNOS TAEMS, Mission Planner, Simulink para Matlab, UAVSim e The Network Simulator. Porém, dadas as especificidades dos testes e visando que o simulador funcionasse para algoritmos desenvolvidos em diferentes linguagens/plataformas, foi optada pelo desenvolvimento de um simulador específico que atendesse da melhor maneira possível as necessidades.

Assim, foi desenvolvido o MultiDrone Simulator (Brito et al., 2020b) (artigo completo no Apêndice A.14. A seguir são apresentados alguns dos recursos disponíveis no MultiDrone Simulator:

- **Definição do tamanho da área a ser sobrevoada.:** é possível definir graficamente as dimensões das áreas que serão sobrevoadas. O tamanho é definido em números de *waypoints*, que são os pontos que devem ser sobrevoados pelos Drones;

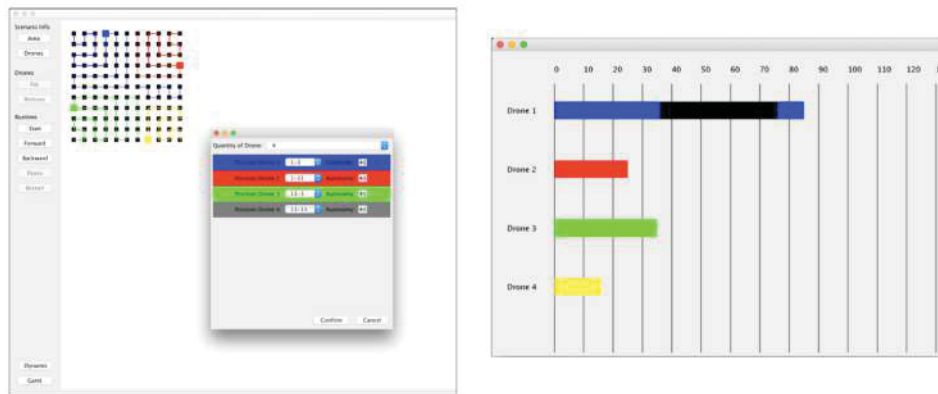


Figura 4.3: Tela do MultiDrone Simulator. A esquerda, simulação de um voo com quatro Drones. A direita, a alocação de recursos em um gráfico de Gantt

- **Definição dos Drones:** possibilidade de adicionar um ou mais Drones para os testes, especificando nome, autonomia, tempo de recarga e sua posição inicial na área a ser sobrevoada;
- **Importar um plano de voo inicial:** o ambiente permite a importação de um plano de voo inicial, gerado por qualquer outra ferramenta ou linguagem, no formato JSON (*JavaScript Object Notation*), sendo que este é executado de forma *off-line* pelo simulador, assim, o usuário não pode interagir com a execução do plano de voo;
- **Visualização da Rota de voo no Simulador:** o simulador, de forma gráfica, apresenta a rota de voo de cada Drone, diferenciando-o um do outro pelas cores das linhas;
- **Execução Interativa:** o usuário pode optar por uma execução interativa, sendo possível visualizar passo a passo e interagir com o ambiente, modificando variáveis, como velocidade e direção do vento, solicitando um retorno precoce para a base de algum Drone e adicionando ou retirando Drones do modelo durante a execução de uma missão. O simulador dispõe de um servidor de Socket que, de forma online, outros programas desenvolvidos em qualquer linguagem podem enviar os comandos de voo, por meio de JSON, assim como receber notificações de mudanças no ambiente, também por meio de JSON;
- **Execução Completa:** com apenas um clique, o usuário pode realizar uma execução completa e, desta forma, não pode interagir com o ambiente, somente visualizar a simulação graficamente;
- **Gráfico de Gantt:** Uma vez que o problema do MAOVGD é uma alocação de tarefa, uma ferramenta adequada para visualizar como cada tarefa está sendo alocada é por meio de um gráfico de Gantt, sendo possível visualizar o voo de cada Drone e o tempo que ele fica na base para recarregar a bateria.

A Figura 4.3 apresenta a tela do MultiDrone Simulator. A esquerda é possível visualizar a execução do plano de voo para 4 Drones, em um cenário com 11 x 11 pontos. A direita o Gráfico de Gantt referente a esta execução.

Este simulador permite testes de otimização do ponto de vista de tempo de processamento, assim como testes de abertura e dinamicidade, sendo este o objetivo principal dos testes do MAOVGD.

#### 4.5 MATERIAIS UTILIZADOS

A preparação do ambiente de execução multiagente, assim como o desenvolvimento e os testes do algoritmo de otimização foram realizados em um laptop Macbook Pro com processador Intel Core I5 Dual com 2,8GHz, memória RAM de 8GB DDR3 1600MHz, com armazenamento em Flash SuperDrive SSD. O sistema operacional instalado no laptop é o MacOS High Sierra versão 10.13.3.

Para o desenvolvimento do simulador e a codificação do Sistema Multiagente e dos Agentes Cognitivos foi utilizada a linguagem de programação Java que é multiplataforma, podendo a execução ocorrer em Windows, Linux e MacOS, entre outros sistemas operacionais.

#### 4.6 CONCLUSÃO

Este capítulo apresentou as características das ferramentas utilizadas para o desenvolvimento do MAOVGD. Todas as ferramentas e tecnologias são *Open-Source*. O MOISE+ permite o desenvolvimento de um sistema Multiagente centrado na organização e a ferramenta JASON permite o desenvolvimento da lógica dos agentes cognitivos baseados em crenças, desejos e intenções. Já o protocolo FIPA Contract-NET pode ser implementado em qualquer linguagem de programação, inclusive utilizando AgentSpeak(L), sendo esta a linguagem utilizada nos agentes cognitivos.

Para os testes, foram estudados simuladores, porém, não foi identificado um que atendesse plenamente as características a serem testadas no modelo proposto que são: abertura, dinamicidade e otimização. Assim, um simulador foi desenvolvido, denominado MultiDrone Simulator, com todas as características que permitiram testar o modelo proposto.

## 5 MODELO PROPOSTO

Este capítulo apresenta a metodologia para o desenvolvimento, teste e validação do MAOVD. A Figura 5.1 apresenta uma visão geral da contribuição dada pela Fundamentação Teórica (Capítulo 3) na definição das técnicas apresentadas em Tecnologias e Materiais Utilizados (Capítulo 4) e a sua aplicação no modelo proposto (Capítulo 5).

O conteúdo deste capítulo, com a proposta do modelo e alguns resultados preliminares foi apresentado no documento de qualificação, em outubro de 2018. Este documento gerou um artigo, uma das publicações mais importantes até o momento (Brito et al., 2019b) - o texto completo está no Apêndice A.8.

Este capítulo apresenta como é realizada a organização dos agentes na ferramenta MOISE+ e a estrutura dos agentes cognitivos desenvolvidos com a ferramenta JASON. A estrutura da troca de mensagens usando o protocolo FIPA Contract-NET também é apresentada, assim como a metodologia utilizada para a otimização utilizando leilões recursivos. O trabalho completo foi submetido para (Brito et al., 2020e) (Artigo completo no Apêndice A.16).

### 5.1 ORGANIZAÇÃO DOS AGENTES PARA O MODELO PROPOSTO

Para o modelo proposto foi utilizada uma abordagem baseada em Sistema Multiagente, que possui foco na organização. O motivo para tal decisão está na abertura do modelo, assim como na garantia de sua dinamicidade. Com um modelo com grupos de agentes bem definidos e com papéis conhecidos por todos os agentes, será possível que, mesmo com a dinamicidade do ambiente no qual acontece o voo de grupo de Drones, o modelo possa continuar em execução, recuperando-se de eventos adversos, tais como término precoce da bateria de algum Drone, aumento da área a ser explorada durante a execução de uma missão, mudança na velocidade e direção do vento, inclusão ou exclusão de Drones, entre outros.

A organização do Sistema Multiagente iniciou com a criação da Especificação Organizacional feita por um arquivo XML na plataforma MOISE+. Nessa especificação, foram definidos três grupos de agentes, sendo eles:

- grupo-drone: agentes que representaram os Drones;
- grupo-base: agentes que representam as bases de recarga;



Figura 5.1: Contribuição do capítulo fundamentação teórica para definição das tecnologias e para o modelo proposto



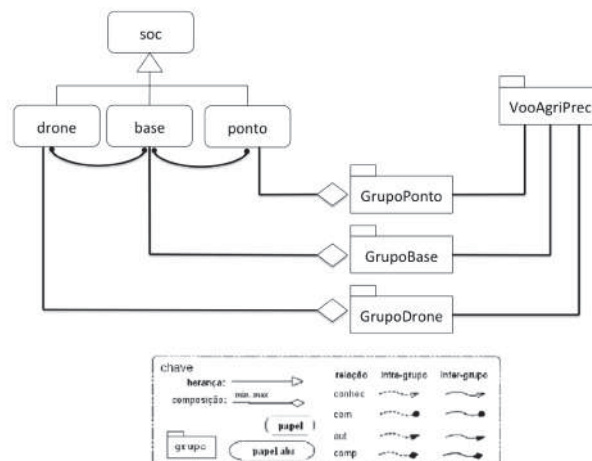


Figura 5.2: Definição dos grupos e papéis dos agentes no modelo e as restrições da comunicação

- grupo-ponto: agentes que representam os pontos a serem visitados no mapa.

A Especificação Estrutural do Sistema Multiagente também define que os agentes no sistema podem adotar três papéis: drone, base ou ponto. Será inserido no modelo um agente para cada elemento existente no cenário real.

A quantidade de agentes em cada grupo pode variar, devendo ter o mínimo de um agente e não havendo limite máximo. Existe também uma restrição de comunicação entre os grupos de agentes, sendo que os agentes do grupo-drone podem comunicar-se com os agentes do grupo-base e os agentes do grupo-base podem comunicar-se com os agentes do grupo-ponto; e os agentes do mesmo grupo podem-se comunicar entre si. Uma visão geral dos grupos de agentes, os seus papéis e a comunicação é apresentado na Figura 5.2.

A comunicação dentro do modelo acontece em uma analogia a um leilão tradicional - Figura 2.2, no qual, o agente com papel drone seria o cliente que estaria disposto a adquirir um produto e este possui como moeda de negociação a sua autonomia. Já o produto a ser adquirido seriam os pontos que serão visitados e o objetivo é maximizar o número de pontos. O agente drone negocia com as bases de recargas, que negocia com os pontos próximos para formar uma proposta (plano de voo), que será escolhida pelo drone.

Para a definição deste processo de negociação, foi criada uma Especificação Funcional no MOISE+, sendo atribuído aos agentes que desempenham o papel de drone iniciar o leilão. Esse processo é executado no início da simulação para todos os Drones, ou quando um Drone retorna para a base para recarga.

O modelo de negociação é chamado de *half-line*, já que é orientado ao evento de retorno do Drone para a base de recarga. Nesse momento, o Drone alimenta o modelo com as informações coletadas durante sua missão e o modelo iniciará um novo processo de otimização, criando e enviando aos Drones em solo, seu novo plano de voo.

Como a Especificação Funcional é dividida em objetivo global (esquema) e objetivo local (meta) - Figura 5.3, sendo que o objetivo local é atribuído a um agente e o objetivo global só é atingido quando todos os objetivos locais forem satisfeitos. Foi criada uma especificação hierárquica sequencial. Assim, o esquema otimizar-voo-drone só é atingido após as metas drone-call-for-proposal, base-call-for-proposal, ponto-call-for-proposal, ponto-accept-propose, base-accept-propose, drone-accept-propose serem alcançadas de forma sequencial. A partir do prefixo de cada meta é possível identificar qual é o agente responsável por sua execução.

As metas call-for-proposal são responsáveis por solicitar pedidos de proposta aos respectivos agentes. Após receber todas as propostas e selecionar a melhor, o agente executa

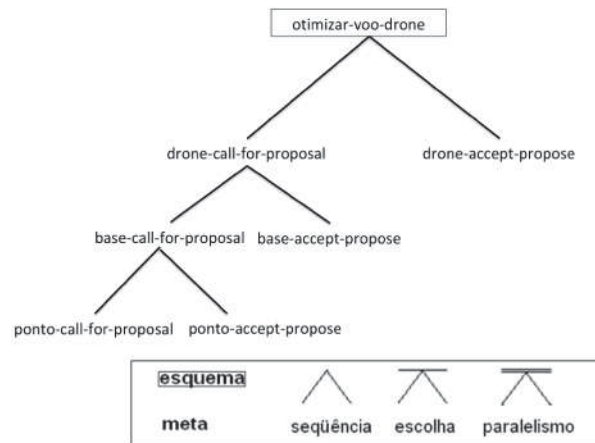


Figura 5.3: Especificação Funcional para o Sistema Multiagente

a meta accept-propose, que atualizará todos os agentes do modelo sobre qual a proposta foi escolhida.

Por fim, no ambiente MOISE+ é necessário relacionar a Especificação Estrutural (organização do Sistema Multiagente) com a Especificação Funcional (atividades do Sistema Multiagente). Isso é feito pela Especificação Deontica.

Essa especificação define que quem inicia o leilão são os agentes do tipo Drone, que se comunicam com os agentes bases que, por sua vez, inicia a negociação com os agentes pontos e que essas ações são realizadas de forma sequencial.

## 5.2 PROPOSTA DE AGENTES COGNITIVOS

Os agentes no Sistema Multiagentes proposto são cognitivos, o que permite que eles conheçam a organização na qual foram inseridos e negociem entre si para atingir o objetivo global.

Para sua codificação foi utilizada a linguagem JASON que é baseada na linguagem AgentSpeak(L) e utiliza os princípios de BDI, permitindo que os agentes tenham bem especificadas no seu código suas crenças, desejos e intenções.

As crenças são informações que todos os agentes compartilham de um repositório de crenças. Esse repositório é atualizado a cada aterrissagem do Drone. Existem alguns repositórios de crenças que podem ser utilizados pelo presente modelo, já que distribui essas informações para todos os agentes. Destaca-se como repositório a ferramenta CArAgO, utilizada para este fim na presente tese.

A partir das crenças, os Drones conhecem todos os pontos do mapa que já foram visitados e alocados, informações sobre o vento e posições das bases, entre outros dados. Essas informações são necessárias para o cálculo do custo de deslocamento do Drone de um ponto para outro.

Demais informações sobre o Sistema Multiagente, como, por exemplo, quantos Drones e quantas bases estão disponíveis, os agentes cognitivos não precisam armazenar como crenças, uma vez que o modelo MOISE+ fornece tais informações.

Para os agentes, o único desejo é concluir a visita a todos os pontos. Ao atingir esse estado, o desejo é satisfeito e o objetivo global é atingido. Porém, cada tipo de agente tem intenções específicas para atingir o seu desejo. O Drone, por exemplo, tem a intenção de iniciar um leilão com a base e essa intenção só é atingida quando ele recebe todas as propostas, contudo, isso não significa que o desejo foi atingido, pois se for necessário, novas intenções de iniciar a

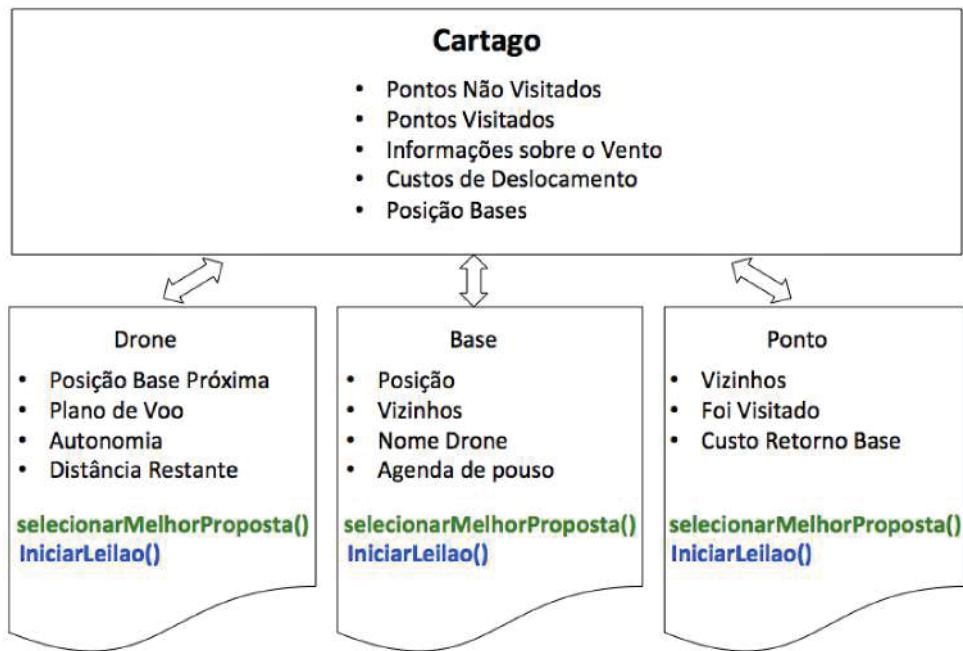


Figura 5.4: Estrutura dos agentes codificados em BDI, com as crenças em preto, desejos em verde e as intenções em azul

negociação com a base poderão ser geradas dinamicamente. O mesmo acontece com as bases e os pontos.

A Figura 5.4 apresenta a estrutura de codificação dos agentes cognitivos, apresentando o repositório de crenças globais que é compartilhado entre todos os agentes (CArtAgO) e cada um dos agentes, com as crenças em preto, os desejos em verde e as intenções em azul.

### 5.3 FIPA CONTRACT-NET APLICADO AO MODELO DE OTIMIZAÇÃO DE VOOS DE DRONES

A utilização do protocolo FIPA Contract-Net para leilões a partir de agentes codificados em JASON garante que os agentes conseguirão negociar entre si mesmo estando em plataformas distintas.

Basicamente, o leilão iniciará quando um agente drone é adicionado ao grupo-drone ou quando este retorna para base. Neste momento, o agente assume a meta para iniciar um call-for-proposal com o agente base no qual ele foi inserido.

O agente base, por sua vez, assume a meta call-for-proposal, que envia pedido de proposta aos agentes pontos. Esses agentes iniciam um leilão recursivo visando apresentar a melhor proposta para o voo do Drone.

Ao identificar a melhor proposta, o agente inicia a meta accept-propose, que atualiza o sistema como um todo com a nova proposta, informando quais pontos o Drone sobrevoará.

Da mesma forma, após o Drone terminar um leilão e compartilhar seu plano de voo, ele envia uma mensagem para os demais Drones que estão nas bases, para que eles atualizem suas intenções com os novos planos de voos otimizados.

A Figura 5.5 apresenta um Diagrama de Sequência do uso do protocolo FIPA Contract-NET com o algoritmo de otimização recursivo.

Ao ser inserido no ambiente, ou ao retornar para a base, o Drone inicia um leilão para o agente base, que por sua vez, inicia um novo leilão para cada ponto próximo. Cada ponto que

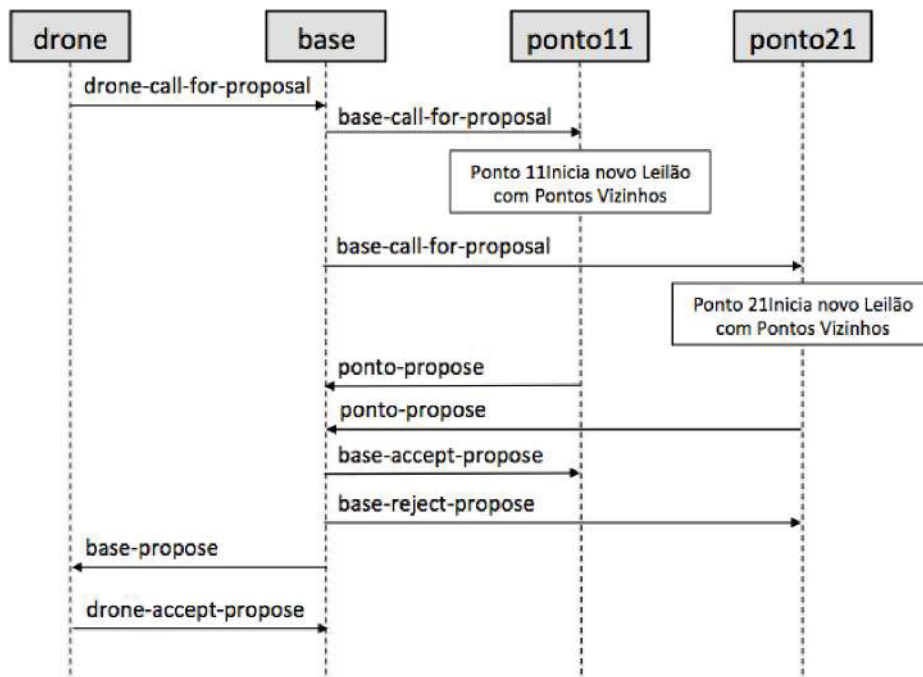


Figura 5.5: Diagrama de sequência de leilões usando FIPA Contract-Net.

recebe o pedido de proposta, inicia um novo leilão para cada um de seus pontos próximos. Após o ponto receber a proposta dos seus pontos vizinhos, ele escolhe a melhor, coloca o custo para a visita nele mesmo, e retorna para a base sua proposta. A base, por sua vez, após receber a proposta dos seus pontos vizinhos, escolhe a melhor e envia para o Drone. Neste exemplo, tem-se um Drone por Base, mas se existisse mais de uma base, o Drone escolheria a melhor proposta, que é a que possui o menor custo, e informa a base, que por sua vez, torna o plano de voo público.

#### 5.4 LEILÕES RECURSIVOS

Para exemplificar o funcionamento do algoritmo de leilão recursivo, passo a passo, será considerado o cenário com um Drone, uma base de recarga e quatro pontos de passagem. Para o exemplo, considera-se a autonomia do Drone em 5 pontos. No modelo, todos os elementos são codificados como agentes autônomos, conforme exemplificado na Figura 5.6.a.

O leilão inicia quando a simulação começa, ou quando o Drone retorna para a base. O Drone informa o valor de sua autonomia para o agente base de recarga na qual está pousado. O agente base trabalha como um leiloeiro. Conhecendo a autonomia do Drone, o agente base envia um pedido de proposta a todos os agentes pontos vizinhos desta base, representado na figura por agente ponto1 e ponto3, conforme Figura 5.6.b.

A quantidade de vizinhos, também conhecida como  $k$ , é um parâmetro definido pelo usuário e interfere no tempo de processamento. Quanto menos vizinhos, mais rápido é o processamento, porém, menos otimizado fica o modelo. Para o exemplo, a quantidade de vizinhos foi definida como dois.

Outro parâmetro que deve ser conhecido pelo modelo é o custo de movimentação de um ponto para outro. Este custo pode variar na prática, considerando a direção do vento e sua velocidade. Para este exemplo, todos os custos de movimentação foram valorizados como 1.

Neste cenário, o agente ponto1 responde a base com sua proposta informando que o Drone visitará 1 ponto único, com custo para visitar e retornar para a base de 2 pontos (1 para chegar no ponto1 mais 1 para retornar para base), sobrando uma autonomia de 3 pontos para

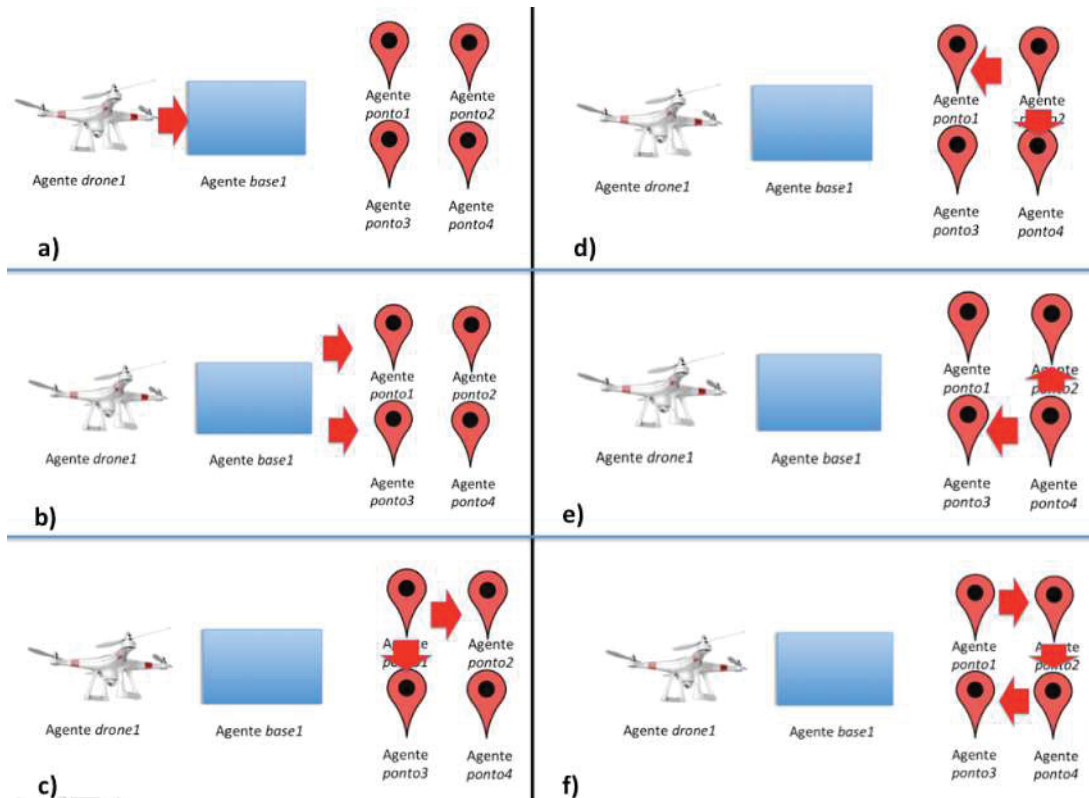


Figura 5.6: Agentes do modelo dispostos para o processo de otimização por meio do leilão recursivo - passo a passo

o Drone. O agente ponto3, como também está próximo à base, retorna para ela uma proposta informando que será o único ponto visitado e que para ser visitado o custo também será de 2 pontos e que sobrarão para o Drone uma autonomia de 3 pontos.

Como ambos os agentes, pontos 1 e 3, identificaram que sobrou autonomia para o Drone, eles iniciam uma segunda rodada de leilões. Cada um desses agentes solicita aos seus dois vizinhos propostas para visita do Drone.

O agente Ponto1 envia mensagem para seus vizinhos, sendo o agente Ponto2 e o agente Ponto3, informando que um Drone estará sobre o agente ponto1 e que deseja visitar outros pontos únicos e que a autonomia restante para este Drone é de 3 pontos. O agente Ponto2 responde ao Ponto1, informando que pode ser visitado e, nesse caso, o Drone visitará 2 pontos únicos (Ponto 1 e Ponto 2), com um custo total de 4 pontos (1 ponto para chegar ao Ponto1 + 1 ponto para chegar ao ponto2 + 2 pontos para voltar a base), sobrando para o Drone a autonomia de 1 ponto - Figura 5.6.c. Da mesma forma, o agente Ponto3 também envia sua proposta ao agente Ponto1, que iniciou a segunda rodada do leilão recursivo.

O agente Ponto1, ao receber as propostas de seus dois vizinhos mais próximos (pontos 2 e 3), seleciona a melhor, que pela função objetivo do modelo é a proposta que visita o maior número de pontos únicos e que possua o menor custo, e formata uma resposta para base, contendo a proposta mais lucrativa. Caso tenha duas propostas com a mesma quantidade de pontos únicos visitados e o mesmo custo, é enviada para a base a primeira proposta recebida.

Da mesma forma, o agente Ponto2 e o agente Ponto3 iniciam uma terceira rodada de leilões, já que eles sabem que o Drone ainda possui autonomia para visitar outros pontos.

O agente Ponto2 faz uma solicitação de proposta de visita para seus dois vizinhos mais próximos - agente Ponto1 e agente Ponto4, solicitando propostas para a visita do Drone - Figura 5.6.d.

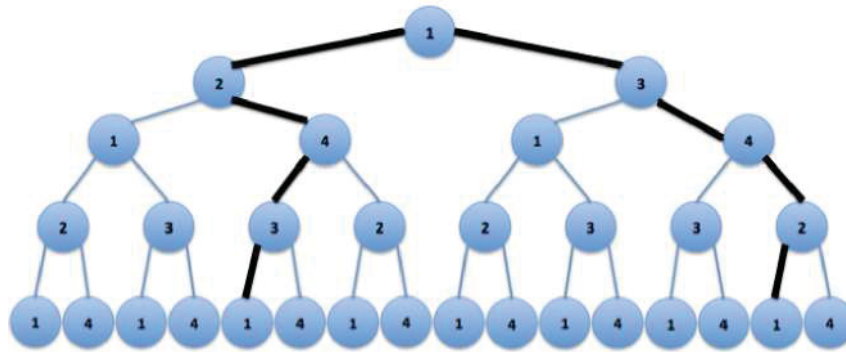


Figura 5.7: Grafo com as possibilidades de deslocamento de um Drone considerando a posição inicial o vértice 11,  $k = 2$  e autonomia do Drone igual a 5

O agente Ponto4 informa que pode ser visitado, porém, com um custo acumulado de 3 pontos. Para chegar até ele (1 ponto para chegar no Ponto1 + 1 ponto para chegar no Ponto2 + 1 ponto para chegar no Ponto4) e terá um custo para retornar a base de 2 pontos. Assim, a resposta dada ao agente Ponto2 é que serão visitados 3 pontos únicos, com um custo de 5 pontos. em paralelo o agente Ponto1 também inicia uma nova rodada de leilão com seus vizinhos.

Por fim, a última rodada de leilão inicia com o agente Ponto4 enviando propostas de visitas a seus dois vizinhos, que são o agente Ponto2 e Ponto 3 - Figura 5.6.e. Estes enviam suas propostas de visitas, sendo que o Ponto4 informa que pode ser visitado também, com um custo de deslocamento até ele de 4 pontos, sendo que o itinerário será formado pelo Ponto1, Ponto2, Ponto4 e Ponto3, havendo um custo de retorno para a base de 1 ponto, conforme apresentado na Figura 5.6.f. Dentre as propostas recebidas pela base, esta é uma das mais lucrativas para o modelo, uma vez que o Drone, com 5 pontos de autonomia, pode visitar 4 pontos únicos e retornar para a base.

Vale ressaltar que para o leilão, com a base de recarga na posição que está, dois resultados ótimos são encontrados. O primeiro, com o Drone percorrendo o Ponto1, o Ponto2, o Ponto4 e o Ponto3, fazendo um giro no sentido horário e, o segundo, com o Drone percorrendo o Ponto1, o Ponto3, o Ponto4 e o Ponto2, fazendo o giro em sentido anti-horário, conforme pode ser observado no grafo de visitas da Figura 5.7.

Com isso, ao concluir o processo recursivo, que ocorre quando a autonomia do Drone passa a ser negativa, situações nas quais o Drone não teria autonomia para retornar à base, as propostas dos leilões são passadas aos agentes que solicitaram a proposta, selecionam a melhor e retornam ao agente que solicitou a proposta inicial. Isso ocorre sucessivamente até que a proposta retorne ao agente Base, que seleciona a melhor proposta de voo e atualiza uma tabela com o agendamento dos eventos que ocorrerão, eventos como início e término de cada voo e os pontos nos quais eles ocorrerão.

Esse processo de leilão considera que a base de recarga está em todas as posições possíveis do cenário, ou seja, primeiro é realizada a otimização considerando que a base esteja no Ponto1, o melhor resultado é armazenado, na sequência, considera-se que a base esteja no Ponto2, armazena o resultado e assim por diante. Ao final, o melhor resultado é escolhido, permitindo a escolha da melhor posição inicial da base. Apesar da grande quantidade de leilões, esse processo garante a escolha da melhor posição inicial da base de recarga. Porém, heurísticas podem ser adotadas para diminuir o número de leilões, conforme apresentado na Seção 6.2.

## 5.5 CONCLUSÃO

Este capítulo apresentou o modelo proposto com base nas ferramentas constantes no Capítulo 4.

Foi apresentada a estrutura em MOISE+ para a organização Estrutural, Funcional e Deontica do Sistema Multiagente, garantindo a abertura do modelo.

Para o tratamento da dinamicidade, os agentes foram estruturados para compartilhar crenças, desejos e intensões, sendo codificados utilizando a ferramenta JASON com a linguagem AgentSpeak(L).

A troca de mensagens, que acontece basicamente enquanto ocorrem os leilões, é realizada por meio de um fluxo de leilões estruturados utilizando protocolo FIPA Contract-NET

Já a otimização para a escolha do melhor plano de voo ocorre por meio de leilões recursivos, que acontecem nos pontos base.

O objetivo deste capítulo foi apresentar o modelo. O Capítulo 6 apresentará a metodologia dos testes, sendo que os resultados serão apresentados no Capítulo 7.

## 6 HEURÍSTICAS E METODOLOGIAS APLICADAS AO ALGORITMO DE OTIMIZAÇÃO BASEADO EM LEILÕES RECURSIVOS

Este capítulo apresenta inicialmente o algoritmo de leilão recursivo no seu modelo original, com os testes de seu desempenho. Em seguida, são propostas três heurísticas que visam diminuir o número de leilões, sem comprometer significativamente os resultados encontrados. Ao final do capítulo, são apresentadas algumas metodologias para otimização de resultados que apresentam bons resultados em cenários maiores, que são comuns na agricultura de precisão, permitindo a diminuição no tempo de processamento.

### 6.1 CENÁRIO E METODOLOGIA PARA A REALIZAÇÃO DOS TESTES

O Modelo Aplicado de Otimização de Voo de Grupo de Drones é formado por Drones, bases de recargas e pontos de passagem. Cada um destes elementos é codificado como um agente cognitivo e inserido em um Sistema Multiagente. Detalhes da organização do modelo são apresentados no Capítulo 5.

Os pontos de passagens, conhecidos como *waypoints*, são os pontos identificados dentro de uma área bidimensional, sobre os quais os Drones devem sobrevoar. Estes pontos são escolhidos dada uma determinada precisão, quanto mais perto um ponto do outro, maior a precisão. A precisão influencia diretamente na quantidade de pontos que serão sobrevoados durante um voo de Drone.

Para os testes, foi optado por uma precisão considerada viável para um Modelo Digital de Elevação, assim, um ponto estará distante do outro em 10 metros. Para um cenário de 5 x 5 pontos de passagem, por exemplo, pode-se considerar que a área a ser sobrevoada será de 50m x 50m. Basicamente, a área sobrevoada será um plano cartesiano. Embora a área possa ser qualquer figura geométrica, para os testes foram utilizados apenas quadriláteros que é o formato mais comum das áreas agrícolas.

Como a área a ser sobrevoada é tratado como um plano cartesiano, ela é dividida em linhas e colunas. Por convenção, as linhas são representadas pela letra x e as colunas pela letra y, assim, os pontos são identificados pelo número da linha (x) e coluna (y). Desta forma, o primeiro ponto no canto superior esquerdo do plano cartesiano corresponde ao índice 11 ( $x = 1$  e  $y = 1$ ).

Embora existam algoritmos para escolha da melhor posição inicial para a base de recarga, do ponto de vista prático, as melhores posições seriam os vértices da área a ser sobrevoada, o que facilitaria sua alocação, não necessitando de mensuração para colocar a base em uma posição exata no meio de uma área. Uma simplificação utilizada nos testes foi que a base de recarga fosse estática, embora o modelo também permita a utilização de bases de recarga dinâmicas.

Em relação ao número de Drones, o modelo não possui um limite máximo, devendo ser inserido no mínimo um Drone no ambiente. Foi estabelecido a necessidade de uma base de recarga por Drone. Esta foi uma simplificação utilizada para os testes, mas o modelo prevê uma proporção diferente de Drones para bases de recarga.

Os Drones possuem autonomia e tempo de recarga variável. A autonomia nesta tese é medida em número de pontos que ele pode sobrevoar, também chamado de *holps*. Considerando uma distância de 10 metros entre um ponto e outro, se o Drone possui autonomia de 121 pontos, isso significa que ele pode sobrevoar 121 pontos distantes 10 metros um do outro, o que corresponde a uma autonomia de 1.210 metros lineares, considerando que o Drone sobrevoa uma área a 1 m/s. No estudo realizado em Brito et al. (2019a), foi verificada que a autonomia do



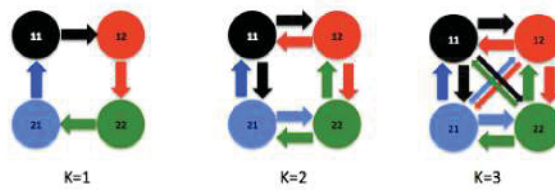


Figura 6.1: Cenário com 4 pontos, com  $k$  variando de 1 a 3

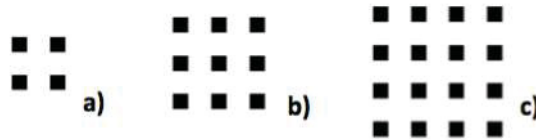


Figura 6.2: a) Cenário com 4 pontos. b) Cenário com 9 pontos. c) Cenário com 16 pontos

Phantom IV da empresa DJI, foco do presente estudo, é de 121 pontos. Em relação ao tempo de recarregamento da bateria do Drone, foi considerado o mesmo tempo que este permaneceu em voo, assim, para voos maiores, será necessário um maior tempo de recarga.

A constante  $k$  do modelo influencia diretamente a qualidade da resposta, assim como no tempo de processamento. Com o valor de  $k$  pequeno, resultados podem ser encontrados de forma rápida, porém, serão explorados menos possibilidades no grafo de busca e as chances de encontrar resultados ótimos diminuem. Em contrapartida, utilizando um valor de  $k$  maior, tem-se a possibilidade de encontrar resultados melhores, porém, com um tempo de processamento maior.

A Figura 6.1 apresenta um exemplo com  $k = 1$ ,  $k = 2$  e  $k = 3$  para um cenário de quatro pontos.

Uma vez conhecidas as características utilizadas para os testes do MAOVDG, em especial no que diz respeito ao algoritmo de otimização, serão apresentadas heurísticas para diminuir a complexidade computacional.

## 6.2 OTIMIZAÇÃO DO LEILÃO RECURSIVO POR MEIO DE HEURÍSTICAS

Inicialmente foi realizada uma análise do algoritmo de leilão recursivo para identificar suas limitações, visando otimizá-lo para atender ambientes dinâmicos. Esta otimização acontece com o uso de heurísticas.

Para esta seção, como o objetivo é identificar e avaliar o comportamento do algoritmo, considerou-se apenas um Drone e uma base de recarga, realizando sobrevoo em cenários com 4 pontos (2x2), 9 pontos (3x3) e 16 pontos (4x4), conforme Figura 6.2. Para estes testes foi utilizado um valor de  $k=2$  e considerada a autonomia do Drone suficiente para sobrevoar todos os pontos.

A utilização das heurísticas visa diminuir o número de leilões realizados pelo algoritmo e, conseqüentemente, diminuir o tempo de processamento. O número de leilões depende da autonomia do Drone: quanto maior a autonomia, maior a quantidade de leilões dentro do modelo, assim como a quantidade  $k$  de vizinhos.

Primeiramente será apresentado o modelo original, que não possui heurística. Na sequência é apresentada a heurística h1, que define um ponto inicial para base, depois h2 que limita o sobrevoo dos pontos e, por último, h3, definindo um parâmetro de término do leilão.

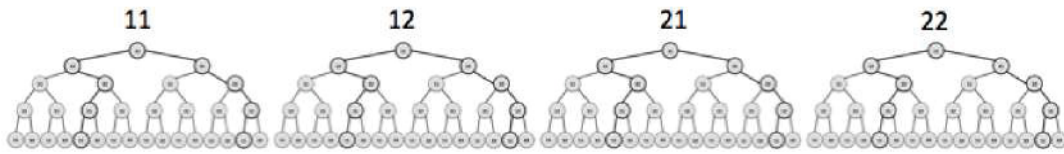


Figura 6.3: Grafo com 120 possibilidades leilões, autonomia igual a 5,  $k = 2$ , e todos os pontos sendo testados como inicial

Tabela 6.1: Resultados da execução do algoritmo de otimização com leilões recursivos para cenário com 4, 9 e 16 pontos

Cenário	Qtd. de Ótimos encontrados	Tempo de Processamento	Qtd. de Leilões
4 pontos	8	32 ms	120
9 pontos	234	1.908 ms	9.198
16 pontos	2.432	20.4448 ms	2.242.784

### 6.2.1 Testes com o Modelo Original

É chamado de modelo original os testes de otimização utilizando leilões recursivos sem nenhum tipo de poda, ou seja, o grafo de leilões é percorrido em sua totalidade, assim, todas as possibilidades são exploradas, encontrando todas as soluções ótimas. Os resultados obtidos a partir da execução deste modelo foi utilizado como parâmetro de comparação para os modelos nos quais foram aplicadas heurísticas.

Para o cenário com quatro pontos, tem-se um total de 120 leilões, 30 leilões para cada ponto inicial. Estes leilões podem ser vistos como arestas da Figura 6.3. Nesta execução são encontrados 8 resultados ótimos (arestas em destaque, são 2 resultados ótimos para cada ponto inicial).

O mesmo teste do Modelo Original foi executado para os outros dois cenários: 3x3 pontos e 4x4 pontos e os resultados são apresentados na Tabela 6.1.

Pode-se observar na coluna Qtd. de Leilões que os números de leilões aumentam muito, e em um cenário pequeno, com 16 pontos, a quantidade de leilões supera 2 milhões. Este aumento no número de leilão não é bom, já que também influencia no tempo de processamento. Isso demonstra a necessidade do uso de heurísticas para obter resultados equivalentes em tempo computacional menor.

Para saber a quantidade de leilões que acontecerá no modelo é utilizada a Equação 6.1, sendo que a quantidade de leilões depende da quantidade  $k$  de vizinhos que os pontos terão, assim como a autonomia do Drone ( $n$ ). Nesta análise assintótica, observa-se que a complexidade computacional é uma somatória de exponencial, a qual ainda é multiplicada pelo número de pontos  $n$ , o que não é bom do ponto de vista de desempenho computacional.

$$n \cdot \sum_{i=1}^n k^i \quad (6.1)$$

Para cada cenário, foi escolhido um dos resultados ótimos para ser apresentado em uma interface gráfica, conforme Figura 6.4. Neste, o ponto inicial está em vermelho, e a direção do voo é representado pelas setas.

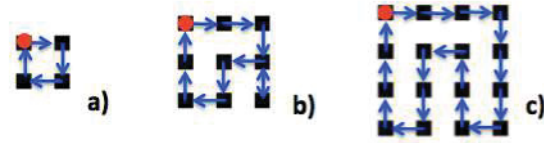


Figura 6.4: Primeiro resultado obtido para o cenário com 4 pontos (a), 9 pontos (b) e 16 pontos (c), em todos, a base está no Ponto 1 e o voo acontece no sentido horário

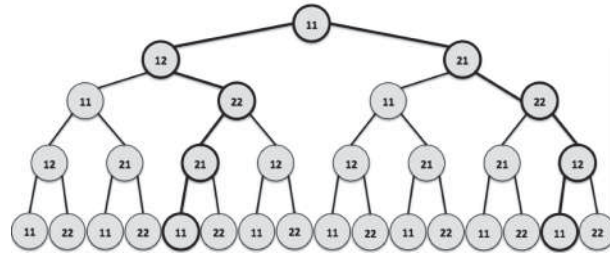


Figura 6.5: Modelo de leilões representado em um grafo após aplicação de h1

### 6.2.2 Testes do Modelo com Heurística h1 - Definição do ponto inicial

Para diminuir o número de leilões, a primeira heurística, chamada de h1, é aplicada, a qual define um único ponto inicial para a base de recarga, não sendo necessário que todos os pontos sejam testados como ponto inicial. Este ponto, quando se trata de base de recarga fixa, será um dos vértices da área a ser sobrevoada. Com bases móveis, a primeira posição da base será um vértice, facilitando sua alocação por parte do usuário e, em seguida, os outros pontos podem ser obtidos por algoritmos, tais como Saque et al. (2019).

Após a aplicação da heurísticas h1, o número de leilões diminuiu consideravelmente e o grafo resultante é apresentado na Figura 6.5.

Apenas uma das ramificações do grafo foi explorada, a correspondente ao ponto inicial 11, o que já garante dois resultados ótimos. Para os testes, também foram utilizados os cenários com 3x3 pontos e 4x4 pontos e os resultados são apresentados na Tabela 6.2.

A quantidade de leilões diminuiu ainda mais à medida que aumenta o tamanho da área a ser sobrevoada, uma vez que a quantidade de pontos iniciais, se todos fossem testados, é maior, chegando a uma redução de 93,75% no cenário com 16 pontos, porém, mesmo assim, o número de leilões para 16 pontos é superior 140 mil leilões.

Tabela 6.2: Quantidade de leilões antes e após a aplicação da heurística h1, que define o ponto inicial para base de recarga

Cenário	Qtd. Leilões no Modelo Original	Qtd. Leilões no Modelo com Heurística h1	% redução
4 pontos	120	30	75%
9 pontos	9.198	1.022	88,8%
16 pontos	2.242.784	140.174	93,75%

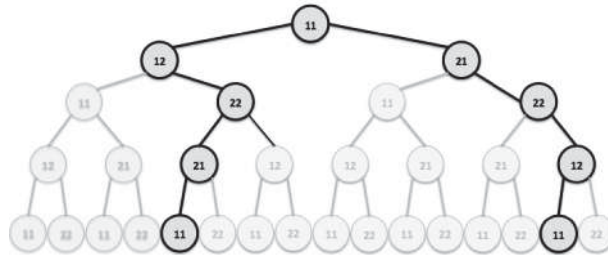


Figura 6.6: Grafo de poda após a aplicação da heurística h2

Tabela 6.3: Quantidade de leilões antes e após a aplicação da heurística h2, que não permite que um mesmo ponto seja visitado mais de uma vez

Cenário	Qtd. Leilões após Heurística h1	Qtd. Leilões no Modelo com Heurística h2	% redução
4 pontos	30	8	73,3%
9 pontos	1.023	18	98,24%
16 pontos	249.398	32	99,98%

Em uma análise assintótica, a complexidade do algoritmo, após a utilização de h1, ainda é uma somatório de exponencial, porém, agora não possui mais o fator de multiplicação que havia no modelo original. A quantidade de leilões após h1 pode ser calculada pela Equação 6.2.

$$\sum_{i=1}^n k^i \quad (6.2)$$

As execuções com a utilização de h1 encontraram resultados ótimos, assim como aconteceu no modelo original - Figura 6.4.

### 6.2.3 Testes do Modelo com Heurística h2 - um ponto não pode ser visitado duas vezes

Para diminuir o número de leilões, é proposta a heurística h2 que define que pontos já visitados em um itinerário não podem ser visitados duas vezes, uma vez que o problema em questão é encontrar um circuito Hamiltoniano, quando possível.

Assim, a heurística h2 foi aplicada ao resultado obtido após h1, obtendo-se uma melhora ainda mais significativa. Para uma análise gráfica da poda, tem-se a Figura 6.6, sendo aplicada a heurística h2. As áreas em menor destaque eram exploradas em h1, mas não são mais exploradas após h2.

Os mesmos testes foram realizados com o cenário de 3x3 pontos e 4x4 pontos, conforme apresentado na Tabela 6.3.

Para cenários grandes, com 16 pontos de passagem, por exemplo, a redução no número de leilões chegou a superar 99%. A fórmula que representa a quantidade de leilões após aplicação da heurística h2 é apresentada na Equação 6.3. Esta fórmula é a mesma do somatório dos elementos de uma progressão geométrica, considerando que todo ponto possua como vizinho o ponto anterior, que já foi sobrevoado e que não entrará na próxima rodada de leilões.

$$k \frac{(k-1)^n - 1}{k-2} \quad (6.3)$$

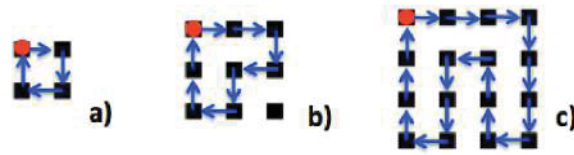


Figura 6.7: Resultados após h2, observando que o cenário com 9 pontos não possui todos os pontos sobrevoados.

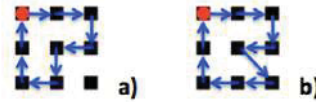


Figura 6.8: Resultado para o cenário com 9 pontos. a) Utilizando  $k = 2$  b) utilizando  $k = 5$

Assim, tem-se duas situações específicas. A Equação 6.3 é aplicada para  $k=3$  ou superior, para  $k=2$  é utilizada a Equação 6.4.

$$k.n \quad (6.4)$$

Nesta situação, a complexidade do algoritmo, que era na ordem de somatória de exponencial em h1 e h2, passa à somente exponencial, dependendo da autonomia do Drone e do valor de  $k$ , o que representa um bom ganho em termos de desempenho computacional.

Em relação ao ótimo global, para o cenário com 4 e 16 pontos, esses ótimos foram encontrados, já que são cenários que formam um circuito Hamiltoniano. Para o cenário de 9 pontos, que não é um circuito Hamiltoniano, a solução encontrada permitiu passar em 8 pontos, uma vez que pela heurística h2 dois pontos não podem ser visitados no mesmo voo. Os resultados da execução, após h2, são apresentados na Figura 6.7.

Para um monitoramento formado por grupo de Drones, ou em cenários nos quais um único Drone deve realizar mais de um voo, o que é comum na agricultura de precisão, o fato de não sobrevoar um dos pontos não é um problema, pois este ponto poderá ser coberto por outro Drone ou em um próximo voo.

Porém, em situações nas quais há um único Drone e este deve realizar apenas um voo para sobrevoar todos os pontos, em circuitos não hamiltoniano, como este, o resultado ótimo consegue ser atingido apenas se modificado o valor de  $k$ . Na Figura 6.8, tem-se um exemplo para  $k=2$  e para  $k=5$ .

Assim, é possível verificar que o valor de  $k$  é um fator importante para a qualidade da resposta encontrada, influenciando também na quantidade de leilões que serão realizados. Estudo sobre o melhor valor de  $k$  foi realizado e publicado em Brito et al. (2020c) e será detalhado na próxima seção.

#### 6.2.4 Testes do Modelo com Heurística h3 - Parâmetro de parada de processamento após o primeiro ótimo ser encontrado

Sobre os resultados obtidos após a heurística h2, foi realizada a execução da heurística h3, sendo que nesta é definido um ponto de parada no algoritmo de otimização. Assim, após encontrar o primeiro resultado ótimo, ou viável, depende como o usuário parametriza o h3, o processamento é encerrado.

É possível comparar graficamente a poda na quantidade de leilões, observando a Figura 6.9. Observa-se que apesar de haver dois resultados ótimos neste cenário, o processamento encerra após o primeiro ótimo ser encontrado.

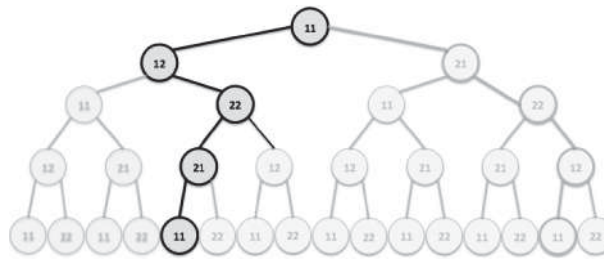


Figura 6.9: Grafo de poda após a aplicação da heurística h3

Tabela 6.4: Quantidade de leilões antes e após a aplicação da heurística h3, que possui um fator de término de processamento

Cenário	Qtd. Leilões após Heurística h2	Qtd. Leilões no Modelo com Heurística h3	% redução
4 pontos	8	4	50.0%
9 pontos	18	9	50.0%
16 pontos	32	16	50.0%

A heurística h3 também foi aplicada ao cenário de 3x3 pontos e 4x4 pontos, sendo o resultado apresentado na Tabela 6.4.

A redução na quantidade de leilões, assim como no tempo de processamento na heurística h3 não é determinística, pois depende de quantos resultados ótimos foram encontrados após a heurística h2, assim como qual será o primeiro ramo a ser explorado. Desta forma, a complexidade computacional do algoritmo após h3 pode variar de linear (melhor caso - Equação 6.5) a ter a mesma complexidade computacional da heurística h2 (pior caso - Equação 6.6).

$$n \quad (6.5)$$

$$k \frac{(k-1)^n - 1}{k-2} \quad (6.6)$$

Em relação aos resultados, as execuções após o h3 encontraram resultados idênticos ao processamento com h2 - Figura 6.7, porém, com um tempo de processamento menor.

Dada a eficiência da aplicação de h1, h2 e h3, todos os testes realizados a partir deste já consideram essas três heurísticas no algoritmo de otimização baseado em leilões recursivos.

O conteúdo desta seção foi aplicado ao cenário da agricultura de precisão, e **Submitted** um artigo ao Journal Computer and Electronic in Agriculture (Brito et al., 2020a) (artigo completo no Apêndice A.12).

### 6.3 METODOLOGIA DE OTIMIZAÇÃO APLICADA A CENÁRIOS REAIS

Os testes realizados na seção anterior, assim como esta sessão, apresentaram o comportamento do algoritmo antes e depois da aplicação de heurísticas e por este motivo foi optado por cenários menores, com poucos pontos de passagem, para ser mais didática a apresentação das podas.

Esta seção objetiva mostrar novas metodologias de otimização do algoritmo. Estas metodologias não são consideradas heurísticas, pois não realizam podas no grafo gerado, porém, permitem que o primeiro resultado factível seja encontrado mais rapidamente.

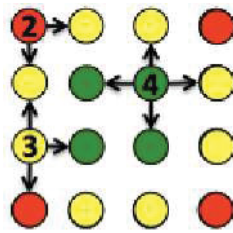


Figura 6.10: Representação gráfica da quantidade ideal de vizinhos dada a posição de um ponto no plano cartesiano

Tabela 6.5: Quantidade de leilões após h3 com  $k=4$  e, após a aplicação da metodologia de diferentes valores para  $k$

Cenário	Qtd. Leilões após h3	Qtd. Leilões após $k$ variável	Qtd. De Troca de Mensagens
16 pontos	170 leilões	114 leilões	32,94%

A seguir serão apresentadas três metodologias: utilização de diferentes  $k$ , escolha do vizinho mais próximo e divisão em áreas de voo. Para os testes é utilizado apenas o cenário de 4x4 pontos, assim, é possível realizar uma comparação em relação aos testes realizados com as heurísticas, estas apresentadas na seção anterior.

### 6.3.1 Utilização de Diferentes Valores de $k$ em um Mesmo Cenário

Para otimizar os resultados do sobrevoo de Drones em uma grande área quadrática, comuns na agricultura de precisão, alguns fatores podem ser otimizados, entre eles a quantidade de vizinhos  $k$ . Para uma análise de qual seria o melhor valor de  $k$ , uma pesquisa foi realizada e publicada em Brito et al. (2020c), sendo definida a utilização de diferentes valores de  $k$ , dependendo da posição do ponto no cenário.

Com o estudo foi identificado que os pontos dos vértices podem ter um número  $k$  menor, uma vez que eles possuem apenas dois vizinhos significativos para os quais podem ocorrer voos. Já os pontos internos devem ter um número de vizinhos maior, com  $k = 4$ , já que nestes pontos o Drone pode fazer um voo para Norte, Sul, Leste ou Oeste. Por fim, nas arestas da área a ser sobrevoada, o número de vizinhos ideal é  $k = 3$ , já que em um dos lados, se o Drone sobrevoar sairá do perímetro de voo. A Figura 6.10 apresenta o cenário com 4x4 pontos, com os respectivos valores de  $k$ . Em vermelho os pontos dos vértices, com dois vizinhos, em amarelo os pontos das arestas, com 3 vizinhos, e em verde os pontos centrais com 4 vizinhos.

O artigo completo se encontra no Apêndice A.9, com detalhes de como o estudo foi realizado.

Uma análise comparativa da melhora após a metodologia de utilizar diferentes valores de  $k$  é apresentada na Tabela 6.5. Para a comparação, foi utilizada  $k=4$ . No primeiro teste, com h3, foram necessários 170 leilões para obter o primeiro resultado factível, e com diferentes valores de  $k$  o primeiro resultado foi encontrado após 114 leilões.

A complexidade computacional com a aplicação da metodologia de diferentes valores de  $k$  não é possível de ser calculada, como ocorreu com as heurísticas, pois a complexidade depende do formato da Figura e do seu tamanho, mas é possível inferir pela Figura 6.11 que quanto mais desproporcional, ou seja, maior a diferença entre um lado e outro do quadrilátero, menor será a proporção de pontos internos em relação aos pontos externos, e, com isso, menos vizinhos para os pontos serão necessários, assim como uma menor quantidade de leilões.

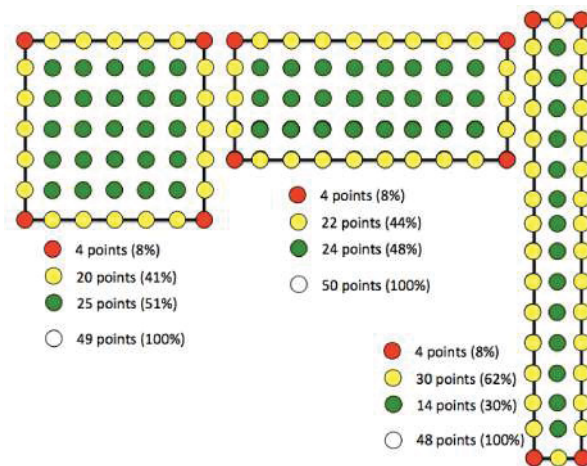


Figura 6.11: Proporção de pontos internos (maior quantidade de vizinhos) e pontos externos (menor quantidade de vizinhos) para diferentes formatos de área

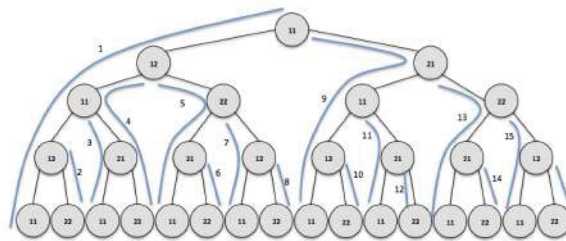


Figura 6.12: Fluxo de leilões realizados em um grafo onde a busca em profundidade acontece

### 6.3.2 Identificação do Vizinho mais Próximo

Outra técnica que diminui consideravelmente o tempo de processamento para encontrar a primeira solução factível, uma vez que a heurística  $h_3$  define este como parâmetro de término do processamento, é a escolha do primeiro vizinho mais próximo. Esta técnica foi testada e submetido os resultados em (Brito et al., 2020d). O artigo completo se encontra no Apêndice A.15, com detalhes de como o estudo foi realizado.

O processamento baseado em leilões recursivos escolhe o primeiro vizinho próximo e a partir deste é iniciada uma nova rodada de leilão recursivo com seu próximo vizinho e, assim por diante, então, em um grafo é realizada uma busca em profundidade em cada um de seus ramos. Assim, uma boa escolha do primeiro vizinho melhora significativamente o desempenho do algoritmo. Observe na Figura 6.12 o fluxo dos leilões recursivos. Se o resultado ótimo estiver no fluxo representado pela linha 9 e a linha 1 for escolhida como início do leilão, isso fará que todos os leilões referentes a linha 1 seja realizado, mesmo que o resultado esteja na linha 9. Uma forma de otimizar este resultado é uma boa escolha do primeiro vizinho mais próximo.

Para a definição de quem será o primeiro vizinho próximo, deve-se saber qual é o padrão de voo que será realizado, por exemplo, um voo em espiral do centro para a extremidade, um voo espiral da extremidade para o centro, um voo em linha ou um voo aleatório.

Como trata-se do sobrevoo de uma área quadrática, foi optado por escolhas de vizinhos próximo para voo em linha, sendo nas linhas pares o vizinho mais próximo o da esquerda, e nas linhas ímpares, o vizinho mais próximo o da direita, já que muito provavelmente, é neste padrão de vai-e-vem que acontecerá o encontro do primeiro resultado ótimo, ou ainda, do primeiro resultado factível de processamento. A Figura 6.13 apresenta uma representação gráfica para a metodologia da escolha do vizinho mais próximo para um cenário de 4x4 pontos.



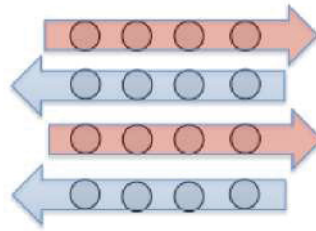


Figura 6.13: Representação gráfica da metodologia de escolha do vizinho mais próximo para um cenário de 4x4

Tabela 6.6: Quantidade de leilões antes e após a aplicação da metodologia de escolha do vizinho mais próximo

Cenário	Qtd. Leilões após k variável	Qtd. leilões após Vizinho mais próximo	Redução no n. de leilões
16 pontos	114 leilões	43 leilões	62,8%

Para uma análise comparativa, foram utilizados os resultados obtidos após a aplicação da metodologia de diferentes valores de  $k$  com o resultado obtido após a metodologia do vizinho mais próximo. A comparação é apresentada na Tabela 6.6.

Foi observado, também, que em cenários que formam um circuito Hamiltoniano foi possível encontrar resultados ótimos com uma quantidade de leilões menores do que em cenários que não formam circuito Hamiltoniano. Uma comparação destes planos de voo é apresentada na Figura 6.14.

Em relação a quantidade de leilões, para circuitos Hamiltonianos o mesmo consegue encontrar resultados mais rapidamente, dado o padrão de vai-e-vem dos voos, para circuitos não Hamiltonianos, a quantidade de mensagens é maior, conforme pode ser visto na Tabela 6.7.

### 6.3.3 Definição da Áreas de Atuação

Por fim, para voos envolvendo múltiplos Drones foi observado que muitas vezes os leilões recursivos se estendem para áreas que normalmente seriam sobrevoadas por outros Drones e, por este motivo, estes leilões acabavam não sendo escolhidos por sobrevoarem poucos pontos. Esta metodologia também foi testada e submetido os resultados, junto com os resultados da metodologia anterior, em (Brito et al., 2020d).

Para evitar que este tipo de leilões acontecesse, foi criada uma metodologia para divisão da área de atuação do Drone por conta de sua proximidade com a base, desta forma, cada Drone faz os leilões com os pontos de sua área de atuação, só após esta ter sido sobrevoada é que um Drone pode então sobrevoar pontos da área de atuação de outros Drones.

A Figura 6.15 apresenta diferentes áreas de atuação para um cenário de 8x8 pontos, sobrevoada por quatro Drones. Na Figura, cada cor representa a área de atuação de um Drone.

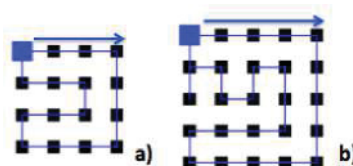


Figura 6.14: Comparação dos percursos obtidos para circuito Hamiltoniano (a) e não Hamiltoniano (b)

Tabela 6.7: Quantidade de leilões para circuito Hamiltoniano e não Hamiltoniano

Cenário	Quantidade de Pontos	Quantidade de Leilões
4x4 pontos	16 pontos	43 leilões
5x5 pontos	25 pontos	137 leilões

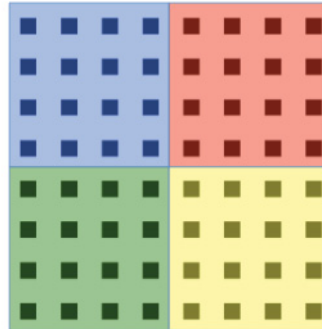


Figura 6.15: Representação gráfica da metodologia da divisão de área de voo para um cenário de 8x8 com quatro Drones.

Para testar esta metodologia foi inicialmente gerado o plano de voo para um cenário no qual o algoritmo de otimização faz uso das três heurísticas e das duas metodologias apresentadas anteriormente. Na sequência, foi gerado plano de voos após a utilização da metodologia da divisão da área de voo dos Drones. O resultado é apresentado na Tabela 6.8.

Como pode ser observado, após a aplicação da metodologia da divisão da área de atuação dos Drones, o primeiro resultado foi obtido com uma diminuição de 65,1% da quantidade de leilões.

#### 6.4 CONCLUSÃO

Este capítulo apresentou uma análise do algoritmo de otimização baseado em leilões recursivos, utilizado pelo MAOVDG.

Esta análise foi importante, pois antes de ser realizada, não se sabia o comportamento do leilão recursivo no seu modelo original e, após a análise, foi possível a aplicação de heurísticas para melhorar o tempo de processamento, que depende da quantidade de leilões.

Da mesma forma, ao se aplicar o algoritmo de leilões recursivos em cenários formados por centena e até milhares de pontos, foi identificado que algumas metodologias poderiam otimizar o tempo de processamento, uma vez que ao encontrar o primeiro resultado factível, o processamento se encerra.

Tabela 6.8: Quantidade de leilões antes e após a aplicação da metodologia de divisão de área de atuação por Drones

Cenário	Sem divisão de área de atuação	Com divisão de área de atuação	Qtd. De Troca de Mensagens
8x8 pontos 4 drones	536 leilões	187 leilões	65,1%

No Capítulo 5 foi apresentado o modelo proposto para o MAOVD e neste Capítulo as heurísticas e metodologias para otimizar o tempo de processamento. O Capítulo 6 apresentou os resultados encontrados quando este algoritmo foi aplicado em cenários reais.

## 7 RESULTADOS E DISCUSSÕES

Este capítulo apresenta os testes realizados para validação do MAOVDG, focando no desempenho do algoritmo de otimização baseado em leilão recursivo, e na garantia de abertura e dinamicidade do modelo em relação a eventos adversos que ocorrem no ambiente, como mudanças de direção/velocidade do vento e nos comportamentos dos Drones. Foram, ainda, realizadas comparações quantitativas e qualitativas em relação à modelos existentes na literatura. A realização dos testes ocorreu no ambiente de simulação MultiDrone Simulator.

As comparações quantitativas do modelo proposto foram realizadas com o algoritmo de otimização dinâmica proposta por Khaledi et al. (2018) e adaptado para execução de teste no Multidrone Simulator em Afghah et al. (2020) (Artigo completo disponível no Apêndice A.13).

Por fim, foi realizada uma comparação qualitativa com trabalhos simulares encontrados na literatura, apresentados na seção 2.3. Trabalhos Correlatos.

### 7.1 TESTES DE OTIMIZAÇÃO DO MODELO APLICADO DE VOO DE GRUPO DE DRONES

Nesta seção são apresentados três testes para validação do MAOVDG: cenário com 64 pontos - Circuito Hamiltoniano com um Drone, cenário com 121 pontos - Circuito Não Hamiltoniano com um Drone e cenário com 1.156 pontos com quatro Drones.

Os dados obtidos dos testes realizados foram comparados quantitativamente com o algoritmo proposto originalmente por Khaledi et al. (2018).

#### 7.1.1 Testes em Cenário com 64 Pontos - Circuito Hamiltoniano

Para os testes com um único Drone, foi inicialmente utilizada uma área de 8x8 pontos, que equivale a  $80m \times 80m = 6.400$  metros quadrados, ou 0,64 hectare. Esta área é composta por 64 pontos de passagem e forma um circuito hamiltoniano.

A Figura 7.1.a. apresenta o resultado obtido pelo algoritmo de otimização baseado em leilão recursivo. Para completar o percurso, que é um circuito Hamiltoniano, o algoritmo precisou de 65 pontos de autonomia.

O mesmo teste foi realizado pelo algoritmo de otimização guloso, iniciando também no ponto 11. Este conseguiu fazer o percurso com uma autonomia de 71 pontos, 6 a mais do que a execução do leilão recursivo. O resultado é apresentado na Figura 7.1.b.

A comparação dos dados das duas execuções é apresentada na Tabela 7.1.

Como pode ser observado, o algoritmo de leilões recursivo conseguiu um resultado que passava por todos os pontos e retornava para a base, precisando de uma autonomia 8,5%

Tabela 7.1: Análise comparativa entre Leilões Recursivo x Otimização Gulosa para um cenário com 64 pontos

	Qtd. pontos para retornar a base	Tempo de processamento	Qtd. de troca de mensagens
Otimização com Leilão Recursivo	65 pontos	1.6s	167
Otimização Gulosa	71 pontos	0.8s	71

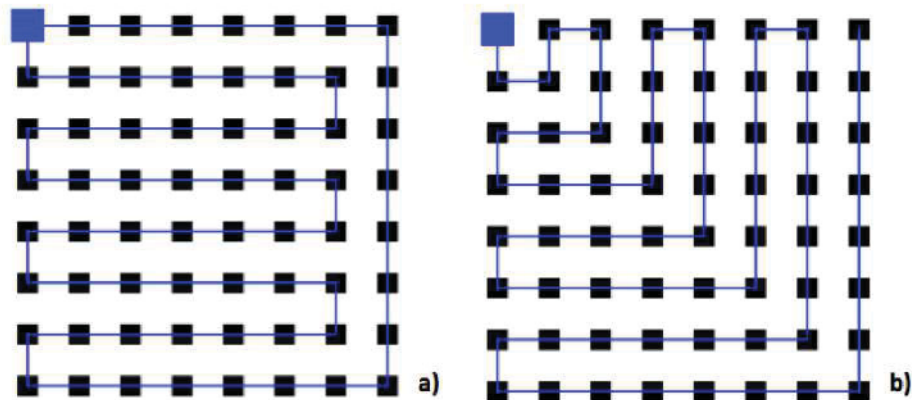


Figura 7.1: a) Resultado obtido pelo Leilão Recursivo no cenário com 64 Pontos. b) Resultado obtido pela Otimização Gulosa no cenário com 64 Pontos

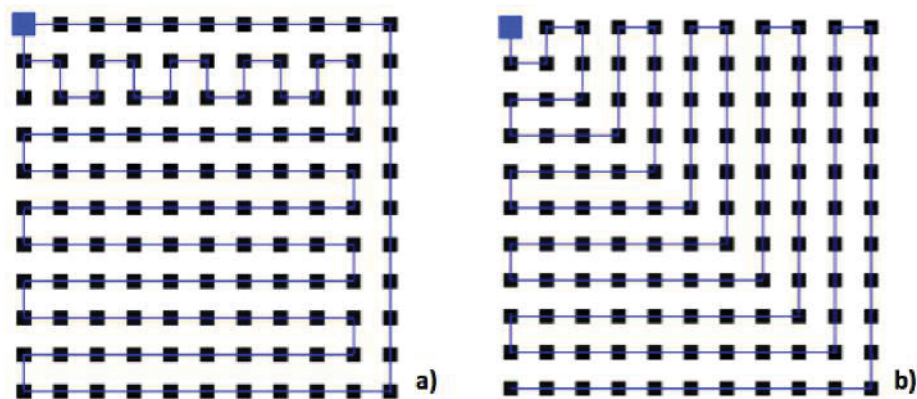


Figura 7.2: a) Resultado obtido pelo Leilão Recursivo no cenário com 121 Pontos. b) Resultado obtido pela Otimização Gulosa no cenário com 121 Pontos

inferior. Em relação ao tempo de processamento, assim como o número de mensagens trocadas, o algoritmo de leilão recursivo precisou de mais recurso computacional. Porém, mesmo neste cenário, próximo do limite máximo de voo do Drone, o processamento ficou inferior a 2 segundos.

### 7.1.2 Testes em Cenário com 121 pontos - Circuito Não Hamiltoniano

Em um segundo momento foi realizada a comparação para um cenário com 121 pontos, sendo esta a autonomia de um Drone DJI Phantom. Este cenário tem como característica não formar um circuito hamiltoniano. O resultado da otimização por meio dos leilões recursivos é apresentado na Figura 7.2.a.

O mesmo teste foi realizado com o algoritmo de otimização guloso - Figura 7.2.b.

Neste cenário, a otimização por leilão recursivo conseguiu sobrevoar todos os pontos e retornar para a base utilizando 122 pontos. Já a Otimização Gulosa permitiu sobrevoar todos os pontos e retornar para a base gastando 132 pontos, 10 pontos a mais, ou seja, uma performance 8% inferior se comparado ao resultado obtido com o leilão recursivo. Como 121 pontos (11x11) é a autonomia máxima de um DJI Phantom. Este é considerado o pior caso para voos com um único Drone e, ainda assim, o processamento utilizando leilões recursivos foi realizado em 8,2 segundos.

Tabela 7.2: Análise comparativa entre Leilões Recursivo x Otimização Gulosa para um cenário com 121 pontos

	Qtd. pontos para retornar a base	Tempo de processamento	Qtd. de troca de mensagens
Otimização com Leilão Recursivo	122 pontos	8.2s	978
Otimização Gulosa	132 pontos	1.4s	153

.156

Tabela 7.3: Análise comparativa entre Leilões Recursivo x Otimização Gulosa para um cenário com 1 ponto e 4 Drones

	Qtd. pontos para retornar a base	Tempo de processamento	Qtd. de troca de mensagens
Otimização com Leilão Recursivo	1.435 pontos	20.4s	4.976
Otimização Gulosa	1554 pontos	6.2s	602

Os dois testes anteriores consideraram um único Drone sobrevoando a área em um único voo, considerando o tempo de sobrevoe e o retorno para a base. Observa-se a superioridade do algoritmo de otimização baseado em leilões recursivos, em especial, para cenários maiores. Para o cenário com 64 pontos, o algoritmo baseado em leilões recursivos conseguiu o resultado 9,23% melhor considerando o número de pontos sobrevoados, já para cenário maior, com 121 pontos, o algoritmo a melhora foi de 8,1%.

### 7.1.3 Testes em Cenário com 1.156 Pontos - Quatro Drones

O próximo teste foi realizado considerando o cenário sobrevoado por 4 Drones. Para os testes foi considerada parcialmente a área experimental do curso de Agronomia da UTFPR-Câmpus Pato Branco, que possui 340x340 metros, equivalente a 115.600 metros quadrados, ou 11,56 hectares. A Figura 7.3.a. apresenta o resultado do sobrevoe utilizando o algoritmo de otimização baseado em leilões recursivos.

Como pode ser observado, cada uma dos quatro Drones foi posicionado nos vértices da área a ser sobrevoada. O uso de cores auxilia a identificação dos voos de cada Drone. Na figura, para facilitar a visualização, as linhas de retorno do Drone para a base foram retiradas, porém, a cada voo o Drone volta para base de recarga.

Para este cenário foram necessários no total 12 voos de Drones, sendo que cada Drone realizou 3 voos para sobrevoar a área.

O mesmo teste foi realizado com o algoritmo de Otimização Guloso e o resultado é apresentado na Figura 7.3.b.

Na Tabela 7.3 estão dados obtidos que permitem comparar o desempenho dos voos alocados pelo algoritmo de leilões recursivos e pela busca gulosa.

Para melhor visualização da alocação dos recursos são apresentados gráficos de Gantt - A Figura 7.4.

No gráfico, as linhas representam cada um dos 4 Drones do modelo, diferenciados por cor. O tempo é medido por voo de pontos, assim, o Drone 1 realiza o voo em 108 pontos inicialmente. Considerando a velocidade de um metro por segundo, o Drone levaria 10 segundos

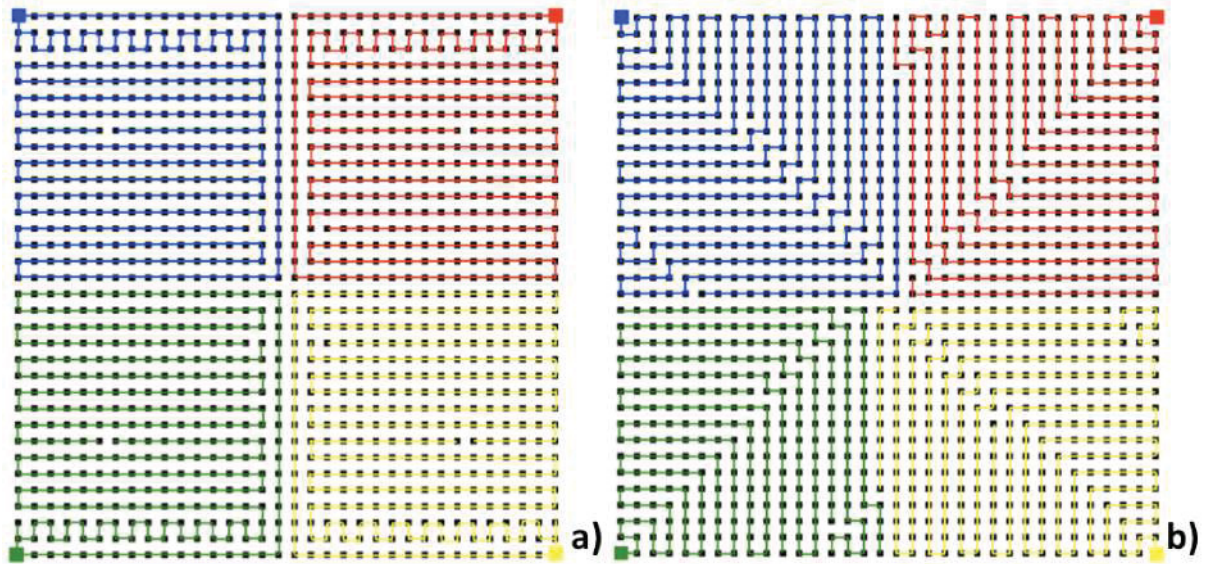


Figura 7.3: a) Resultado obtido pela otimização por leilões recursivos no cenário com 1.156 pontos e quatro Drones.  
b) Resultado obtido pela Otimização Gulosa no cenário com 1.156 pontos e quatro Drones

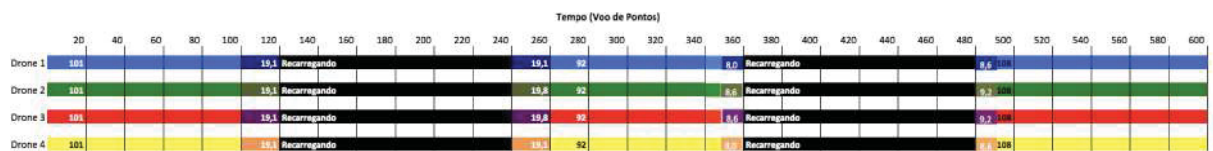


Figura 7.4: Gráfico de Gantt para o cenário com 1156 pontos, 4 Drones, autonomia de 121 pontos e otimização realizada por meio de leilões recursivos.



Figura 7.5: Gráfico de Gantt para o cenário com 1156 pontos, 4 Drones, autonomia de 121 pontos e otimização realizada por meio de algoritmo guloso.

para alcançar cada ponto, assim, o tempo de voo em segundos é obtido multiplicando o valor da coluna por 10.

Após o primeiro voo que cobre 108 pontos, o Drone precisa retornar para a base em linha reta, dando um tempo equivalente ao sobrevoo de 19,1 pontos (ou 191 segundos). Na base, o Drone permanece carregando a bateria antes de iniciar um novo voo. Os tempos de deslocamento da base até o ponto a ser sobrevoado e o retorno à base são apresentados no Gantt com uma cor diferente. Esta cor, antes do Recarregamento, significa que o Drone está retornando para base e, após o recarregamento, significa que ele está avançando até o próximo ponto a ser sobrevoado. O horizonte total para sobrevoar os 1.156 pontos, utilizando 4 Drones e a otimização com leilões recursivo é de 600 pontos, ou seja, 6.000 segundos ou 1,67 horas.

Gráfico semelhante foi composto com os dados resultantes do uso da técnica de otimização gulosa e o resultado é apresentado na Figura 7.5.

Neste, observa-se que o mapa só foi percorrido após 780 voos de pontos utilizando o algoritmo Guloso, o que equivale a um aumento de 30% em relação aos leilões recursivos, demonstrando que a superioridade dos leilões recursivos é ainda maior em cenários maiores e com uso de vários Drones.

## 7.2 APLICAÇÃO DO MODELO DE OTIMIZAÇÃO DE VOO DE GRUPO DE DRONES EM UM DISPOSITIVO MÓVEL

Uma preocupação ao desenvolver o MAOVGD foi permitir que o modelo pudesse ser executado em computadores portáteis, tais como Raspberry, devices Androids, Smartphones, entre outros, já que este provavelmente será o tipo de processamento que o algoritmo terá em campo. Assim, o presente modelo foi utilizado em um Device Android (Smartphone Galaxy S5), por meio de uma aplicação. O resultado desta aplicação e dos testes foi publicado em Brito et al. (2020f) (artigo completo no Apêndice A.10).

Para realizar os testes, foi desenvolvido um app Android do tipo *wizard*, no qual o usuário informa a posição inicial do Drone (para os testes, considerou apenas um Drone), a área a ser mapeada, a precisão entre um ponto e, desta forma, os pontos de passagens são gerados automaticamente. E o dispositivo gera o plano de voo e o desenha na tela. As telas do aplicativo são apresentadas na Figura 7.6.

Conforme apresentado na Figura, em a) é possível digitar o nome do local onde acontecerá o voo; em b) é apresentado o mapa, que o usuário pode movimentar e enquadrar na área em que acontecerá o voo; em c) é possível definir o perímetro do voo; em d) são apresentados os pontos de passagem; em e) é selecionada a precisão, ou seja, a distância entre um ponto e outro; e em f) é gerada a rota que é apresentada graficamente para o usuário.

Para o teste apresentado no artigo é utilizado o cenário da área experimental do curso de Agronomia da UTFPR-Câmpus Pato Branco. Esta área tem 340 x 340 metros, totalizando 1.156 pontos a serem sobrevoados. Foi considerado Drone com esta autonomia, o que aumenta significativamente o tempo de processamento, embora sabe-se que a autonomia de um DJI Phantom é de 121 pontos. Mesmo considerando uma autonomia de 1.156 pontos, o algoritmo





Figura 7.6: Telas do aplicativo Android para otimização de voo de grupo de Drones

encontrou uma solução após 176 segundos de processamento, isso em um *device* Android, o que mostra a viabilidade do algoritmo, mesmo sendo executado em um *device* limitado.

### 7.3 AVALIAÇÃO DA ABERTURA E DINAMICIDADE DO MODELO

Os testes de desempenho do algoritmo de otimização baseado em leilões recursivos apresentados nas seções anteriores foram realizados no ambiente Multidrone Simulator. O desempenho do algoritmo, em si, não depende de sua organização em relação aos Sistema Multiagente ou aos agentes cognitivos, já que o desempenho do leilão recursivo é medido pelo número de troca de mensagens e, conseqüentemente, tempo de processamento; já a qualidade do plano de voo encontrado é medida pela quantidade de pontos únicos sobrevoados a cada voo.

Nesta seção, é apresentada a avaliação do algoritmo em relação a sua abertura e dinamicidade.

Para esta análise, não serão apresentadas informações quantitativas, já que se trata de uma análise qualitativa. Assim, para facilitar o entendimento, foi utilizado o mesmo cenário, com 34x34 pontos, com 4 Drones, autonomia de voo de 121 pontos (equivalente a um DJI Phantom 4).

Em um primeiro momento foi testada a abertura do modelo, verificando se ele permite, durante uma missão:

- retirar Drones do ambiente;
- incluir Drone ao ambiente.

Da mesma forma, em outra bateria de testes foi verificada a dinamicidade do modelo. Para isso, durante uma missão foi:

- modificada a velocidade do vento;
- solicitado o retorno precoce de um dos Drone para a base.

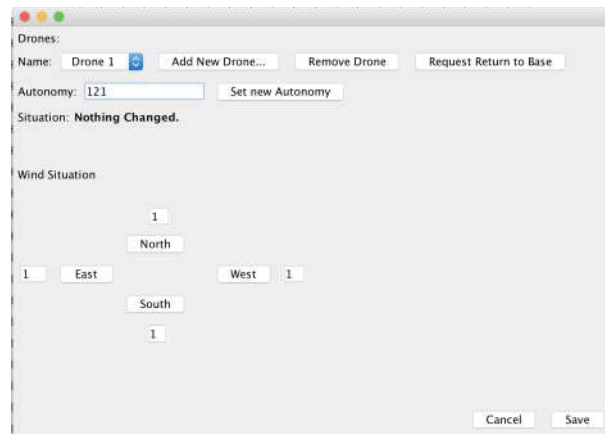


Figura 7.7: Tela para configuração de eventos externos ao ambiente, para teste de abertura e dinamicidade

Para trabalhar com a abertura/dinamicidade é necessário executar a simulação no ambiente MultiDrone Simulator passo a passo. Toda mudança no ambiente só será considerada após o retorno do primeiro Drone à base. É neste momento que o ambiente tem percepção das mudanças e os novos planos de voo são executados. A tela de configuração é apresentada na Figura 7.7.

Nesta tela do MultiDrone Simulator é possível incluir Drones, informando seu nome, sua posição inicial e sua autonomia; escolher um Drone e excluí-lo do ambiente; escolher um Drone e solicitar seu retorno para a base; e modificar a intensidade/velocidade do vento.

### 7.3.1 Testes de Abertura do Modelo

O modelo de Sistema Multiagente focado na organização, com a especificação estrutural, funcional e deôntica, faz com que o modelo funcionasse independentemente dos seus elementos, já que nesta estrutura, todo novo elemento inserido já conhece o modelo por conta da documentação, assumindo de imediato o seu papel. Caso algum elemento seja excluído, os recursos alocados a ele, se ainda não sobrevoado, são liberados para outros agentes.

Do ponto de vista de abertura, o MAOVGD se comportou como esperado e no primeiro teste foi iniciado o plano de voo com quatro Drones (Figura 7.8.a). Ao final do primeiro voo, o Drone do canto inferior direito, em amarelo, foi retirado e a rota existente foi assumida pelos demais Drones, como pode ser observado nas Figura 7.8.b. Na Figura, o Drone em amarelo permanece no modelo, já que seu primeiro voo foi concretizado, mas a partir deste ponto a área restante foi assumida por Drone 1 (em azul), Drone 2 (em vermelho) e Drone 3 (em verde).

O segundo teste realizado considerou um cenário inicial com 3 Drones e após o seu primeiro retorno a base, um quarto Drone (em magenta) foi adicionado ao modelo e um novo plano de voo foi elaborado. Na Figura 7.9.a) um plano de voo original com três Drones foi gerado e, após o primeiro voo, na Figura 7.9.b), o plano de voo atualizado já com o quarto Drone.

Em ambos os testes, com a inclusão e a retirada de Drones, o modelo conseguiu se recuperar deste evento considerado adverso e novas alocações foram realizadas após o primeiro evento de pouso, o que comprova que o modelo proposto é um modelo aberto. O tempo de processamento, em médio, para a geração de novos planos de voo para cada Drone foi, em média, de 8 segundos.

Embora nesta Tese foram apresentados apenas os testes de inclusão/retirada de Drones, o modelo também permite os testes de inclusão/retirada de pontos de passagem. O funcionamento é parecido com o que foi apresentado com Drones, pois, ao modificar o número de pontos, ao

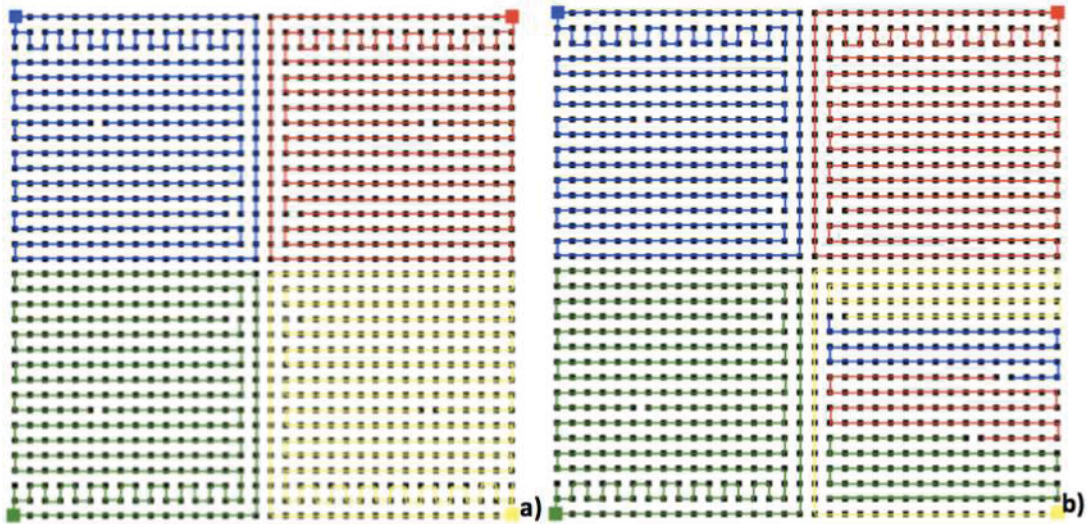


Figura 7.8: a) Plano de voo original com quatro Drones. b) Plano de voo após a retirada do Drone 4 (em amarelo) que aconteceu após o pouso do primeiro voo

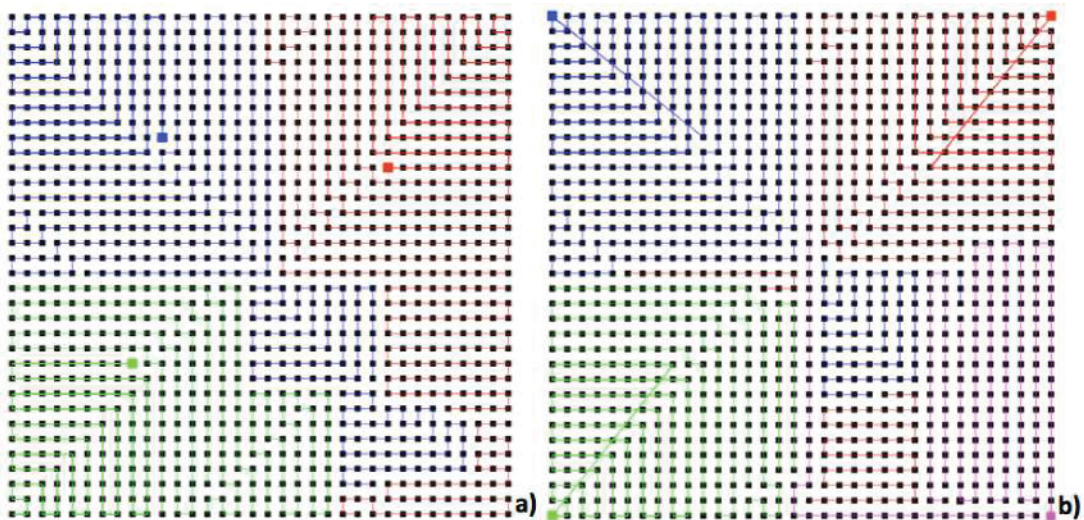


Figura 7.9: a) Plano de voo original com três Drones. b) Plano de voo atualizado com a inclusão de um quarto Drone no canto inferior direito.

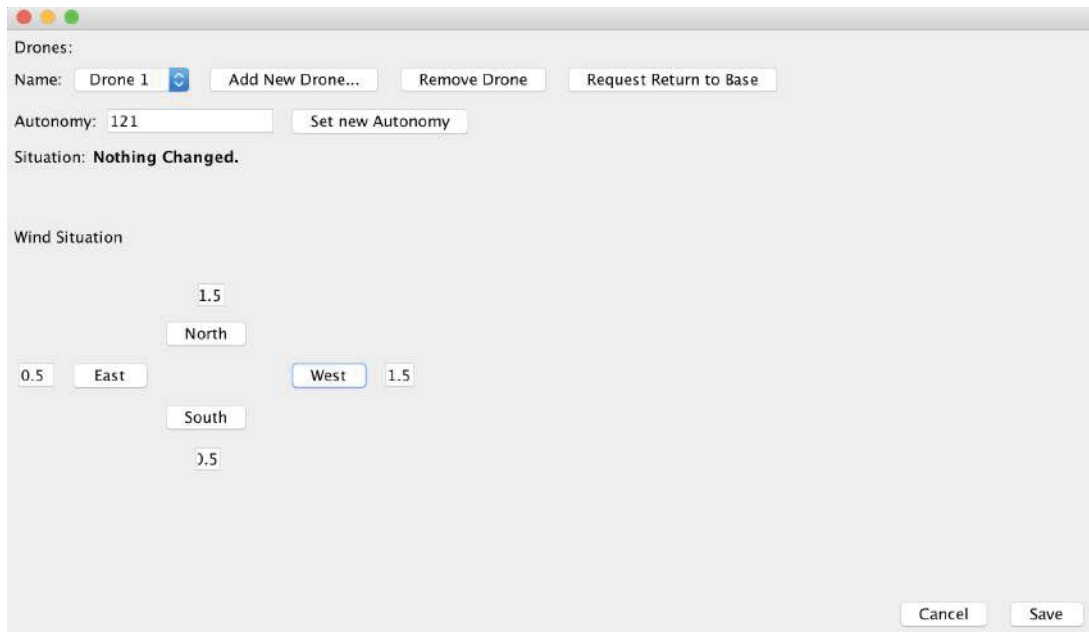


Figura 7.10: Tela de configuração para mudança da direção/velocidade do vento.

retornar para a base, o Drone identifica tais mudanças e estas já refletem na nova sequência de leilões. Em relação aos pontos, a única mudança que não pode ocorrer são nos pontos que já foram sobrevoados, sendo que estes não podem ser excluídos.

### 7.3.2 Testes de Dinamicidade do Modelo

Em um segundo momento, foi testada a dinamicidade do ambiente. O ponto mais importante para que esta característica fosse alcançada pelo MAOVGD é a inteligência dos agentes inseridos.

Todos os agentes, sejam eles pontos, bases ou Drones, foram codificados de acordo com as características do BDI, na linguagem AgentSpeak(L), por meio do *framework* Jason. Estes agentes possuem suas crenças, sendo que algumas delas são compartilhadas com todos por meio da ferramenta Cartago. Entre as crenças compartilhadas estão os pontos já sobrevoados e a velocidade e a direção do vento, entre outras características.

Assim, a cada mudança no ambiente, o agente Drone tem acesso a elas quando retorna para a base, iniciando um novo processo de leilão recursivo, já considerando as novas informações.

Um primeiro teste de dinamicidade foi realizado mudando a velocidade e a direção do vento. Inicialmente, foi realizada a execução de um voo sem vento, conforme já apresentado na Figura 7.3.a. Mas, após o primeiro retorno a base, foi considerada uma mudança na velocidade e direção do vento, que era nulo na primeira execução. As novas configurações do vento foram realizadas conforme apresenta a tela da Figura 7.11, com o vento passando a ser no sentido nordeste, aumentando em 50% sua intensidade ao parâmetro inicialmente definido.

O plano de voo original é apresentado na Figura 7.11.a. Após esta mudança, no primeiro pouso dos Drones, novos planos de voo foram definidos, observando que agora a preferência é para voos na vertical, do Sul para o Norte, e horizontal, de Oeste para Leste, conforme apresentado na Figura 7.11.b.

Por fim, o último teste realizado para avaliar a dinamicidade do ambiente foi a simulação precoce de descarga da bateria de um Drone, considerando a execução inicial apresenta na 7.3.b., foi solicitado um retorno imediato após um voo de 13 pontos do Drone 1 (em azul). Observa-se

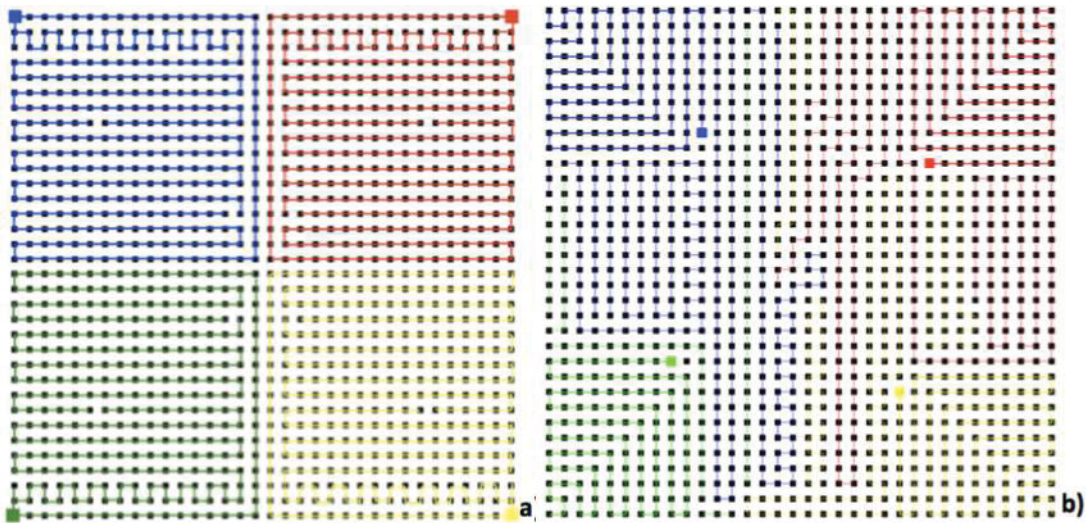


Figura 7.11: a) Plano de voo original com quatro Drones. b) Plano de voo adaptado a nova direção/velocidade do vento.

que mesmo com um Drone a menos, os demais Drones continuaram suas missões. O Drone 1 retornou à base, recarregou sua bateria, um novo plano de voo foi gerado (Figura 7.12) e a missão continuou.

Isso só é possível devido a três características importantes do MAOVGD:

1. **Otimização Rápida:** o algoritmo de otimização permite realizar alocação de novos voos, para um cenário de 121 pontos de autonomia, em média, em 8 segundos de processamento, assim, o algoritmo é rápido o suficiente, mesmo com a complexidade do problema, para encontrar a solução;
2. **Sistema Multiagente:** dada a organização dos agentes em um modelo baseado em organização, o modelo fica aberto para alocação e exclusão de agentes em qualquer momento, a única condição de parada seria um dos tipos de agente ter um número igual a zero, o que não pode acontecer no MAOVGD, por isso, a abertura do ambiente é garantida, independente do número de agentes;
3. **Agentes Cognitivos:** a dinamicidade do ambiente se dá pelo uso de agentes cognitivos, que possuem crenças (percepções do ambiente em que são inseridos), desejos (otimização do voo) e intenções (planos de voos). Isso garante que o modelo se recupere de adversidades e novos planos de voo sejam gerados a cada evento de pouso.

#### 7.4 ANÁLISE QUALITATIVA DO MODELO APLICADO DE OTIMIZAÇÃO DE VOO DE GRUPO DE DRONE

Embora muitos trabalhos tenham sido estudados, os 7 mais importantes foram descritos na Seção 2.3. Trabalhos Correlatos e são utilizados para um comparação qualitativa nesta seção. Pela especificidade da presente tese, não foi possível uma comparação quantitativa com todos esses. A comparação qualitativa aconteceu na Seção 7.1 Teste de Otimização do Modelo Aplicado de Voo de Grupo de Drones, com o algoritmo proposto comparado com o algoritmo presente em Khaledi et al. (2018).

As principais características do modelo proposto ao relação aos outros 7 trabalhos são apresentados na Tabela 7.4.

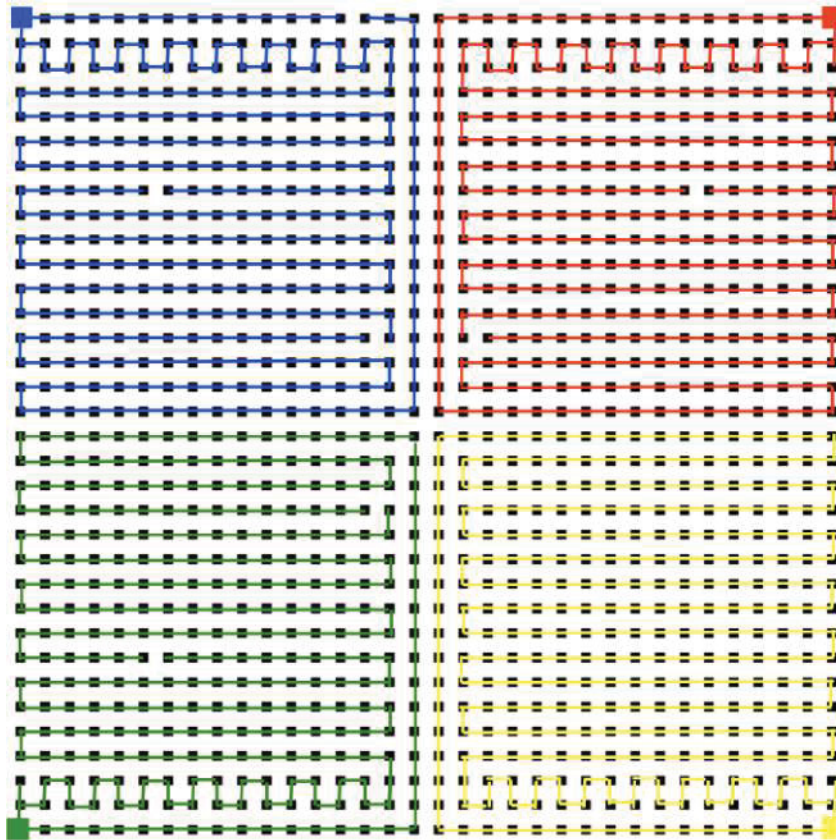


Figura 7.12: Plano de voo gerado após o retorno precoce do Drone 1, em azul, após sobrevoar 13 pontos.

Tabela 7.4: Características dos trabalhos que mais contribuíram para o modelo aplicado proposto. Na tabela, WP significa se permite ponto de passagem, FP se trata pontos proibidos, 3D se permite traçar rotas considerando a altitude de voo, Tipo de UAV qual o tipo de veículo aéreo foi utilizado nos testes, em Téc. Otim. qual o algoritmo de otimização foi utilizado, em abertura se o modelo permite a inclusão de novos elementos durante a sua execução, e por fim, dinamicidade se elementos como velocidade e direção do vento são considerados pelo modelo.

Trabalho	WP	FP	3D	Tipo de UAV	Qtd. UAV	Téc. Otim.	Abert.	Dinam.
(Avellar et al., 2015)	S	N	N	A.Fixa	2	PLIM	N	N
(Ergezer e Leblebicioğlu, 2013)	S	S	S	A.Fixa	n	AG	N	N
(Ho et al., 2013)	S	N	N	A.Fixa	1	PSO	N	N
(Menegol et al., 2018)	N	N	N	A.Fixa	n	N/A	S	N
(Ernest et al., 2013)	S	N	N	N/A	n	Híbrido	S	N
(Roberge et al., 2014)	S	N	S	A.Fixa	n	PSO/AG	S	N
(Caska e Gayretli, 2015)	S	N	S	Drone	n	AG	S	N
MAOVGD	S	N	S	Drone	n	Leilão Recurs.	S	S

A tabela apresenta os trabalhos relacionados em cada linha, com suas características nas colunas. A coluna WP (*WayPoint*) indica se os trabalhos tratam pontos obrigatórios de passagens. Neste item, o MAOGVDG é baseado em pontos que são utilizados na otimização e na geração de plano de voo. O modelo proposto permite, inclusive, que estes pontos sejam dinâmicos, assim como os Drones, podendo ser incluídos e retirados pontos durante a execução de uma missão.

Já a coluna FP (*Forbidden Point*) informa se o trabalho trata pontos proibidos. No modelo proposto tal característica não foi testada, porém, em trabalho futuro esta característica será validade, colocando nos pontos proibidos um custo superior a autonomia do Drone em determinados pontos do mapa, para evitar voo sobre lagos, rios ou mata e outros visando minimizar o risco de perder um Drone, caso alguma pane ocorra.

A coluna 3D informa se o trabalho considera a altura nos pontos de passagem. A presente tese permite definir uma altura de voo em relação ao solo (utilizado para monitoramento de produção, por exemplo) ou em relação ao nível do mar (utilizado para o Modelo Digital de Elevação). Em situações de colisão de Drones, o plano de contingência para evitar choques não é tratado pelo simulador, o próprio Drone é responsável por evitar este tipo de problema, uma vez que este tipo de operação costuma ser embarcado em alguns Drones comerciais, tais como o DJI Phantom 4.

Em relação ao tipo de UAV, é possível observar que a grande maioria dos trabalhos faz uso de Asas Fixas, dada a limitação da autonomia dos Drones, este problema é contornado pelo modelo proposto.

A coluna Qtd. UAV informa quantos veículos aéreos são suportados pela solução proposta, com "n" indicando que não há limite. Neste item, o trabalho de Avellar et al. (2015) é o mais próximo do modelo proposto, faz a otimização do voo de dois UAVs de Asa Fixa aplicado a agricultura de precisão, sendo que esse modelo não suporta mais de dois veículos pela custo de operação dos Asas Fixas, que precisam ter interação com o ser humano tanto no pouso quanto na decolagem.

A coluna Téc. Otim. apresenta o algoritmo de otimização utilizado pelos trabalhos. Destaca-se o uso de algoritmos Bioinspirados e do Programação Linear Inteira Mista utilizado por Avellar et al. (2015).

Alguns trabalhos preveem a Abertura, o que significa que suportam inclusão e exclusão de novos elementos durante a execução de uma missão. Esta característica é suprida pelo Sistema Multiagente no modelo proposto.

Por fim, dinamicidade está relacionada à recuperação de eventos adversos que acontecem durante a execução de uma missão. Dada as características do Sistema Multiagente em conjunto com os agentes cognitivos, tal dinamicidade é provida pelo modelo proposto.

## 7.5 PUBLICAÇÕES

Ao longo do desenvolvimento desta Tese, 16 artigos foram publicados/submetidos. Como existe o interesse de transformar o Modelo Aplicado de Otimização de Voo de Grupo de Drone em um produto aplicado para o desenvolvimento de um Modelo Digital de Elevação, o autor da Tese esteve envolvido em projetos do ponto de vista operacional do modelo, como pouso de decolagem autônoma de Drones, assim como em projetos que envolvem a abertura, a dinamicidade e a otimização do modelo, sendo estes últimos o foco da presente Tese.

Assim, os artigos são divididos em dois grupos: Grupo Operacional, que são trabalhos com a cooperação de alunos de graduação em Trabalhos de Conclusão de Curso e Iniciação Científica, nestes o autor desta tese foi co-autor dos trabalhos; e o Grupo Modelo de Otimização, com todos os trabalhos tendo como autoria principal o autor desta teste.

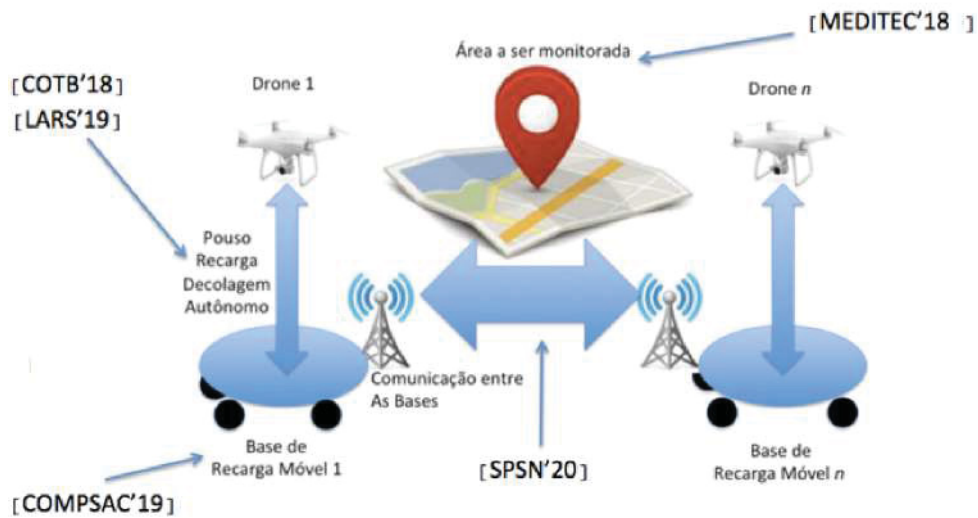


Figura 7.13: Grupo de publicações Operacional, apresentando uma visão geral do MAOVGD, já com as publicações específicas.

Inicialmente será apresentado o Grupo Operacional, cujas publicações podem ser representadas pela Figura 7.13, uma variação da Figura 2.1 apresentada na seção 2.1 deste documento.

Embora todos os trabalhos publicados tenham sido citados no texto, nesta seção serão apresentados detalhes de cada publicação. Os artigos a seguir são apresentados em ordem cronológica da pesquisa, não necessariamente da publicação.

1. **MEDITEC'18 - Sistema Web para Gerenciamento de Voos de Aeronaves não Tripuladas.** O Meditec é um evento regional, e neste foi apresentado o que seria um sistema web para o gerenciamento da missão de um grupo de Drones, onde o usuário poderia escolher os Drones que fariam parte da missão, as bases de recargas móveis, o local inicial de cada base de recarga, assim como definição do perímetro que seria sobrevoado. Este trabalho não possui inteligência, ao final, ele gera apenas um JSON com as informações registrada via interface gráfica, sendo que em um futuro este arquivo pode ser enviado para os elementos do sistema. O trabalho foi desenvolvido por um aluno da Especialização em Tecnologia Java. Neste trabalho, onde o autor da Tese foi orientador;
2. **LARS'18 - Comparison Analysis Between PID and Fuzzy Logic Controllers for Quadrotor in a Simulated and in a Real Environment.** Conferência é indexada como B3. A ideia primária, quando se iniciou as pesquisas, era utilizar Drones desenvolvidos pela própria Universidade para ser utilizado no monitoramento, assim, inicialmente foi projetado um Drone e um aluno do curso de Engenharia de Computação, desenvolveu uma análise comparativa entre PID e Fuzzy para realizar o pouso deste sobre uma base de recarga. Neste trabalho, onde o autor da Tese foi colaborador;
3. **COTB'19 - A Model for landing, taking off and autonomous battery recharging of a Parrot Ar.Drone 2.0 using computational vision and GPS features.** Conferência é indexada como B4. Com as dificuldades encontradas com os Drones montados na própria universidade, o passo seguinte foi escolher um Drone comercial, categoria 3, para realizar o pouso e decolagem autônoma. Este trabalho foi realizado por um aluno



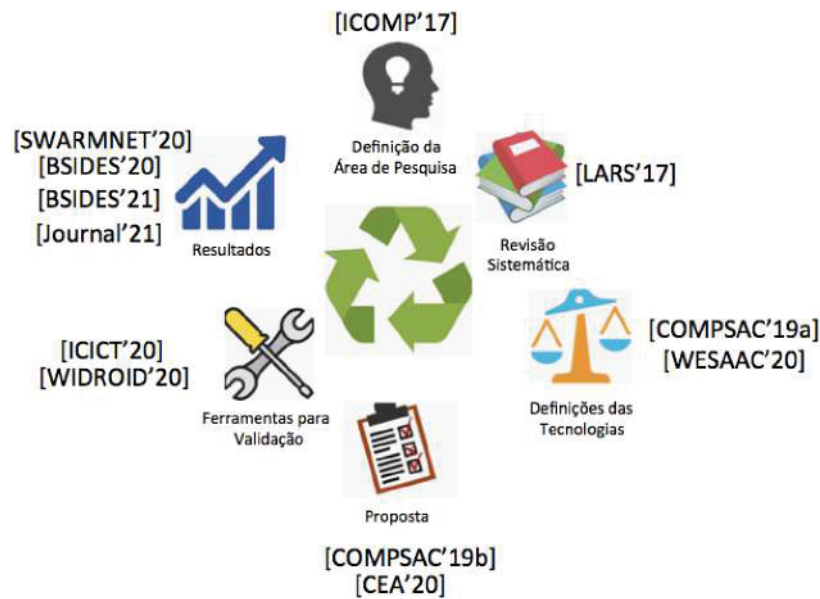


Figura 7.14: Visão geral das fases de pesquisa, já com os artigos do grupo Modelo de Otimização publicados/submetidos.

de Engenharia de Computação da UTFPR, o qual desenvolveu um sistema para o Ar. Drone da empresa Parrot, que permitia retornar para a base utilizando GPS e pousar por meio de visão computacional. Neste, o autor da Tese foi orientador;

4. **COMPSAC'19 - Optimized division of exploration areas in multi-robot systems considering static and dynamic charging stations.** Esta conferência é indexada como A2. No desenvolvimento dos trabalhos para a qualificação, foi identificado o quão custoso é testar todas as possibilidades necessárias como ponto inicial para o Drone, desta forma, com o auxílio de um aluno de IC da UFPR, foram criados mecanismos para encontrar o ponto central de figuras geométricas, usando duas técnicas descritas no trabalho. O trabalho foi realizado por um aluno da UFPR e neste trabalho o autor da Tese foi orientador;
5. **SPSN'20 - Implementing a Communication Network between Bases Station applied for Group of Drones.** Workshop internacional sem qualis no Brasil, mas que acontece dentro do DCOSS (A2). Este trabalho apresenta uma metodologia de comunicação entre as bases de recarga móveis utilizadas para o recarregamento da bateria dos Drones. Este trabalho foi **submetido** por um aluno de graduação do curso de Engenharia da Computação da UTFPR, em co-autoria com o autor desta Tese. Até o fechamento desta Tese, não havia o resultado da avaliação do artigo.

A seguir serão apresentados os artigos do Grupo Modelo de Otimização, sendo estes os artigos desenvolvidos focando na teste. O desenvolvimento da Tese pode ser divididos em etapas, sendo estas apresentadas na Figura 7.14.

A seguir um breve apresentação de cada fase, assim como os resultados.

1. **Definição da Área de Pesquisa:** a primeira etapa, iniciada em 2016, quando no ingresso do programa de doutorado, foi a definição da área de pesquisa, para isso, foram realizados estudos da viabilidade da aplicação de Drones no meio agrícola, a fim de encontrar potenciais lacunas de pesquisa. Como resultado, tem-se o Brito et al. (2017b);

2. **Revisão Sistemática:** após definir que a pesquisa seria realizada no desenvolvimento de um modelo de cooperação para múltiplos Drones, um mapeamento sistemático foi realizado, resultando um Technical Report (não publicado) e um artigo publicado em Brito et al. (2017a);
3. **Definição de Tecnologias:** com o mapeamento sistemático realizado, assim como com a leitura de vários artigos, foi definida a área de pesquisa, resultando no MAOVGD. Com subsídio do mapeamento sistemático, foram definidas as ferramentas e as tecnologias que seriam utilizadas para o desenvolvimento do MAOVGD. Como muitos trabalhos utilizam veículos aéreos do tipo asa fixa na agricultura de precisão, inicialmente foi feito um trabalho para justificar o uso de Drones neste cenário, este publicado em Brito et al. (2019a) e um trabalho **submetido** ao WESAAC'20 (Brito et al., 2020e) que apresenta o modelo baseado em sistema multiagente e agentes cognitivos para garantir a abertura e dinamicidade do modelo;
4. **Proposta:** em outubro de 2018, foi realizada a defesa da qualificação da Tese. Após as contribuições da banca, a qualificação foi dividida em dois documentos, o primeiro deles apresentando a proposta do MAOVGD e alguns resultados preliminares, que foi publicado em Brito et al. (2019b) e um artigo de Journal que foi **submetido** ao CEA'20, focando os resultados do algoritmo de otimização aplicados a agricultura de precisão para um único Drone (Brito et al., 2020a);
5. **Ferramentas e Validação:** após a definição do modelo, foi iniciado o desenvolvimento dos aplicativos e ferramentas para a validação do modelo. Para testar como o algoritmo se comportava em um dispositivo móvel, ele foi executado em um app Android e o resultado foi publicado em Brito et al. (2020f). Também foi desenvolvido um simulador para testar o MAOVGD, com os resultados de testes do simulador **submetidos** na conferência WIDROID'20 (Brito et al., 2020b);
6. **Resultados:** para otimizar o tempo de processamento em uma grande área a ser sobrevoado, uma metodologia de escolha de valor de  $k$  foi proposta e publicada em Brito et al. (2020c), o que ocorreu também com duas metodologias, que permitem otimizar o tempo por meio da escolha do vizinho mais próximo e da divisão da área de atuação dos Drones. Este artigo será **submetido** para o evento BSIDES'2021 (Brito et al., 2020d). Para os testes do leilão recursivo, foi adaptado um artigo de busca gulosa baseado em distância euclidiana (Khaledi et al., 2018) para roteamento de dados para rede de Drones. O resultado desta adaptação foi **submetido** ao SWARMNET'20 (Afghah et al., 2020).

Dadas as etapas realizadas para o desenvolvimento da Tese e as publicações obtidas, a seguir elas serão detalhadas em ordem cronológica.

1. **ICOMP'17 - Towards an IoT Cooperation Model: understanding how a sociotechnical approach matters.** Conferência indexada com B1 na época da publicação. Este artigo foi o trabalho de uma disciplina de doutorado, no qual é apresentada a ideia de pesquisa a qual foram aplicadas técnicas de IHC/Engenharia de Software. O resultado foi uma visão mais ampla, em especial em relação ao impacto do que estava sendo proposto em relação à sociedade;
2. **LARS'17 - A Systematic Mapping for the Scenario of Non-Urban Autonomous Vehicle Cooperation Systems.** Conferência indexada como B3. Após a definição do

tema de pesquisa, praticamente em paralelo ao estudo desenvolvido e publicado no ICOMP'17, foi realizado um mapeamento sistemático para identificar como estavam as publicações envolvendo voo colaborativo de Drones aplicado a agricultura de precisão. Após a publicação, ainda continuaram havendo pesquisas e leituras de artigos para identificar novas publicações, porém sem a aplicação do mapeamento sistemático;

3. **COMPSAC'19 - A Comparative Approach on the use of Unmanned Aerial Vehicles kind of Fixed-Wing and Rotative Wing Applied to the Precision Agriculture Scenario.** Conferência é indexada como A2. Após a definição do cenário de pesquisa, foi necessário escrever uma seção apresentando as vantagens de um Drone sobre os veículos aéreos do tipo asa fixa. Foi sugerido na qualificação transformá-lo em artigo, assim, foram realizadas, além de comparações técnicas envolvendo os *datasheets* das aeronaves, as comparações reais de voo;
4. **WESAAC'20 - A Model based on Multiagent System and Cognitive Agents applied for Optimization of Group of Drones Flight.** Conferência indexada como B4. Após a definição do escopo do projeto, realizado um estudo sobre as tecnologias a serem utilizadas. Ainda em 2018, foram definidas a estrutura do Sistema Multiagente e a definição dos agentes cognitivos. O artigo ficou aguardando os resultados, que só foram obtidos no início de 2020, assim este artigo foi **submetido** ao WESAAC. Até o fechamento desta Tese, não havia o resultado da avaliação do artigo;
5. **COMPSAC'19 - Optimization System for Dynamic Flight Planning for Groups of Drones using Cooperation with Mobile Recharge Bases by means of Multiagent System and Recursive Auctions.** Conferência é indexada como A2. Este artigo foi a mais importante publicação, sendo o marco da qualificação. Com o objetivo de ter feedbacks do trabalho, este artigo foi submetido a três congressos A1 e suas contribuições foram acatadas, sendo ele aceito em uma conferência A2. Trata-se basicamente de um resumo do documento da qualificação, com os resultados preliminares e a aposta em relação ao algoritmo de otimização baseado em leilões recursivos;
6. **CEA'20 - Evaluation of a Dynamic Optimization Method based on Recursive Auctions for use in Flight planning of a Group of Drones applied to Precision Agriculture.** Revista indexada como A2. Este foi o primeiro artigo **submetido** para revista, além de ser o primeiro trabalho multi-área submetido. O objetivo deste trabalho é apresentar os resultados obtidos pelo algoritmo de otimização baseado em leilões recursivos para cenários agrícolas. No trabalho, simplificações foram realizadas para testar o desempenho do algoritmo e não necessariamente seu comportamento em ambientes abertos e dinâmicos. Até o fechamento desta Tese, não havia o resultado da avaliação do artigo;
7. **ICICT'20 - Mobile System for Optimized Planning to Drone Flight applied to the Precision Agriculture.** Conferência internacional sem qualis no Brasil, possui foco em informação e tecnologias computacionais. Neste, foi apresentado um aplicativo Android desenvolvido que executa o algoritmo de otimização proposto pela presente Tese. O aplicativo funciona como um passo a passo e substituiria o processamento em um computador. Ele é integrando com recursos como Google Maps e outras APIs de manipulação de mapas. Com o app, o usuário pode escolher no mapa a área que será sobrevoada, escolher a distância entre um ponto e outro, definir quantidade de Drones e

a posição inicial e o app gera a rota de voo, que pode ser enviado para o Drone por meio de uma mensagem JSON;

8. **WIDROID'20 - MultiDrone Simulator - An Open Source Multi-Plataform Tool to Use in Tests of Optimized Flight of Group of Drones.** Workshop internacional sem qualis, mas que acontece dentro do DCOSS (A2). Neste, foi **submetido** um artigo que apresenta com detalhes o simulador MultiDrone Simulator utilizado para os testes do MAOVD. No artigo, características como simulação on-line (passo a passo, mudando o valor de parâmetros) e off-line (sem a interação do usuário) são apresentadas. Até o fechamento desta Tese, não havia o resultado da avaliação do artigo;
9. **BSIDES'20 - Choosing the Best k-Neighbor Parameters for Optimizing a Flight of a Group of Drone using Recursive Auction Algorithm applied to the Precision Agriculture.** Este é um evento regional que aconteceu na cidade de Tampa-Fl. Neste artigo são apresentados os testes realizados para identificar o melhor valor de  $k$ , sendo este um parâmetro muito importante para o algoritmo de otimização baseado em leilões recursivos;
10. **BSIDES'21 - Applying the Methodology of the Nearest Neighbor and Area Division for optimizate group of Drones Flight through Recursive Auctions.** Este é um evento regional que aconteceu na cidade de Tampa-Fl. Este artigo apresenta duas metodologias para melhorar o tempo de processamento no algoritmo de otimização de voo de grupo de Drones, sendo a metodologia do vizinho mais próximo e a metodologia de divisão de área de atuação. **Este artigo está pronto, aguardando a chamada de trabalho de evento para submissão;**
11. **SWARMNET'20 - Greedy Optimization applied on Group Drones Flights: An Event-based Approach.** Este é um workshop sem qualis no Brasil, mas que acontece dentro do WOWMOM (A2). Como parâmetro de comparação do algoritmo de otimização baseado em leilões recursivo, foi utilizado o algoritmo proposto por Khaledi et al. (2018), porém, para uso no cenário da agricultura de precisão, algumas customizações foram necessárias, o qual justificou a proposta e submissão deste artigo, escrito em co-autoria da autoria do artigo principal utilizado nas comparações. Até o fechamento desta Tese, não havia o resultado da avaliação do artigo

## 7.6 CONCLUSÃO

Este capítulo apresentou os resultados do Modelo Aplicado de Voo de Grupo de Drones. Os resultados foram obtidos após o modelo proposto no Capítulo 5 ter sido executado com a aplicação das heurísticas e metodologias de otimização apresentadas no Capítulo 6.

Em uma comparação quantitativa em relação a outra otimização dinâmica presente na literatura, o algoritmo de otimização recursiva conseguiu resultados 30% melhores em cenários reais, com voo de múltiplos Drones. Considerando voo de um único Drone, a melhora foi mais sutil, com percentuais entre 7 e 8%.

Nos testes de abertura e dinamicidade, situações adversas foram simuladas para testar o Sistema Multiagente e os agentes cognitivos propostos e, em todas as situações, o modelo conseguiu se readequar aos eventos adversos. Esta abertura e dinamicidade foram comparados com outros 7 trabalhos presentes na literatura.

Após, é apresentada a instanciabilidade do modelo, apresentando outro cenário que poderia ser beneficiado com o Modelo Aplicado de Otimização de voo de Grupo de Drones.

Ao final, os resultados do ponto de vista de publicações são apresentados, divididos em publicações operacionais, das quais o autor deste trabalho foi colaborador e os trabalhos principais que abordam temas desta Tese.

## 8 CONCLUSÃO

Este capítulo apresenta as considerações finais do trabalho, assim como os seus encaminhamentos.

### 8.1 CONSIDERAÇÕES FINAIS

O presente trabalho propõe um modelo de cooperação para otimizar voos de Drones aplicado à agricultura de precisão. Os Drones cooperam com bases de recargas e seus pontos de passagem. A proposta desse modelo visa resolver limitações existentes para o uso de Drones no monitoramento de grandes áreas, assim como na elaboração de MDE.

A metodologia de pesquisa utilizada para o desenvolvimento da tese auxiliou na delimitação do escopo do problema, uma vez que após definido o tema de pesquisa, as técnicas de *Problem Solving* (Oulasvirta e Hornbæk, 2016), *Stakeholder Diagram* (Liu, 2000), *Evaluation Frame* (Baranauskas et al., 2000), *Value Pie* (Pereira e Baranauskas, 2014) e *Building Blocks of Culture* (Hall, 1959) consolidaram o cenário de pesquisa e o mapeamento sistemático adotado permitiu uma pesquisa por trabalhos relevantes de forma consistente e organizada.

Referente ao mapeamento sistemático, como ele foi realizado de forma ampla, foram verificadas outras possibilidades, além do foco principal deste estudo, o que resultou em 6 submissões/publicações de artigos em congressos ao longo desses quatro anos de estudo. Os artigos completos estão disponíveis para consulta nos Apêndices da presente tese.

Na sequência, foi definida uma estrutura organizacional dos elementos para o modelo baseado em Sistemas Multiagente. Já os agentes deste modelo são cognitivos, com sua codificação baseada nos estados mentais, o que garantem a abertura e a dinamicidade do modelo.

No que diz respeito a ambientes dinâmicos, cooperação entre veículos e técnicas de cooperação, vários artigos foram selecionados e lidos na íntegra, contribuindo consideravelmente para a proposta do presente modelo, bem como para a decisão de quais tecnologias seriam adotadas para sua implementação.

Para o Sistema Multiagente proposto, foi optado pelo Framework Moise+ que permitiu a abertura do modelo e foi utilizado o Framework JASON para a codificação dos agentes cognitivos, o que permitiu que o modelo funcionasse em ambientes dinâmicos. Para a troca de mensagens entre os agentes foi utilizado o protocolo FIPA Contract-NET, que permitiu que o modelo funcionasse independente da quantidade de elementos inseridos. Para a otimização foram utilizados leilões recursivos entre os pontos a serem sobrevoados, o quais permitiram encontrar os resultados ótimos de cada cenário.

O Sistema Multiagente proposto foi testado e estruturado com base no problema, porém, em um trabalho futuro, sua estrutura pode ser validada em outros problemas de otimização que possuem a característica de *planning*, tais como a otimização do uso da bateria de veículos elétricos em ambientes urbanos. Seu teste foi realizado por meio de ambiente de simulação MultiDrone Simulator, com a inclusão e exclusão de elementos do Sistema Multiagente durante sua execução.

A codificação de agentes cognitivos foi realizada em JASON, utilizando a linguagem AgentSpeak(L). Apesar de muito diferente das linguagens de programação imperativas, foi possível codificar agentes de software, mudando seus estados mentais dinamicamente e realizando a comunicação entre eles. A ferramenta Cartago permitiu o compartilhamento das crenças entre diferentes Agentes e os testes de dinamicidade foram realizados no ambiente MultiDrone

Simulator a partir de mudanças do ambiente, tais como velocidade/direção do vento ou solicitando o retorno precoce de Drones. Em todos os testes, o modelo conseguiu gerar novos planos de voos.

Já para algoritmo de negociação, foi optado pelo FIPA Contract-Net para o desenvolvimento de leilões entre os agentes no modelo. Esta técnica é bastante simples de ser aplicada em ambientes computacionais e como o próprio modelo se comporta como um grande leilão, este tipo de protocolo conseguiu ser aplicado sem dificuldades, funcionando para diferentes quantidades de agentes no modelo.

Nos testes, o algoritmo de otimização baseado em leilões recursivos, no seu modelo original, não apresentou bom desempenho do ponto de vista computacional, sendo necessária a aplicação de 3 heurísticas para diminuir o tempo de processamento, permitindo, assim, encontrar resultados factíveis em poucos segundos de processamento, sendo este um dos requisitos do modelo dinâmico no qual o modelo se aplica.

Para cenários maiores, também foram propostas três metodologias que levam o algoritmo a encontrar resultados factíveis em menor tempo de processamento. As três heurísticas, assim como as três metodologias de otimização foram aplicadas ao leilão recursivo para os testes de desempenho do modelo.

Em relação ao ponto de vista de otimização, o algoritmo de leilão recursivo conseguiu atingir resultados superiores em cenários simples, com um Drone, em relação a outro algoritmo de otimização dinâmica encontrado na literatura. Em relação a cenários maiores, formados por mais de um Drone, a melhora no resultado chegou a 30% em relação ao outro algoritmo de otimização dinâmica encontrado na literatura.

Os testes de abertura e dinamicidade aconteceram em ambiente de simulação, e em todos os testes, o modelo conseguiu se recuperar de eventos adversos, apresentando novas sugestões de plano de voo.

Foi possível, ao final desta Tese, verificar que o Modelo proposto se comporta como esperado e é uma promissora plataforma de pesquisa para uso de Drones em cenários da agricultura de precisão.

Dentre as principais limitações da pesquisa realizada, vale registrar que o modelo foi apenas testado em ambiente de simulação, e para que o mesmo seja embarcado e aplicado em um cenário real, novos estudos sobre a aplicabilidade do modelo em Drones reais devem ser realizados, sendo que até o fechamento da presente tese, apenas um trabalho detalha tal procedimento (Menegol et al., 2018).

## 8.2 ENCAMINHAMENTOS

Como encaminhamentos futuros, sugere-se a implementação de bases de cargas móveis no modelo. Na prática, o modelo já suporta tal característica, porém, do ponto de vista operacional, o algoritmo que define os planos de voo devem considerar também a mudança da base de recarga de posição, assim como o seu tempo de deslocamento.

Outra característica importante do modelo, porém, não aplicada nesta Tese, é a possibilidade de trabalhar com um número maior de Drones do que base de recargas, otimizando, entre os outros recursos, o pouso, o recarregamento e a decolagem dos Drones.

Em seguida, é possível iniciar o processo de embarcar os Sistemas Multiagentes em Drones e começar a fazer testes reais. Estes são possíveis e viáveis, como apresentando em Menegol et al. (2018), que embarcou a plataforma MOISE+ e JASON em um Drone.

Em paralelo, pretende-se abstrair do MAOVDG um modelo genérico e com a ajuda de um aluno do curso de Computação, pretende-se aplicar o modelo a problemática de roteamento de

automóveis elétricos, visando testar a instanciabilidade do modelo gerado a partir do MAOVGD, gerando, com isso, novos resultados.

Do ponto de vista desta Tese, serão realizados os encaminhamentos dos artigos ainda não publicados, ao total 6 artigos estão pendentes de publicações. A justificativa para isso é o tempo que os Journals levam para responder sobre a publicação e a crise do COVID-19, que cancelaram dois eventos no ano de 2020 nos Estados Unidos, nos quais, o autor desta Tese teve artigos submetidos, o que fez com que outros eventos similares fossem encontrados e realizado a re-submissão dos trabalhos.

Embora o trabalho ainda não está na fase de embarcar o software em Drones, quando isto ocorrer, tem-se a disposição no Programa de Pós Graduação em Informática da UFPR quatro Asas Rotativas modelo Phantom 4 da empresa DJI. Foram realizados alguns voos experimentais utilizando o ambiente de desenvolvimento DJI Developer SDK, que permite a comunicação com o Drone utilizando o Mobile SDK.

Para realizar os testes, se necessário, há ainda no Departamento Acadêmico de Informática da UTFPR-Câmpus Pato Branco 2 Drones modelo AR.Drone da empresa Parrot que permitem o desenvolvimento de software utilizando a API *Open Source* Java Drone.



## REFERÊNCIAS

- AERONÁUTICA (2015). Ica 100-40 sistemas de aeronaves remotamente pilotadas e o acesso ao espaço aéreo brasileiro . Em *Ministério da Defesa - Comando da Aeronáutica. Disponível em <https://www.decea.gov.br/static/uploads/2015/12/Instrucao-do-Comando-da-Aeronautica-ICA-100-40.pdf>*. AERONÁUTICA.
- Afghah, F., Brito, R. C., Weitzenfeld, A. e Todt, E. (2020). Optimization of group drone flights: A event based greedy approach. Em *Submtido para 2nd IEEE WoWMoM Workshop on Wireless Networking, Planning, and Computing for UAV Swarms*. SwarmNet 2020.
- Alvares, L. O. e Sichman, J. S. (1997). Introdução aos sistemas multiagentes. Em *XVII Congresso da SBC. Jornadas de Atualização em Informática - JAI*.
- ANAC (2017). Regras da anac para uso de drones. Em *Agência Nacional de Aviação Civil. Disponível em [http://www.anac.gov.br/noticias/2017/regras-da-anac-para-uso-de-drones-entram-em-vigor/release\\_drone.pdf](http://www.anac.gov.br/noticias/2017/regras-da-anac-para-uso-de-drones-entram-em-vigor/release_drone.pdf)*. Agência Nacional de Aviação Civil.
- Anthony, D., Elbaum, S., Lorenz, A. e Detweiler, C. (2014). On crop height estimation with UAVs. Em *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Institute of Electrical & Electronics Engineers (IEEE).
- Austin, J. (1975). How to do things with words. Harvard University Press.
- Avellar, G., Pereira, G., Pimenta, L. e Iscold, P. (2015). Multi-UAV routing for area coverage and remote sensing with minimum time. *Sensors*, 15(11):27783–27803.
- Baranauskas, M. C. C., Schimiguel, J., Simoni, C. A. C. e Medeiros, C. B. (2000). *Semiotics in Information Systems Engineering*. Proceedings of 11th International Conference on Human-Computer Interaction.
- Bastianelli, G., Salamon, D., Schisano, A. e Iacobacci, A. (2012). Agent-based simulation of collaborative unmanned satellite vehicles. Em *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*. Institute of Electrical & Electronics Engineers (IEEE).
- Brandão, A. S., Martins, F. N. e Soneguetti, H. B. (2015). A vision-based line following strategy for an autonomous uav. Em *2015 IEEE 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. Institute of Electrical & Electronics Engineers (IEEE).
- Brito, R. C., Borsoi, B., da Rocha Campos, J. R., Weitzenfeld, A. e Todt, E. (2020a). Evaluation of a dynamic optimization method based on recursive auctions for use in flight planning of a group of drones applied to precision agriculture. Em *Submtido para Computer and Electronics in Agriculture*. Elsevier.
- Brito, R. C., Favarim, F., Weitzenfeld, A. e Todt, E. (2020b). Multidrone simulator - an open source multi-plataform tool to use in tests of optimized flight of group of drones. Em *Submtido para 2020 IEEE 2nd International Workshop on Wireless Sensors and Drones in Internet of Things (WI-DROID 2020)*. IEEE.

- Brito, R. C., Lorencena, M. C., Loureiro, J. F., Favarim, F. e Todt, E. (2019a). A comparative approach on the use of unmanned aerial vehicles kind of fixed-wing and rotative wing applied to the precision agriculture scenario. Em *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE.
- Brito, R. C., Loureiro, J. F., Guedes, A. e Todt, E. (2019b). Optimization system for dynamic flight planning for groups of drones using cooperation with mobile recharge bases by means of multiagent system and recursive auctions. Em *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE.
- Brito, R. C., Loureiro, J. F. e Todt, E. (2020c). Choosing the best k-neighbor for optimizing a flight of a group of drone using recursive auction algorithm applied to the precision agriculture. Em *7th Annual BSides Tampa IT Conference*. BSides.
- Brito, R. C., Loureiro, J. F., Todt, E. e Pereira, R. (2017a). A systematic mapping for the scenario of non-urban autonomous vehicle cooperation systems. Em *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*. IEEE.
- Brito, R. C., Priester, L. F. e Todt, E. (2020d). Applying the methodology of the nearest neighbor and area division of for optimizate group of drones flight through recursive auctions. Em *Submetido para 7th Annual BSides Tampa IT Conference*. IEEE.
- Brito, R. C., Priester, L. F. e Todt, E. (2020e). A model based on multiagent system and cognitive agents applied for optimization of froup of drones flight. Em *Submetido para 14th Workshop-School on Agents, Environments, and Applications*. IEEE.
- Brito, R. C., Rabello, A., Favarim, F., Weitzenfeld, A. e Todt, E. (2020f). Mobile system for optimized planning to drone flight applied to the precision agriculture. Em *2020 The 3rd International Conference on Information and Computer Technologies*. ICICT.
- Brito, R. C., Todt, E., Pereira, R. e Loureiro, J. F. (2017b). Towards an iot cooperation model: Understanding how a sociotechnical approach matters. Em *The 18th International Conferenece on Internet Computing and Internet of Things (ICOMP'17)*. Institute of Electrical & Electronics Engineers (IEEE).
- Caska, S. e Gayretli, A. (2015). An algorithm for collaborative patrolling systems with unmanned air vehicles and unmanned ground vehicles. Em *2015 7th International Conference on Recent Advances in Space Technologies (RAST)*. IEEE.
- Chen, T., Campbell, D., Gonzalez, L. F. e Coppin, G. (2015). Increasing autonomy transparency through capability communication in multiple heterogeneous UAV management. Em *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.
- Chen, Y., Zhang, H. e Xu, M. (2014). The coverage problem in UAV network: A survey. Em *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. IEEE.
- Chirwa, R. M. N. e Lauf, A. P. (2014). Performance improvement of transmission in unmanned aerial systems using multipath TCP. Em *2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE.
- Cormen, T. H. (2009). *Introduction to Algorithms*. MIT Press, Massachusetts.

- de Oliveira Zamberlam, A. e Giraffa, L. M. M. (2001). Modelagem de agentes utilizando a arquitetura bdi. Em *Technical Report Series*. Faculdade de Informática – PUC-RS.
- Dyba, T., Dingsoyr, T. e Hanssen, G. K. (2007). Applying systematic reviews to diverse study types: An experience report. Em *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*. Institute of Electrical and Electronics Engineers (IEEE).
- Embrapa (2016). Programa nacional de solos do brasil (pronasolos). *Embrapa Solos*.
- Ergezer, H. e Leblebicioğlu, K. (2013). 3d path planning for multiple UAVs for maximum information collection. *Journal of Intelligent & Robotic Systems*, 73(1-4):737–762.
- Ernest, N. D., Cohen, K. e Schumacher, C. J. (2013). UAV swarm routing through genetic fuzzy learning methods. Em *AIAA Infotech@Aerospace (I@A) Conference*. American Institute of Aeronautics and Astronautics.
- Favarim, F., Linares, K., Brito, R. e Todt, E. (2018). Comparison analysis between pid and fuzzy logic controllers for quadrotor in a simulated and in a real environment. Em *SBR-LARS 2018* ().
- Feo, T. A. e Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133.
- Filho, M. M., Gonçalves, D. A., Brito, R. C. e Todt, E. (2018). Sistema web para gerenciamento de vôos de aeronaves não tripuladas. Em *Medianeira in Technology 2018*. MEDITEC.
- Finin, T., Fritzson, R., McKay, D. e MCentire, R. (1994). Kqml: A language and protocol for knowledge and information exchange. Em *International Workshop on distributed Artificial Intelligence*.
- FIPA (2002). Fipa contract net interaction protocol specification. Em [www.fipa.org/specs/fipa00029/SC00029H.pdf](http://www.fipa.org/specs/fipa00029/SC00029H.pdf). Acessado em 10 de julho de 2018.
- FIPA (2012). Foundation for intelligent physical agent - agent communication language. Em <http://www.fipa.org/repository/ips.php3>. Acessado em 10 de julho de 2018.
- George, E. A., Tiwari, G., Yadav, R. N., Peters, E. e Sadana, S. (2013). UAV systems for parameter identification in agriculture. Em *2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS)*. Institute of Electrical & Electronics Engineers (IEEE).
- Hall, E. (1959). *The Silent Language*. NY: Anchor Book.
- Ho, D.-T., Grotli, E. I., Sujit, P. B., Johansen, T. A. e Sousa, J. B. (2013). Cluster-based communication topology selection and UAV path planning in wireless sensor networks. Em *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE.
- Hübner, J. F., Bordini, R. H. e Vieira, R. (2004). Introdução ao desenvolvimento de sistemas multiagentes com jason. Em *XII Escola de Informática da SBC, volume 2*. SBC.
- Hübner, J. F. e Sichman, J. S. (2000). Uma ferramenta para implementação e monitoração da comunicação entre agentes. Em *Open Discussion Track. Ibero-American Conference on AI, 15th Brazilian Symposium on AI*.

- Hübner, J. F. e Sichman, J. S. (2003). Organização de sistemas multiagentes. Em *Jornada de Atualização em Inteligência Artificial*. JAIA.
- Khaledi, M., Rovira-Sugranes, A., Afghah, F. e Razi, A. (2018). On greedy routing in dynamic UAV networks. Em *2018 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*. IEEE.
- Kitchenham, B. e Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Relatório Técnico EBSE 2007-001, Keele University and Durham University Joint Report.
- Kleinberg, J. e Tardos, E. (2005). *Algorithm Design*. Pearson, 1st. edition.
- Lemaître, C. e Excelente, C. B. (1998). Multi-agent organization approach. Em *II Iberoamerican Workshop on DAI and MAS*.
- Liu, K. (2000). *Semiotics in Information Systems Engineering*. Cambridge University Press (CUP).
- Loureiro, J. F., Brito, R. C., Magrin, C. E. e Todt, E. (2019). A model for landing, taking off and autonomous battery recharging of a parrot ar.drone 2.0 using computational vision and gps features. Em *2019 Computer on the Beach*. COTB.
- Menegol, M. S., Hübner, J. F. e Becker, L. B. (2018). Evaluation of multi-agent coordination on embedded systems. Em *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*, páginas 212–223. Springer International Publishing.
- Miranda, E. (2005). Brasil em relevo. *Embrapa Monitoramento por Satélite*.
- Netto, J. O., Favarim, F., Brito, R. C., Priester, L. F. e Todt, E. (2020). Implementing a communication network between bases station applied for group of drones. Em *Submetido para 2020 IEEE 29th Annual Software Technology Conference (STC)*. IEEE.
- Oulasvirta, A. e Hornbæk, K. (2016). HCI research as problem-solving. Em *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*. Association for Computing Machinery (ACM).
- Pereira, R. e Baranauskas, M. C. C. (2014). Value pie: A culturally informed conceptual scheme for understanding values in design. Em *Lecture Notes in Computer Science*, páginas 122–133. Springer Science Business Media.
- Petersen, K., Vakkalanka, S. e Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18.
- Pippin, C. E. e Christensen, H. (2012). Cooperation based dynamic team formation in multi-agent auctions. Em *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III*. SPIE.
- Puhl, L., Favarim, F., Pegorini, V. e Torrico, C. (2018). Desenvolvimento de robô trekking autônomo integrando sensoriamento inercial, geoespacial e visual. Em *2018 Computer on the Beach*. Meditec.
- Rao, A. S. (1996). Agentspeak(1): Bdi agents speak out in a logical computable language. Em *Seventh Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW)*.

- Roberge, V., Tarbouchi, M. e Allaire, F. (2014). Parallel Hybrid Metaheuristic on shared Memory System for real-time UAV Path Planning. *International Journal of Computational Intelligence and Applications*, 13(02):1450008.
- Saque, N., Brito, R. C., Goncalves, D. A., Favarim, F. e Todt, E. (2019). Optimized division of exploration areas in multi-robot systems considering static and dynamic charging stations. Em *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE.
- Searle, J. R. e Vanderveken, D. (1985). Foundations of illocutionary logic. Cambridge University Press.
- Sipser, M. (2007). *Introdução a Teoria da Computação*. Thomson, São Paulo, 2nd. edition.
- Valeriano, M. D. M. (2005). Modelo digital de variáveis morfométricas com dados srtm para o território nacional: o projeto topodata. *XII Simpósio Brasileiro de Sensoriamento Remoto*, 119:19–32.
- Vidal, J. M. (2006). Fundamentals of multiagent system. Em <http://www.scribd.com/doc/info/2094479?accesskey=key-1b77wmyq0mf8i6tefdn6>. Acessado em 10 de julho de 2018.
- Wikipedia (2019a). Facility location problem. [Online; Acessado em 09 de abril de 2019].
- Wikipedia (2019b). Hamiltonian path. [Online; Acessado em 09 de abril de 2019].
- Wikipedia (2019c). Travelling salesman problem. [Online; Acessado em 09 de abril de 2019].
- Wooldridge, M. (1997). Agent-based software engineering. Em *Proc Software Engineering. number 1*. Institute of Electrical & Electronics Engineers (IEEE).

**APÊNDICE A – ARTIGOS SUBMETIDOS/PUBLICADOS**

# Towards an IoT Cooperation Model: understanding how a sociotechnical approach matters

Robison Cris Brito  
Federal University  
of Technology - Paraná  
Campus Pato Branco  
robison@utfpr.edu.br

José Felipe Loureiro  
Federal University  
of Technology - Paraná  
Campus Pato Branco  
zefelipe\_0@hotmail.com

Eduardo Todt  
Federal University  
of Paraná  
UFPR  
todt@inf.ufpr.br

Roberto Pereira  
Federal University  
of Paraná  
UFPR  
rpereira@inf.ufpr.br

## Abstract—

Designing solutions for the IoT domain has been recognized as a challenging and complex scenario for which our technical-centered approaches seem not to hold anymore. This paper draws on a real project experience to discuss how a sociotechnical perspective to the development of an IoT solution can support a better understanding of a project, including its scope, problem, solution, and impacts. The research project considered aims at developing a collaboration model among things, for instance UAVs, in the IoT context. First, the problem understanding was focused on formal aspects of the model as well as functional requirements of IoT as a platform and its technical viability. Second, the problem was discussed from a sociotechnical perspective, informed by different artifacts and methods from Human-Computer Interaction and Information Systems domains. This paper presents a comparison between the first and second moments, presenting both the artifacts and their main outcomes, highlighting critical issues not being considered in the first attempt of understanding the problem, but brought to light when sociotechnical artifacts and methods were applied. Thus, the usefulness and viability of using sociotechnical artifacts in the development of a IoT solution was enlightened.

*Keywords—Research problem, HCI, Stakeholder Diagram, Assessment Framework, Value Pie.*

## I. INTRODUCTION

A major challenge to researchers and developers around the world is to find mechanisms to understand and visualize a research problem from different perspectives, from other paradigms or references. This practice facilitates the solution of the problem as a whole, or at least, subsidizes the development of a partial solution to this problem.

Among the known techniques, one that deserves to be highlighted is to observe the problem from a Human-Computer Interaction (HCI) perspective, not necessarily thinking about the graphical user interface, which commonly refers to this acronym, but rather to the impact of this solution to Stakeholders affected by the solution of the problem: a sociotechnical, or socially-aware, perspective.

In order to consider such an impact and demonstrate the contributions of a sociotechnical perspective, this paper presents a research problem in the IoT area: the use of several Unmanned Aerial Vehicles (UAV), which communicate/negotiate to achieve a common goal (i.e., optimizing the time to perform a specific task, such as monitoring an

agricultural area). Such cooperation allows the optimization of the flight time of the UAVs, as well as the expenses, the wear of the devices, and the availability, among the optimization of many other resources.

Initially, the problem was approached in a traditional way: focusing on the model to be developed for a generic solution of the problem, considering the IoT from the perspective of a technical platform. In a second moment, this same research problem was supported by HCI and Information Systems techniques specific for sociotechnical analysis.

The trigger for the second moment was a mindset changing in perspective for IoT: from starting thinking of network infrastructure, the things to be linked, their sensing and actuation capabilities, to start thinking of the stakeholders, their possible needs and responsibilities, and their social world. Therefore, the first changing was looking at our research problem from a HCI Problem Solving [1] perspective: methodology based on the philosophy of Larry Laudan [2], which focus on identifying a problem and investigating solutions to solve the problem (or to augment our capacity of solving it) supported by computing technology, rather than restricting the discussion on whether a given solution is valid or not.

From the sociotechnical artifacts and methods applied, the first one was Stakeholder Diagram (SD) which, according to [3], supports the identification of all stakeholders directly or indirectly affected by the problem being considered and by solution to be designed. With this artifacts, the project is divided into layers that focus on interaction with stakeholders, from the more informal level (stakeholders present in society as a whole, for example) to the more formal levels (stakeholders who will technically develop the solution). In this perspective, the researcher is forced to see the research problem from the optics of each stakeholder present in each layer.

To complement the Stakeholder Diagram, the Evaluation Frame ([4]) was applied to anticipate issues and problems for each stakeholder, and to propose possible solutions to them.

Finally, a conceptual model called Value Pie ([5]) was used to look at the problem setting from a different dimensions, such as formality and culture. The Value Pie was built on the basis of the Organizational Semiotics theory and the Building Blocks of Culture ([6]), thus including a view of the system from the point of view of Interaction, Association, Learning, Game, Protection, Exploration, Temporal, Territorial, Classification

and Subsistence aspects. Each of these layers is analyzed informally (as the society sees the layer), formally (as it is formalized, via documents, laws, for example) and technically (how a system can be implemented to make it operational). Together, the 3 artifacts provided a critical, deeper and wider understanding of the problem being discussed and the solution to be designed.

This paper initially presents the research problem, as well as a brief literature review. First the problem is presented as approached in the traditional way, and then, with the support of the aforementioned sociotechnical techniques. As results, the paper presents a comparison between the two approaches, followed by a discussion on the impact of the sociotechnical techniques in the project, mainly for opening new horizons regarding the research problem.

## II. RESEARCH PROBLEM

The Internet of Things, or IoT, arose from the concept of pervasive computing, introduced in 1991 by Mark Weiser ([7]). According to the author, the most important technologies are those whose perception disappear and are so intertwined with the daily lives of people that they go unnoticed.

In order to encourage IoT, it is necessary to create saturated environments of computational capacity and communication, and at the same time, that these environments are subtly integrated with the user ([8]). With the existing technological capability, it is possible to develop applications using this concept.

According to ([9]) there is no widely accepted definition for IoT. The first time the acronym was used was in 1998, by Kevin Ashton. It describes IoT as a global Internet-based information services architecture, integrating Hardware and Software into a broadly integrated environment. This understanding for IoT, however, focus on the technical and does not account for the "people" present in Weiser's thoughts.

Technically, the IoT architecture is based mostly on things that communicate over wireless networks. Some features can be used to enrich IoT solutions such as Radio Frequency Identification (RFID), Global Position System (GPS), Electronic Data Interchange (EDI), Geographic Information System (GIS), Just-in-Time (JIT), among others. IoT can also be applied in different scenarios, prevailing dynamic and heterogeneous environments. For the present work, the chosen scenario was the collaboration between unmanned aerial vehicles, popularly known in Brazil as Drone.

Drone, Quadricopter, Unmanned Aerial Vehicle (UAV), Remotely Piloted Aircraft Systems (RPAS) or Unmanned Aerial Vehicle (UAV) are some of the name variations for remotely controlled and / or autonomous unmanned aerial vehicles. They were originally a hobby, but with the evolution and accessibility of new technologies, what was called Aeromodelling, today is treated with seriousness, since its applications have been growing in the most diverse areas, as: aid in rescue of people, obtaining aerial images for mapping, agricultural management, among others.

A UAV, to generalize the various denominations assigned to unmanned aerial vehicles, is an equipment that needs to be handled with care, a fact that has made it difficult to

homologate in several countries. Its propellers that rotate at high speeds and their weight, which can vary from a few grams to several kilograms, can cause serious injuries in the event of accidents or falls. In some countries, such as Brazil, for approval, autonomous flight can only occur in non-populated areas. [10]

To minimize risks, the Brazilian Air Force (FAB) published in late 2015 the new legislation on commercial use of UAVs, which promises to boost (and bring greater security) its commercial use in Brazil. The Brazilian regulations follow the line of action adopted by the International Civil Aviation Organization (ICAO), based on amendments to the annexes of the Chicago Convention. Nevertheless, legislation that addresses the use of Brazilian airspace by remotely piloted aircraft must be constantly revised and adjusted, given the dynamic nature of the activity and the recurrent technological advances [10].

UAV flights can occur in two ways: remotely controlled or autonomous. In this second, there are the biggest advantages and biggest challenges. In the stand-alone flight there is no limitation on the distance between the UAV and its controller. In controlled flight, this control often occurs via radiofrequency, which is limited to transmitter power, not exceeding a few kilometers. In this way, in autonomous flights it is possible to cover a larger area, as well as a higher altitude of flight, making the search for information through sensors or cameras attached to the UAV more efficient than in controlled flights.

In this way, the use of autonomous UAV stands out in the scenario of precision agriculture. According to [11], the concept of precision agriculture is usually associated with the use of high technology equipment to assess or monitor conditions on a given parcel of land, and then apply in accordance, the various factors of production (seeds, fertilizers, water, etc.). Thus, based on specific data from geographically referenced areas, the process of agricultural automation is implemented, and inputs are proportionally measured for each area, unlike traditional agriculture, where the same amount of input is used throughout the area to be treated.

The importance of UAVs in precision agriculture is highlighted by authors such as [12] who record that UAVs provide the best platform for evaluating agricultural production, being fast and efficient when compared to traditional terrestrial techniques, allowing reading on virtually all types of terrain. [13] complement, saying that UAVs are versatile devices that can be used in a wide variety of agricultural applications, such as: automatic monitoring of plantings and water channels, irrigation, pest detection and control, and livestock, counting the animals in a pasture and identification of their positions.

Also according to [13], it is in autonomous flights that UAVs stand out in the agricultural environment, since they are often used in the monitoring and inspections of large areas of planting. [14], whose work proposes the use of UAVs in agriculture to monitor plant growth, also highlights the use of autonomous operating air vehicles, as it can read data and through specific algorithms, perform the classification of the data, without the need of the intervention of a specialist user. With autonomous flights and processing on the device itself, the amount of data processed, as well as the area analyzed, can be much larger.

A limitation of autonomous flights, especially for the



monitoring of a large extent of cultivated land, is the use of a single UAV. As one of the major limitations of low-cost UAVs are batteries. Due to the short duration of the battery charge, relatively long flights become nonviable, as is the monitoring of a large area.

A possible solution to this kind of situation is to use multiple collaborating UAVs to optimize the monitoring of a large area, ensuring speed and creating means of contingency, so if a problem occurs with one of the UAVs during the monitoring, others take over the tasks of the previous. But the great advantage is in optimizing the use of resources, in this case, the flight time and the use of the battery. Considering the limitations of the batteries, the less time of flight to do a certain monitoring, the greater the effectiveness of the system.

The advantages of multiple UAVs in precision agriculture are many, whether for speed, the possibility of reaching difficult access points (through rugged terrain or dense vegetation), agility in reading information, and since they are located above ground they can travel at higher speeds than land vehicles. In addition, UAVs can move up or down, allowing the acquisition of images from a larger area for a more comprehensive analysis or closer images, for a more detailed analysis. These data are digital and can be processed individually, which ensures a greater amount of information to be applied to precision farming techniques.

Technological evolution allows an increasing autonomy of the batteries of the smaller UAVs, which are usually cheaper and more attractive to be used in precision agriculture. Even for longer flights, there are larger UAVs with combustion engines, with a few hours of flying time, but in these cases they are less accessible due to high acquisition and maintenance costs.

These larger UAVs can also act in the environment, be it in the spraying process, seed dispersion, physical sample collection of soil or plant tissue for analysis in laboratories. In short, the possibilities are many, especially highlighting the acquisition of images from cameras coupled to the air vehicle. These images can be processed at the time of capture or stored and sent for remote processing on a more robust computer, since in some situations complex image processing logic is required for data collection.

### III. TRADITIONAL APPROACH X HCI BASED APPROACH

In order to choose a research problem, traditionally, we try to develop a generic model to take advantage of it not only in a specific scenario, but also in other possible scenarios.

The research problem in question would deal with the creation of a mechanism for collaboration of things that cooperate with each other in order to achieve a common goal.

By abstracting the model, leaving it more generic, it becomes very flexible, having as its advantage its application in very different scenarios, as in the collaboration of UAVs for autonomous flights applied to precision agriculture, or in the control of road flow of a large city, or even to identify the production cycle of a factory, finally, it can be observed that in all these, the elements can be modeled as agents, one of the challenges being the dynamic creation of groups, as well as identifying which is the best model of collaboration between them.

Traditional Model	Model using Problem Solving
Generic model(abstract)	Specific solution
Need to validate the model	Need to solve the problem
Approach Top-Down	Approach Botton-Up
Increase in complexity	Decrease of complexity

TABLE I. COMPARISON TABLE BETWEEN THE TRADITIONAL APPROACH AND THE APPROACH USING PROBLEM SOLVING.

Applying the problem solving technique to the research problem, a single scenario with well defined problems was used, and this was the collaboration of UAVs applied to precision agriculture. Thus it was possible to focus on the tools, models and techniques to solve the problem in this scenario, the goal not being to make the model generic, but to make it as specialized as necessary to solve this problem, or to at least advancing our capacity to solve it.

Table I shows how the approach was before the perspective of Problem Solving (Traditional Model) and after its use (Model using Problem Solving).

With Problem Solving, the concern is no longer the elaboration and validation of a generic model, but rather the development and implementation of a system to solve a very specific problem. Thus, the model has become Botton-Up (from specific to generic), initially applied to the UAV collaboration scenario, and as far as possible generalized to other scenarios, which considerably reduces the complexity for its Implementation and validation.

After defining the scenario in which the solution was proposed, the next step was the identification of the stakeholders involved in the problem-solution context. In the traditional model, only two stakeholders had been identified: the user of the system (the person who would define the mission of the UAV group) and the precision agriculture company that would contract the system (which would provide subsidies for the solution implementation).

After applying the Stakeholder Diagram, other key stakeholders were identified to work out the solution. For the elaboration of the Stakeholders Diagram, the DSC tool [15] was used. The environment has a graphical interface to define the stakeholders responsible for each of the five layers of interaction: Operational, Contribution, Source, Market and Community. The graphical environment interface is shown in Figure 1.

Table II presents the five layers of interaction of the Stakeholder Diagram, as well as the stakeholders identified for each interaction layer.

As can be seen in the table II, a stakeholder may be present in different layers of interaction, since its contribution can happen in different ways. In the Operational layer, which is the most formal of all, there are stakeholders such as undergraduate students who will work on the development of more operational activities, such as programming UAVs, elaborating the bases for autonomous recharge, among others. The masters/doctoral students will contribute in their researches to the construction of the model applied to the identified problem.

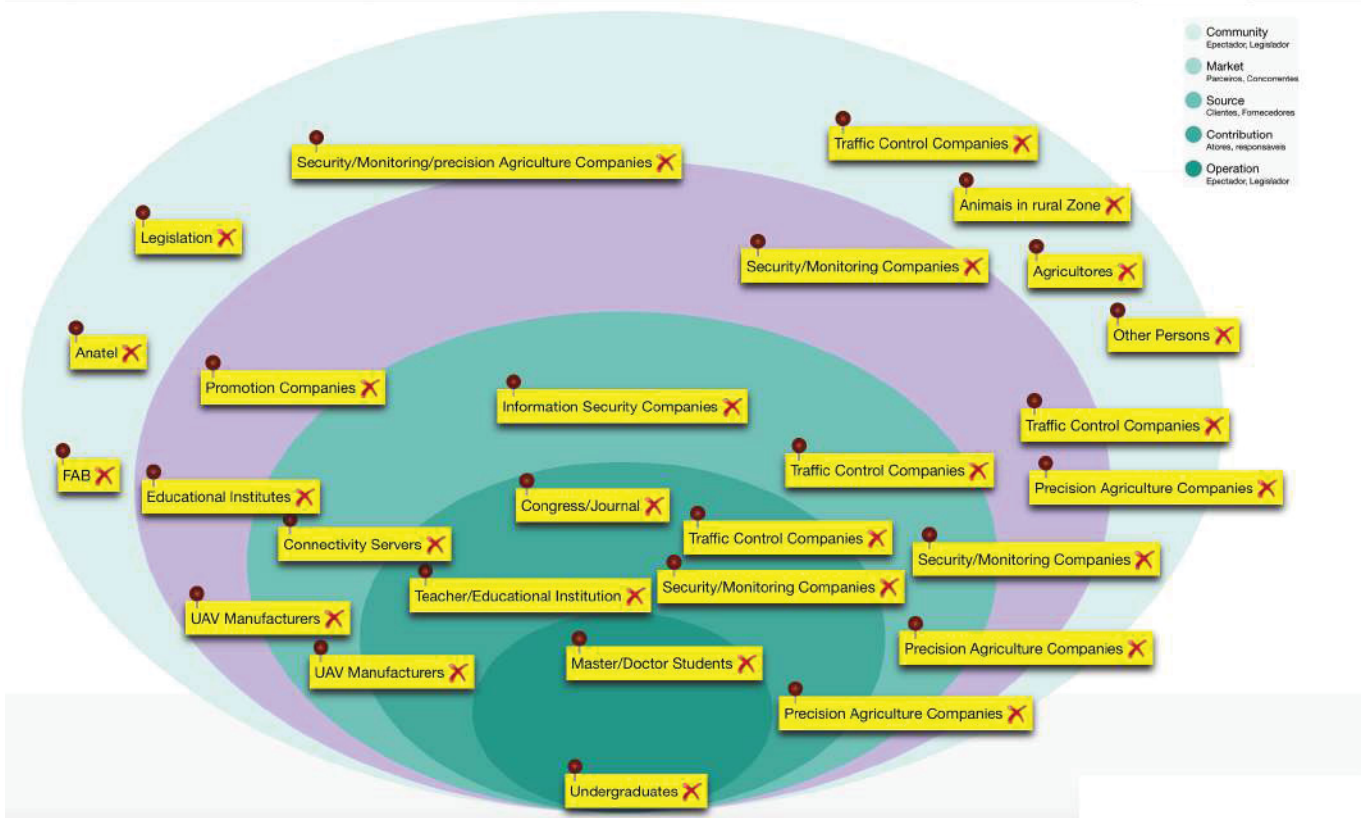


Fig. 1. Outline of the Visual Environment for the development of the Stakeholder Diagram

Layer	Identified Stakeholders
Operational	Undegraduates, Masters/Doctors students
Contribution	Teachers/Educational Institution, Congresses/Journals, Precision Agriculture companies, Traffic control companies, Security/Monitoring companies
Source	Drone manufacturers, Connectivity servers, Information Security companies, Precision Agriculture companies, Traffic control companies, Security/Monitoring companies
Market	Drone manufacturers, Educational Institutes, Promotion companies, Precision Agriculture companies, Traffic control companies, Security/Monitoring companies
Community	FAB, Anatel, Legislation, Animals in rural zone, Farmers, Other persons

TABLE II. IDENTIFIED STAKEHOLDERS TABLE FOR EACH INTERACTION LAYER.

In the Contribution layer, there are stakeholders who will contribute by providing information to improve the system, especially educational institutions, congress papers/journals, as well as companies that can make use of this solution. Emphasizing that the latter is present in the other layers of the Stakeholder Diagram. In addition to Precision Agriculture companies, traffic control companies, as well as Security and Monitoring companies, may share some of their knowledge in

order to improve this solution.

Under Sources, besides the companies that will be able to make use of the solution, are also possible suppliers of services and equipment, like companies that will provide the service of wireless connectivity between the UAVs, as well as those that manufacture UAVs.

Under Market are potential clients for the system, with companies already mentioned in the previous layers and educational institutions, which will have access to the results of the research carried out for the development of new researches.

Finally, the Community layer is the most informal of all, i.e., has indirect impact on the problem-solution context, which has in particular the regulators of the operations of flight (FAB in Brazil), wireless communication (ANATEL in Brazil), federal laws on the commercial use of this type of solution, in addition to rural animals, farmers and other people living in the rural area, because the system has to have the least possible impact on the day to day of these stakeholders.

For the development of the Evaluation Frame the DSC tool [15] was also used. The objective of the Evaluation Frame is to assign to each stakeholder at least one problem caused by the envisaged solution to it, as well as proposals to solve this problem. The contribution of this tool in the project is very important, because it allows the researcher to watch for problems, identifying possible needs and solutions before an actual solution is designed.

Table III presents a sample from the Evaluation Frame.

	questions/problems	idea/solution
FAB	Lack of regulamentation for Drones use in urban ambients.	Immediately use the project only in rural areas. So that when its regulated, application usage can also occur in urban environments (eg highway monitoring).

TABLE III. EVALUATION FRAME OF AN THE IDENTIFIED STAKEHOLDER. .

Traditional Model	Model using Stakeholders Diagram and Evaluation Frame
Little interaction with the system	Lots of interaction with the system
Need of an operator with little training	Need of an operator well trained
Few actors involved in the system (operator and company that hire the service)	Various actors, in different layers
Analysis based on opportunities	Analysis based on difficulties
The company that needs a service is only a client (Market)	The company that needs a service act in the layers of Contribution, Source, Market and Community.
Scientific vision	Commercial and scientific vision

TABLE IV. COMPARISON BETWEEN THE TRADITIONAL MODEL AND THE MODEL WITH THE CONTRIBUTIONS OF THE STAKEHOLDER DIAGRAM AND THE EVALUATION FRAME. .

To illustrate, a problem has been presented that affects the FAB stakeholder, present in the community layer, as well as a possible solution to this problem. In all, there were 24 questions/problems and idea/solution.

In the Evaluation Frame, the researcher must observe the system from the perspective of each one of the identified stakeholders, verifying the limitations they impose on the system, as well as presenting possible solutions to overcome such limitations and turn the project feasible.

Table IV presents a comparison of the contributions provided by the Stakeholder Diagram and the Evaluation Frame to the initial proposal, which is called in the Traditional Model Table.

In the traditional model it was not possible to identify many stakeholders, which was instigated in the creation of the Stakeholder Diagram. Also, in the original model, it was imagined that the operator that would make use of the system could be any user with some training, however, after the Evaluation Frame it was identified that this operator has to be very experienced and of suitable age, once that in the occurrence of an accident involving the UAVs monitored by him, the civil and criminal liability of this accident is the operator's.

The Evaluation Frame also instigated a more critical view of the project, considering the possible difficulties of its adoption, allowing the developed project to become a commercial product or a possible patent. The artifact allowed us to look at the problem from the perspective of different stakeholders.

Finally, another important contribution came from Value

	Informal	Formal	Technical
Interaction	Know the system	Be liable	
Association	The user validates or reorganizes the flight mission		Suggestion of UAVs grouping
Learning	Know-how of the operator		Base of knowledge (Historic)
Game		Gamefication	UAV scores per route
Protection		Lack of Regulation	Cryptography on message exchange
Exploration	User can request the return	UAV doesn't fly with wind/rain	
Temporal		Physical limitations Size/Autonomy	
Territorial	Drones trustworthy in risk areas	Don't fly over urban area	
Classification		Resources of the Drone	
Subsistence	Risk of Unemployment		

TABLE V. VALUE PIE OF THE PROPOSED PROJECT.

Pie, where the project was analyzed according to the 10 dimensions proposed by Hall. Each of these dimensions are discussed from three perspectives: informal, formal and technical, as can be seen in Table V.

With this technique, the design was observed in each of the 10 layers. It has been considered informal how society sees the layer, formal how the layer is viewed by documents and laws, and how technical the way the project can be implemented to make it operational. In sequence each of its layers will be detailed:

- 1) Interaction: the user of the (operational) system must, besides knowing the system, be liable to answer judicially for its decisions.
- 2) Association: The system user can organize groups of UAVs to an exploration, although the system must give grouping suggestions based on the knowledge it has (historical).
- 3) Learning: user knowledge is very important to operate the system, however, the system has a knowledge base that will help the user to make the best decision.
- 4) Game: it was noted that it could use Gamefication to classify UAVs based on the amount of points each gained over its use. These points are obtained when the UAV behaves as expected or exceeds expectations
- 5) Protection: although there is no regulation for urban flights, it does not prevent the use of UAVs in precision agriculture (rural flights), and data transported by the system can be protected using protocols and tools for encryption.
- 6) Exploration: UAVs are not indicated for flights in rainy or windy conditions, thus, it was identified the need of the user to request the return of the UAVs during a mission, if there is any climatic change that compromises the equipment, as well as to find alternatives if these situations occur.
- 7) Temporary: Every UAV has a flight autonomy imposed by its battery or the amount of fuel it has, so this autonomy compromises the flight time of the UAV and must be periodically re-evaluated (for battery replacement or engine adjustment).

Traditional Model	Model using Value Pie
Lack of informal analysis (barely adressed in technical papers)	All 10 Hall dimension consider an informal point of view
Didn't consider the legal parameters of flight	Identified many gaps in the laws for drone use
Exclusive technical concerns	Legal, social and technical concerns
Technical operator for the system	Specialized operator for the system. Legal liability.
One specific problem	Possibilities to explore many problems (business oportunities)

TABLE VI. COMPARISON BETWEEN TRADITIONAL MODEL AND VALUE PIE MODEL.

- 8) Territorial: During flights, it is normal for the UAV to overfly forests or flooded areas, which would make it difficult to recover the equipment if a crash occurs. In this way, more reliable UAVs should be sent to these types of locations given their history. Also, flights in urban areas are prohibited by the FAB and should not be allowed.
- 9) Classification: UAVs should be classified in the system based on some of their characteristics, such as autonomy, camera quality, present sensor quality, size and weight.
- 10) Subsistence: mechanisms should be sought so that the social impact of the use of UAV solutions is as small as possible, since "manual" work will be replaced by more skilled work (system operators and people who will maintain UAVs).

After these surveys, it was possible to perform an analysis comparing the Traditional Model and the proposed model after the application of the Value Pie technique, being the comparison presented in VI.

It was from the analysis of the Value Pie that the project received a very significant contribution from the Informal layer, which contributed to the 10 Hall layers. From the formal point of view, the project also receives major contributions, especially regarding the laws and regulations for the use of UAVs.

Another great advantage of using Value Pie is the peripheral contributions that the project can bring, not necessarily thinking about the proposed project, the focus of this work, which is the cooperation between UAVs to achieve a common goal in precision agriculture, but the emergence of other opportunities in this same scenario, such as the creation of a model to regulate the flight of UAVs in urban areas, the need for an identification of origin for UAVs (similar to the existing plates in cars), so that in case it cause some damage, that the owner and/or its origin can be discovered. Also the likely need for flight controller services for UAVs, similar to what happens in civil aviation, among others.

#### IV. CONCLUSION

This paper presented the contributions of some sociotechnical artifacts and methods from HCI and Information Systems

when applied in a research problem, in order to facilitate its understanding and resolution, as well as to present new opportunities and challenges in its development.

When adopting a Problem Solving perspective, it was observed that it made the problem more practical, facilitating its resolution from the technical point of view, since by solving a specific problem with some predefined technique, the solution itself is already a significant contribution.

As the analysis performed by this work happened in the first moments of the research, after the definition of the research question, it allowed a greater detail of what would be studied, as well as the impact of this study not only from a technical point of view, but also social.

This allowed the research already to be directed to overcome the many limitations and problems raised by the use of the techniques, already predicting problems that would happen if the solution were developed considering only the traditional research model.

Among the contributions, we highlight the creation of the Stakeholder Diagram, with which it was possible to identify a large number of stakeholders in the system, which is a lot to be identified in the traditional approach. Using this diagram, the five levels of interaction were considered, so different types of stakeholders were identified, and it is possible that the same stakeholder can participate at different levels, allowing different contributions to be made by each one.

The use of the Evaluation Frame allowed to visualize the project from a more critical point of view, trying to anticipate potential problems that would be encountered by each stakeholder in the development of the system, as well as potential solutions for each problem.

In the use of the Value Pie, the 10 layers proposed by Hall were evaluated individually, allowing to observe the design of different optics and to easily identify large gaps in the scope of the project, which could generate several other Research Problems.

Another advantage of Value Pie was the possibility of analyzing the problem from the Informal, Formal and Technical points of view, verifying its impact in each of these dimensions.

Thus, at the end of the study the Research Problem was observed in a much more critical and judicial way, facilitating, even, the transformation of this into a product or patent. The experience showed the usefulness and viability of using sociotechnical artifacts to support a critical and more comprehensive understanding of the problem being handled as well as the possible solutions for it. If we are to make IoT solutions more close to Mark Weiser's ideas and expectations, we better start adopting a sociotechnical and systemic perspective for understaing IoT-related domains.

#### REFERENCES

- [1] Antti Oulasvirta and Kasper Hornbæk. HCI research as problem-solving. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI 16*. Association for Computing Machinery (ACM), 2016.
- [2] T. S. Weston and Larry Laudan. Progress and its problems: Towards a theory of scientific growth. *The Philosophical Review*, 87(4):614, oct 1978.

- [3] Kecheng Liu. *Semiotics in Information Systems Engineering*. Cambridge University Press (CUP), 2000.
- [4] M. Cecilia C. Baranauskas, Juliano Schimiguel, Carlos A. C. Simoni, and Cláudia Bauzer Medeiros. *Semiotics in Information Systems Engineering*. Proceedings of 11th International Conference on Human-Computer Interaction, 2000.
- [5] Roberto Pereira and Maria Cecília Calani Baranauskas. Value pie: A culturally informed conceptual scheme for understanding values in design. In *Lecture Notes in Computer Science*, pages 122–133. Springer Science Business Media, 2014.
- [6] E Hall. *The Silent Language*. NY: Anchor Book, 1959.
- [7] M. Weiser. Hot topics-ubiquitous computing. *Computer*, 26(10):71–72, 1993.
- [8] A. Arora V. Dhingra. Pervasive computing: Paradigm for new era computing. *IEEE*, pages 349 – 354, 2008.
- [9] Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun, and Hui-Ying Du. Research on the architecture of internet of things. In *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*. Institute of Electrical and Electronics Engineers (IEEE), aug 2010.
- [10] FAB. Comando da aeronáutica publica nova legislação sobre aeronaves remotamente pilotadas. In <http://www.fab.mil.br/noticias/mostra/23937/>. Acessado em 8 de dezembro de 2016., dez 2015.
- [11] Miguel Castro Neto, Pedro A. Pinto, and José Paulo P. Coelho. *Tecnologias de informação e comunicação e a agricultura*. Porto: Sociedade Portuguesa de Inovação, 2005.
- [12] Eobin Alex George, Gaurav Tiwari, R N Yadav, Edward Peters, and Srishti Sadana. UAV systems for parameter identification in agriculture. In *2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS)*. Institute of Electrical & Electronics Engineers (IEEE), aug 2013.
- [13] Alexandre S. Brandão, Felipe N. Martins, and Higor B. Soneguetti. A vision-based line following strategy for an autonomous uav. In *2015 IEEE 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. Institute of Electrical & Electronics Engineers (IEEE), jul 2015.
- [14] David Anthony, Sebastian Elbaum, Aaron Lorenz, and Carrick Dettweiler. On crop height estimation with UAVs. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Institute of Electrical & Electronics Engineers (IEEE), sep 2014.
- [15] UNICAMP. Dsc - social aware design. In <http://www.nied.unicamp.br/dsc>. Acessado em 8 de dezembro de 2016., dez 2016.

# A Systematic Mapping for the Scenario of Non-Urban Autonomous Vehicle Cooperation Systems

Robison Cris Brito  
Federal University  
of Technology - Paraná  
Campus Pato Branco  
robison@utfpr.edu.br

José Felipe Loureiro  
Federal University  
of Technology - Paraná  
Campus Pato Branco  
zefelippe\_0@hotmail.com

Eduardo Todt  
Federal University  
of Paraná  
UFPR  
todt@inf.ufpr.br

Roberto Pereira  
Federal University  
of Paraná  
UFPR  
rpereira@inf.ufpr.br

**Abstract**—In recent years, with the increase of postgraduate programs at masters and doctoral level, the number of scientific publications has increased in the same proportion. Today, on the Internet, there are many research bases for scientific articles that end up pulverizing important information for the development / maintenance of scientific research. One scenario that is receiving a lot of attention is the cooperation between autonomous vehicles. In this way, the present work proposes the use of a systematic mapping for the scenario of systems with cooperation of non-urban vehicles, presenting the steps for the accomplishment of that, the definition of research questions, and advances to the definition of search terms, selection of papers, categorization and retrieval of relevant information. For the present study, it was observed that approximately 10% of recovered articles deal with systems of vehicle cooperation, of which most publications occurred in the last year (2016), and the main research centers in this area are located in The United States, China and Italy, with isolated research from a few other countries. With this work, it was possible to validate a new research in the scenario of Vehicle Cooperation Systems, focused on non-urban vehicles, particularly using a group of UAVs applied to precision agriculture, as well as identifying the most promising technologies for this development. Finally, the main congresses / events to seek out / publish papers about this topic were presented at the end of this work.

**Keywords**—*Systematic Mapping, Framework, Colaboration, UAV, Multiagent System, Auction Method*

## I. INTRODUCTION

With the evolution of information technology, it's nowadays very easy to recover and / or share data using the world wide web. This easiness reflects the large amount of information produced, ranging from simple text to photos, videos, music, and other media.

According to a study conducted in 2016 by Northeastern University (USA), the amount of digital information available on the Internet in 2013 was 4.4 Zettabytes of information, with the outlook for 2020 being 44 Zettabytes. Every day new 2.5 Petabytes of information are included in the network, which is equivalent to, for example, 90 years of HD videos, or 530 million songs [1].

This evolution also reflects the amount of scientific documents available on the Internet. According to [2], Brazilian authors, published in 2015 approximately 47 thousand scientific articles, an increase of 16% compared to 2014. This

increase has also occurred worldwide, and the main reason is the increase in postgraduate programs, which implicates on the training of new masters and doctors.

Considering the great diversity of available scientific material, as well as the great number of researchers that exist in all the countries, the task of carrying out a bibliographical survey for a specific research becomes quite complex, given the quantity of publications, as well as the diversity of existing documents.

Thus, a deficient bibliographical survey can leave great gaps in the research horizon. Many approaches, as well as articles and materials, may simply go unnoticed.

In order to formalize the bibliographic survey process, as well as to identify possible gaps in the research horizon, techniques such as Systematic Mappings and Systematic Review, which are also known as primary and secondary studies, respectively, can be applied, avoiding the possible random searches for articles on search engines on the Internet.

Primary studies can be characterized as surveys that are guided by the hypotheses that are desired to be investigated or understood, and secondary studies are applied to integrate the results from several primary studies related to a specific subject [3].

Systematic reviews (SR) are very common types of secondary studies. Conventional SRs aggregate results on a specific issue applied to address relational and comparative research questions [3]. In SR, the research process is conducted following a well-defined sequence of steps, according to a previously planned study protocol [4].

Systematic mappings are a particular type of systematic review with a broader view of primary studies and are designed to give an overview of a research area by means of the classification and counting of contributions in relation to the categories of this classification [3] [5].

This type of work consists of searching the literature to know which topics were covered and where they were published. It is very useful in identifying evidence and in building knowledge. While systematic reviews aim to synthesize evidence, systematic mappings are primarily concerned with structuring a research area [5].

## II. SYSTEMATIC MAPPING

Systematic Mapping consists of performing a wide search, exploring a particular scenario more broadly before refining the search to a specific topic through an in-depth search. The steps for the construction of this mapping were defined according to [5].

The authors direct that a Systematic Mapping should be conducted following five steps, applied sequentially:

- Definition of Research Questions (DRQ);
- Search for preliminary studies of relevance;
- Selection of primary studies;
- Categorization of articles;
- Mapping and extraction of information that answers the previously established questions.

Throughout the text is presented in detail of how each of the five steps were developed in the present work.

### A. Definition of Research Questions

In order to map studies on non-urban vehicle co-operation that resulted in some model or framework, it is necessary to define one or more research questions (RQ) that have not yet been answered by other works, at least not using the methodology suggested by the author who proposes these questions.

The PICO methodology, suggested by [6], was used to formulate the Research Questions, as well as to create the search terms, which were submitted to the search engines of the databases of scientific publications. The acronym PICO stands for:

- **Population:** Refers to the population group that will be observed. In the context of this work, the articles that cover cooperation between autonomous vehicles in general, and, preferably, present a framework or model of generic cooperation for any type of vehicle that has emerged;
- **Intervention:** With the existing advantages in the main articles observed in the previous item, it identifies the possible relationships among these, highlighting the strengths and the problems that have not been solved yet. In this item the gaps for potential researches will be identified, as well as listing the most significant projects to fill these gaps;
- **Comparison:** It is considered what will be compared in the context of Systematic Mapping. With the mapping, it will be possible to find similar works and identify ways to compare the performance of what will be proposed by the author with what already exists;
- **Outcomes:** Related to the results achieved after the Systematic Mapping. In this context, it will be verified the best type of vehicle and scenario to explore in a Systematic Review, which technologies to implement and simulate the results, main congresses / magazines that publish papers about it, find the places on the world where this issue is most relevant, as well as

identify whether the amount of publications in this area are increasing or decreasing.

For the present study, four research questions (RQ) were defined that delimit the search space for scientific works:

- RQ01: What are the main studies involving multiple autonomous vehicles that collaborate with each other to achieve a specific goal?
- RQ02: What are the main studies that have resulted in models or frameworks for cooperation between autonomous vehicles;
- RQ03: How to abstract from these studies a generic model that can be applied not only to a specific scenario, such as, for example, unmanned aerial vehicles, but also to any non-urban autonomous vehicles (aerial, terrestrial or aquatic) that adapts to dynamic environments with frequent entrances and exits of vehicles, as well as recover from adverse events, such as the early termination of the battery / fuel of some of these vehicles, as well as natural events (gusting winds, rains, congestion, among others);
- RQ04: What is the best real scenario to test the architecture proposed by RQ03?

The RQ01 and RQ02 will be fully answered in the course of this Systematic Mapping. Already RQ03 is a more specific question, and its answer will probably be the result of a thesis, in this way, the present paper does not answer RQ03, but it gives subsidies so that it can be answered in a more deep work, this being a Systematic Review. RQ04 identifies the best scenario for the validation of RQ03, which is also answered by the present work.

The defined research questions have the main objective of guiding the mapping, which will later deepen into a Systematic Review.

These Research Questions meet the premises defined by the PICO methodology, and from them the preliminary study searches will be carried out.

### B. Search for Preliminary Relevance Studies

Based on the research questions, the keywords, along with their synonyms, are defined for the construction of search strings for the databases of existing publications on the Web. The following bases were chosen: IEEE Xplore (<http://ieeexplore.ieee.org>), ACM (<https://dl.acm.org/>), Scopus (<https://www.scopus.com>), Science Direct (<https://www.scopus.com>) and Compendex (<http://www.engineeringvillage.com/>). The electronic databases were selected based on the experience reported by [7], in addition to the conversation that the author of this work had with doctors and researchers, who mostly suggested these bases for research of scientific articles.

For the formulation of the search terms, three major research axes have been defined:

- Works that show the use of autonomous vehicles of any nature (aerial, terrestrial or aquatic);

TABLE I. KEYWORDS AND SYNONYMS FOR THE CONSTRUCTION OF THE SEARCH STRING.

Key-Word	Synonym
autonomous vehicle	unmanned aerial vehicle, UAV, autonomous aerial vehicle, self-governing aerial vehicle, uncontrolled aerial vehicle, unmanned drone, autonomous drone, self-governing drone, uncontrolled drone, unmanned car, autonomous car, self-governing car, uncontrolled car, unmanned vehicle, autonomous vehicle, self-governing vehicle, uncontrolled vehicle, unmanned boat, autonomous boat, self-governing boat, uncontrolled boat, unmanned submarine, autonomous submarine, self-governing submarine, uncontrolled submarine.
cooperation	association, teamwork, participation, alliance, combination, partnership, negotiation, collaboration
Framework	environment, system, simulation environment, multi agent system, MAS.

- Works that make use of some mechanism of cooperation between the vehicles, in order to reach a common goal;
- Works that have resulted in some model for cooperation between vehicles, this being a framework, a multiagent system, a simulation environment, or something similar.

It was observed that, in this way, search terms would be quite generic, and a large number of articles would be selected, which is the purpose of the present mapping, since, retrieving a large amount of articles increase the possibilities of finding other works that will contribute directly or indirectly to the thesis Research Questionnaire (RQ03).

For the definition of the keywords, terms were used in English, since the great majority of published works are in this language. Also defined were three large sets of keywords, which are believed to return the vast majority of works related to systems for cooperation between autonomous vehicles. They are: Autonomous Vehicle, Cooperation and Framework, as well as their synonyms.

For the synonyms the online environment [www.thesaurus.com](http://www.thesaurus.com) was used, which allows to consult synonyms for words in the English language.

From the identification of the keywords, these were presented in the table I, terms were concatenated using the logical OR connectivity when they belonged to the same set and the AND connective to different sets.

After selecting the keywords to compose search strings, they were submitted to the bases already mentioned (IEEE Xplore, ACM, Scopus, Science Direct and Compendex). Terms were fetched in the title, abstract, and keywords fields of the publications.

As a filter criteria, only articles published after the year 2011 (publications with a maximum of five years from publication) were selected.

It is believed that, because it is a recent issue and with many articles about it, articles prior to 2012 will be referenced or treated within the most recent publications, or will not be

so relevant to the present study, therefore is no need for its selection in Systematic Mapping.

This research was carried out in March 2017, and in the opportunity, 1,575 articles were retrieved, divided as follows: IEEE ([ieeexplore.ieee.org](http://ieeexplore.ieee.org)) - 194 articles, ACM ([dl.acm.org](http://dl.acm.org)) - 549 articles, Scopus ([www.scopus.com](http://www.scopus.com)) - 448 articles, Science Direct ([www.sciencedirect.com](http://www.sciencedirect.com)) - 61 articles, Compense ([www.engineeringvillage.com](http://www.engineeringvillage.com)) - 323 articles.

From these, the main studies were selected, also known as primary studies, according to the criteria explained in the following subsection.

### C. Selection of Primary Studies

For manipulation of the studies retrieved in this research, the Mendeley Desktop tool ([www.mendeley.com](http://www.mendeley.com)) was used. It is used for reference management, as well as an academic social network for researchers.

The results of the searches carried out in the five bases were exported to the BibTeX format and imported into Mendeley. All information, such as title, abstract, keyword, conference name, year of publication, and others, were imported. For some possible publications that had incomplete data, they were filled manually during the categorization process (next section).

After deletion of the duplicate articles, there remained **1076 papers** for analysis. With the unique articles, the work of refinement was begun, in which the articles are selected for the categorization.

Inclusion Criteria (IC) and Exclusion Criteria (EC) were used for the refinement process.

The inclusion criteria for selection of articles are:

- IC1: If several papers present similar studies, only the most recent is selected;
- IC2: If there are two versions of the same article, one summary and one complete, only the complete one will be selected;
- IC3: Articles presenting autonomous vehicles, or communication techniques, or mechanisms of cooperation or negotiation, will be selected.

The exclusion criteria has the function of eliminating studies without relevance to the subject to be mapped. Work that falls into at least one of the exclusion criteria is neglected in the mapping process. The following exclusion criteria were proposed and used:

- EC1: Works that are not related to Research Questions;
- EC2: Studies that are not in the English language;
- EC3: Articles published before the year 2011.

Applying the inclusion and exclusion criteria, which happened through the analysis of the title, abstract and keywords were selected for categorization **545 articles**.

During the categorization process (which is presented in detail in the Categorization of Articles section), a further 23 articles were deleted from the list, result at the end of **523 articles**.



#### D. Categorization of Articles

After selecting the articles, the next step is categorization, which happens through reading the title, abstract and keywords.

The Mendeley Desktop tool was of paramount importance in the categorization process and it was possible to create folders and subfolders within the environment in which articles were copied after reading the meaningful data.

In order to categorize, the articles were initially separated into four major groups: articles dealing with multiple vehicles (387 articles - 74 %), articles dealing with a single vehicle (52 articles 9.94 %), articles which contain basic concepts (16 articles, 3.06 %) and articles dealing with urban land vehicles (68 articles, 13.00 %).

The multiple vehicle category contains the most important articles for the Systematic Mapping, because it is in them that are the works that deal with the cooperation between vehicles.

The category with articles that deal with a single vehicle mostly have studies on very specific features, such as battery optimization, self-localization techniques using cameras, among other concepts, for this reason they have not been discarded, since they can be used for future queries.

The articles that have basic concepts were separated for a preliminary reading, because, although they do not have a very significant scientific value, they present tutorials and how-to-do about resources that could be used in the development of future works.

Finally, in a specific category are the works that approach the autonomous urban terrestrial vehicles, as well as their communication network (VANET - Vehicular ad hoc networks). These were separated into a specific category because, at first, they will not have much meaning in the proposal phase of a vehicle cooperation architecture, however, they may be used for consultations at more advanced stages of study.

For the 387 articles dealing with multiple vehicles, they underwent two new categorization processes, in order to observe the types of vehicles covered by these works, as well as the types of their applications.

Regarding the type of vehicle, the works used exclusively UAVs (262 articles - 67.70 %), UAVs cooperating with other vehicles (50 articles - 12.92 %), generic vehicles (28 articles - 7.24 %) , aquatic (23 articles - 5.94 %), agents in simulation environments (18 articles - 4.65 %), terrestrial vehicles (6 articles - 1.55 %).

The same 387 articles dealing with multiple vehicles were also categorized based on the type of application, the result was: vehicle cooperation (180 articles - 46.81 %), communication techniques (51 articles - 13.18%) (30 articles - 7.75 %), case studies (26 articles - 6.74 %), man-robot interface ( 23 articles - 5.94 %), self-location (19 articles - 4.91 %) and take-off / landing (11 articles - 2.84 %).

As the present work aims at identifying the main techniques of cooperation between vehicles, as well as proposing a Framework for this purpose, these two categories were re-categorized.

For the 180 articles dealing with cooperation between vehicles, the following groupings were identified: Target Tracking (36 articles - 20.00 %), Movement Coordination (35 articles - 19.44 %), Co-operation between heterogeneous vehicles (25 articles - 13.89%), Allocation of Tasks / Resources (16 articles - 8.89 %), **Methods of Cooperation (14 articles - 7.78 %)**, Search and Rescue (11 articles - 13.89 %), Area exploration (10 articles - 5.56 %), Negotiation methods (8 articles - 4.44 %), Area monitoring (5 articles - 2.78 %), Path Planning (5 articles - 2.78 %), Mission Planning (4 articles - 2.22 %), **Cooperation between Software Agents (4 articles - 2.22 %)**, Load Transport (4 articles - 2.22 %) and Team formation (3 articles - 1.67 %). The highlighted articles were selected for the reading in integral, for the identification of the main methodologies used for the cooperation).

In relation to the 47 articles dealing with Frameworks, they were also categorized, with the following groupings being identified: **Cooperation (28 articles- 59.57 %)**, Simulator (8 articles - 17.02 %), Communication (7 articles - 14.89 %), Cloud (3 articles - 6.38 %) and Design / Project (1 articles - 2.13 %). The highlighted articles were also separated for an integral reading, since they deal with Cooperation Frameworks, this being the focus of this Systematic Mapping.

#### E. Mapping and Extracting Information that Responds to Previously Established Search Issues

Regarding the RQ01 research question, of the 523 articles categorized, 180 dealt with some type of cooperation between autonomous vehicles, which shows that this subject is being explored and has a lot of potential. Of these, most deal with cooperation in very specific scenarios, such as exploration, search and rescue, coordination of movements, allocation of tasks / resources, among others.

Considering the vehicle cooperation model in a more general way, we selected 14 articles dealing with cooperation methods used by autonomous vehicles, as well as the 4 articles dealing with the cooperation between software agents in a simulation system , Since the algorithms proposed in the articles could be used for a model of cooperation between vehicles.

Regarding RQ02, 28 articles were selected for full reading for they integrate the group of articles that deal with Frameworks that aim at the cooperation of autonomous vehicles.

In order to further explore the research topic, as well as to accurately answer RQ03, it is necessary to carry out a Systematic Review, where the total of articles presented in the previous paragraph (48 articles) should be read and analyzed. With this reading it is possible to identify the technologies used, the methodology of work and the analysis of the results, in order to propose an unpublished research topic for the doctoral thesis. By reading these articles, it is also possible to identify similar works and propose methodologies for comparing what already exists with what will be proposed.

For RQ04, it was identified that scenarios involving UAVs are more exploited for tests of cooperation between Drones, since 262 articles dealt with the use of UAVs, and 50 articles performed the cooperation between UAVs and other vehicles, so in total there were 312 works that explored this scenario, corresponding to 80.62 % of the articles categorized.

### III. RESULTS

In the previous section, the steps to arrive at the Systematic Mapping of a scenario involving cooperation between non-urban vehicles were presented. In summary, some quantitative results were obtained, such as:

- Number of articles that use multiple vehicles, single vehicles, present basic concepts and deal with urban land vehicles;
- Number of articles grouped by type of vehicle;
- Number of articles grouped by type of application;
- Number of articles grouped by type of cooperation;
- Number of articles grouped by type of Frameworks.

After categorizing the articles using the Mendeley Desktop tool, the next step was to extract qualitative information based on these groupings. Mendeley allows the information to be consulted visually, but for a computational analysis of it, the database of this tool, which is in SQLite (<https://www.sqlite.org>) was manipulated by a Java SE program developed by the author this work.

For the qualitative analysis of the Mendeley data, the 46 selected articles were used for the reading in the integral. The first analysis was performed regarding the keywords of the article, some keywords were neglected from the analysis, such as Unmanned aerial vehicles, since they did not present significant contributions.

The words that most appeared in the key words of the papers, in order, were: Agent, Human Interaction, Algorithms and Heuristics, Auction Method, Bayes Methods, Scheduling, Optimization, Service-Oriented Architecture, Control Theory, Game Theory, Ontology, Genetic Algorithm, Linear Program, Markov decision evolutionary game (MDEG), Matematical Model Multiobjective and Optimization.

For a more specific analysis using the works summary, all abstracts were concatenated in a single text and generated a TagCrowd from the Wordart tool (<https://wordart.com/create>). TagCrowd displays the words in a text that appear more frequently. In addition to the highlighted words already identified in the key words, such as TagCrowd we observed the frequent use of words like Control Mission, Task, Coordinate, Plan, Research and Textile.

The next analysis performed was in relation to the country where the 46 selected articles were produced. To identify the country, a manual process was used, in which the articles were retrieved from the Internet and verified the country / university of the researcher who wrote it. In situations where the authors of different countries wrote the same article, both countries were considered in the count, since the objective is to identify the poles of research in the area.

The countries that produce the most articles on the subject are: United States (13 articles), China (8 articles), Italy (8 articles), France (4 articles), United Kingdom, Sweden and Germany (3 articles), Japan (2 articles), Brazil, Uruguay, Holland, Switzerland, Poland, Norway, Greece, Czech Republic and Australia (1 article each). For a cluster analysis, identifying the geographical location of most publications, the MapChart

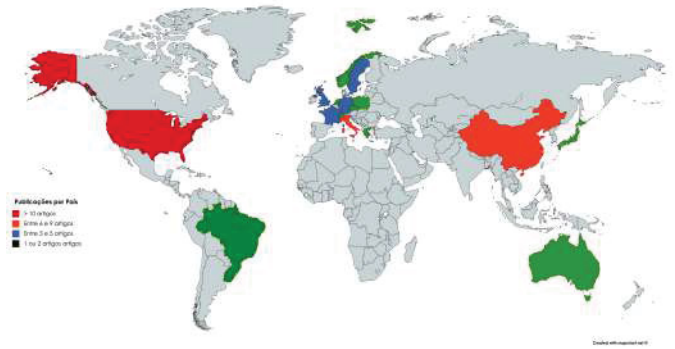


Fig. 1. Map of publications by geographic region.

portal (<http://mapchart.net>) was used. The result is shown in Figure 1.

Regarding the period in which the publications took place, a grouping of the data was carried out and it was verified that over the years the number of publications in the area of systems of cooperation between vehicles increased considerably, starting with two publications in 2011, between 7 and 9 publications per year between 2012 and 2015 and significantly increasing the number of publications in 2016, where there were 13 articles published.

With respect to the tools and software used for testing, simulation or organization of autonomous vehicle cooperation systems, the following tools were identified from the title, summary and keywords analysis: Paparazzi UAV (<http://wiki.paparazziuav.org>), JAUS (<http://openjaus.com>), AETOURNOS (<http://aetornos.gforge.inria.fr>), TAEMS (<http://mas.cs.umass.edu/researchhold/taems/>), Mission Planner (<http://ardupilot.org/planner>), Simulink for Matlab (<https://www.mathworks.com/products/simulink.html>), UAVSim (<https://www.openhub.net/p/uavsim>), The Network Simulator (<http://www.isi.edu/nsnam/ns>) e OMNeT++/Castalia (<http://cpham.perso.univ-pau.fr/WSN-MODEL>).

The identification of the tools is very important to use the same environment used by other articles, in order to perform comparison of results. The study of the tools has not gone into depth. Based on the name, an Internet search was carried out, the project page identified and the characteristics of the tools checked.

Based on the articles, it was also identified which congresses have the largest number of publications in this area. The following stand out:

ICAS - Congress of the International Council of the Aeronautical Sciences (<http://www.icas.org>), DASC - Digital Avionics Systems Conference (<http://2017.dasconline.org>), AUVSI - Unmanned Systems North America Conference (<http://www.auvsi.org>), Handbook of Unmanned Aerial Vehicles (<http://www.springer.com/us/book/9789048197064>) and International Journal of Intelligent Unmanned Systems (<http://www.emeraldinsight.com/Toc/ijius/4/3>).

In addition to the publications of specific events that have been cited previously, a number of articles in the area have been published in other events. Some of these are presented:

AAMAS - International Conference on Autonomous Agents and Multiagent Systems (<http://www.aamas2017.org>), SEEA - Euromicro Conference on Software Engineering and Advanced Applications (<http://dsd-seea2017.ocg.at>), PE-WASUN Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks (<http://pe-wasun.gforge.inria.fr>), among others.

With the analysis of the presented data, it is observed that few congresses in the area of autonomous vehicles possess Qualis, this is due to the short time of life of this specific area. However, other well-qualified congresses (with limited qualifications) tend to accept publications, many of which are presented in the previous paragraph.

#### IV. CONCLUSION

The present work presented the Systematic Mapping applied to the scenario of cooperation systems of non-urban autonomous vehicles, in which more than a thousand articles were retrieved initially, classified, and categorized. Further, based on this mapping, quantitative and qualitative data were collected so that will support a future Systematic Revision in this area, aiming at the definition of a consistent theme for a doctoral thesis.

At first, the work conceptualized Systematic Review, as well as presented the steps necessary for it to be performed. As the success of a systematic mapping depends on the quality of the Research Questions formulated, the PICO (Population, Intervention, Comparison and Outcomes) methodology suggested by [6] was used to formulate these.

Another positive feature of this project was the use of the Mendeley Desktop tool for the organization of articles, as well as their categorization. This tool allows import and export of data in the BibTeX format, which greatly facilitated the manipulation of articles.

With the present work a quantitative survey of the articles was possible, identifying the main areas of research and the impact in relation to the number of published works on these areas, as well as it was possible to identify that the use of UAV, for example, has a very strong tendency for the scenario of cooperation between vehicles.

Other qualitative data were also generated with the development of this work, it was possible to identify the main technologies and tools used for the development of a system of cooperation between vehicles, where the main researchers of these areas are, as well as the main congresses.

As future work, it highlights the need to use the data generated by this Systematic Mapping for the study and development of a Systematic Review, in which articles should be read in their entirety, aiming to take advantage of studies already done to define a unified model of cooperation.

The results of this work shows that the Systematic Review is a valuable tool to support the understanding and review of publications related to the state of the art in a topic, providing solid ground to a research being developed.

#### REFERENCES

- [1] Mikal Khoso. Chow much data is produced every day? In <http://www.northeastern.edu/levelblog/2016/05/13/how-much-data-produced-every-day/>. Acessado em 16 de fevereiro de 2016., Mai 2016.
- [2] Osvaldo N. Oliveira. Research landscape in brazil: Challenges and opportunities. *The Journal of Physical Chemistry C*, 120(10):5273–5276, mar 2016.
- [3] Cleiton V. C. Magalhães, Ronnie E.S. Santos, Fabio Q. B. Da Silva, and Alex Sandro Gomes. Caracterizando a pesquisa em informática na educação no brasil: Um mapeamento sistemático das publicações do SBIE. *Comissao Especial de Informatica na Educacao*, nov 2013.
- [4] Elizabete Munzlinger, Fabricio Batista Narcizo, and José Eustáquio Rangel de Queiroz. SistematizaÇÃo de revisÖes bibliogrÁficas em pesquisas da Área de ihc. In *Companion Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems, IHC '12*, pages 51–54, Porto Alegre, Brazil, Brazil, 2012. Brazilian Computer Society.
- [5] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, aug 2015.
- [6] Barbara Kitchenham and Stuart Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.
- [7] Tore Dyba, Torgeir Dingsoyr, and Geir K. Hanssen. Applying systematic reviews to diverse study types: An experience report. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*. Institute of Electrical and Electronics Engineers (IEEE), sep 2007.

# Comparison Analysis Between PID and Fuzzy Logic Controllers for Quadrotor in a Simulated and in a Real Environment

Fábio Favarim<sup>1</sup>, Robison Cris Brito<sup>1</sup>, Kathya S. C. Linares<sup>1</sup>, Eduardo Todt<sup>2</sup>

**Abstract**—Unmanned aerial vehicles (UAVs) like quadrotors are becoming more popular and are being used in many different applications. While there are many ready-made models on the market, their cost is often high due to industrial value, patents, among other costs. Today it is possible to assemble low cost quadricopters, however, the control of this type of vehicle is not trivial due to the control of several variables that involve the dynamic model of the system. In this work two strategies of attitude control are compared: the classic PID control and the fuzzy control. Both controls were modeled, implemented, simulated in Matlab / Simulink and also applied to a real quadrotor. The results of the execution of the controllers in both simulation and real quadrotor showed that the fuzzy control is robust, especially in real environment, confirming the data obtained in a simulated environment.

## I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have been used in various areas of knowledge. The range of applications for these types of vehicles is wide and can be used for mapping areas (cartography), maintenance of infrastructures such as transmission lines, public safety (border and urban area supervision), intervention in hostile environments and precision agriculture [1].

There are several types of UAVs, the main ones being the fixed-wing, the articulated wings and the Vertical Take-off and Landing (VTOL) vehicles, such as helicopters. In the latter, there is still a subset that needs to be highlighted, which is the object of study of this work, which are the multirotors. These consist of vehicles with multiple propeller-coupled rotors, which are arranged symmetrically in relation to the geometric center of the aircraft. In this work was used a multirotor composed of four rotors, also known as quadrotor or quadricopter. In this case, the rotors are arranged in a cross structure.

There are some advantages of using multirotors compared to other types of UAVs. Among them are the ability to remain in a static position in the air, to fly in any direction, to take off and land vertically and to fly at low speeds. Such behavior resembles conventional helicopters, but with greater stability due to the number of rotors [2]. A disadvantage is flight autonomy when compared to fixed wing UAVs due to their

low aerodynamics, thus requiring a greater amount of energy to maintain in the air. This ability to fly in all directions makes the control of this type of vehicle more complex because it is a system with many variables and a complex and strongly coupled dynamics. However, due to the growing application of these vehicles has motivated several researches in the academic area.

Once in the air, a multirotor uses fixed-pitch propeller blades to control the vehicle motion by varying the relative speed of each rotor to change the thrust and torque produced, allowing a unique range of movement. Attitude control consists of a combination of controls that allows the quadrotor to execute pitch, yaw, roll movements while in flight[3].

Classical control techniques such as Proportional (P), Proportional Derivative (PD), Proportional Integral Derivative (PID) used in the industry is widely used for this control. However, due to the approximation model of these techniques, which require adjustments of gains, in addition to external disturbances, such as wind, the quadrotors lose their effective control. Techniques based on intelligent systems are alternatives to perform this control. Among these techniques is fuzzy logic. Some works available in the literature have explored the application of fuzzy logic in this type of control [4] [5] [6].

This work aims to develop an attitude control for a quadricopter applying fuzzy logic in the control. In order to compare its behavior with classical control techniques, a PID control was also developed. The differential of this work in relation to the others is that in this we compare the performance of the PID control with the Fuzzy, taking into account the same quadrotor as a basis of comparison. In [7] we present the result of this comparison only in a simulated environment in Matlab with the help of its Simulink tool. In the present work, besides the simulation result we present the result of the control applied in a prototype of a quadrotor.

This work is organized as follows: Section II presents the modeling of the dynamics of the quadricopter; the design of the controls used in the simulation is presented in Section III; Section IV presents the results obtained in simulation environments, as well as in real environment. The conclusions are presented in Section V.

## II. QUADROTOR MODELING

The quadrotor is modeled with a four rotors in a cross configuration in rigid structure. Each propeller is connected to the motor through the reduction gears. The quadrotor

<sup>1</sup>Fábio Favarim, Robison C. Brito and Kathya S. C. Linares are with the Informatics Department, Federal University of Technology - Paraná, Pato Branco – PR – Brazil (e-mail: [favarim@utfpr.edu.br](mailto:favarim@utfpr.edu.br), [robison@utfpr.edu.br](mailto:robison@utfpr.edu.br), [kathya@utfpr.edu.br](mailto:kathya@utfpr.edu.br)).

<sup>2</sup>Eduardo Todt is with Informatics Department, Federal University of Paraná, Curitiba – PR – Brazil (e-mail: [todt@inf.ufpr.br](mailto:todt@inf.ufpr.br)).

motion is controlled by varying the speed of the propellers and not by using any mechanical actuators (tail rotor). Figure 1 shows the structure layout a quadrotor. The front and the rear propellers rotate counter-clockwise, while the left and the right ones turn clockwise. This configuration of opposite pair directions removes the need for a tail rotor. Even though the quadrotor has 6 DOF (degrees of freedom), it is equipped just with four propellers, hence it is not possible to reach a desired set-point for all the DOF, but at maximum four. These four basic movements allows the quadrotor to reach a certain height and attitude (roll( $\phi$ ), pitch ( $\theta$ ) and yaw ( $\psi$ )).

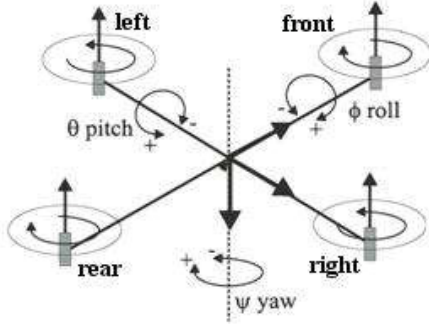


Figure 1. The structure layout of a quadrotor

In Figure 1 all the propellers rotate at the same (hovering) speed  $\Omega_H$  to counterbalance the acceleration due to gravity. Table 1 resumes speed of all propellers to get the desired movement.  $\Delta$  is a positive or negative variable which represents an increment or decrement respect of the constant  $\Omega_H$ .  $\Delta$  can't be too large because the model would eventually be influenced by strong non linearities or saturations. As an example, to get a roll movement it is necessary to increase (or decrease) the left propeller speed and decrease (or increase) the right one.

When simulating a system it is needed to know the system dynamics that can be represented by a mathematical model. In this work we need a dynamic model of the quadrotor structure and a model for and motors.

TABLE 1. SPEED OF PROPELLER AND MOVEMENTS

Movement	Speed of Each Propeller			
	Front ( $\Omega_1$ )	Right ( $\Omega_2$ )	Rear ( $\Omega_3$ )	Left ( $\Omega_4$ )
Hovering	$\Omega_H$	$\Omega_H$	$\Omega_H$	$\Omega_H$
height (Z)	$\Omega_H + \Delta$	$\Omega_H + \Delta$	$\Omega_H + \Delta$	$\Omega_H + \Delta$
roll( $\phi$ ),	$\Omega_H$	$\Omega_H - \Delta$	$\Omega_H$	$\Omega_H + \Delta$
pitch ( $\theta$ )	$\Omega_H - \Delta$	$\Omega_H$	$\Omega_H + \Delta$	$\Omega_H$
yaw ( $\psi$ )	$\Omega_H - \Delta$	$\Omega_H + \Delta$	$\Omega_H - \Delta$	$\Omega_H + \Delta$

#### A. Quadrotor Model Structure

Several studies were performed in order to propose a dynamic model of the quadrotor structure, based mainly on calculating the equations of motion for pitch, roll and yaw [1], [8], [9], [10], [11], [12]. In this work we have used the model described in [10] as show in Equation (1):

$$\begin{cases} \ddot{\phi} = \dot{\theta} \cdot \dot{\psi} \frac{I_y - I_z}{I_x} - \frac{J_r}{I_x} \cdot \dot{\theta} \cdot \Omega + \frac{1}{I_x} \cdot u_2 \\ \ddot{\theta} = \dot{\phi} \cdot \dot{\psi} \frac{I_z - I_x}{I_y} + \frac{J_r}{I_y} \cdot \dot{\phi} \cdot \Omega + \frac{l}{I_y} \cdot u_3 \\ \ddot{\psi} = \dot{\phi} \cdot \dot{\theta} \frac{I_x - I_y}{I_z} + \frac{1}{I_z} \cdot u_4 \\ \ddot{x} = (\cos\phi \cdot \text{sen}\theta \cdot \cos\psi + \text{sen}\phi \cdot \text{sen}\psi) \cdot \frac{1}{m} \cdot u_1 \\ \ddot{y} = (\cos\phi \cdot \text{sen}\theta \cdot \cos\psi - \text{sen}\phi \cdot \text{sen}\psi) \cdot \frac{1}{m} \cdot u_1 \\ \ddot{z} = -g + (\cos\phi \cdot \cos\theta) \frac{1}{m} \cdot u_1 \end{cases} \quad (1)$$

Where  $m$  [kg] is the mass of the quadrotor;  $J_r$  is the rotor moment of inertia;  $g$  is the gravity;  $I_x$ ,  $I_y$ ,  $I_z$  are moment of inertia about  $x$ ,  $y$  and  $z$ , respectively. The propellers' speed inputs are given through Equation (2) [10]:

$$\begin{cases} u_1 = b \cdot (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ u_2 = lb \cdot (\Omega_4^2 - \Omega_2^2) \\ u_3 = lb \cdot (\Omega_3^2 - \Omega_1^2) \\ u_4 = d \cdot (\Omega_2^2 + \Omega_4^2 - \Omega_3^2 - \Omega_1^2) \\ \Omega = \Omega_1 + \Omega_2 - \Omega_3 - \Omega_4 \end{cases} \quad (2)$$

Where  $l$  is the distance between the center of the quadrotor and the center of a propeller;  $b$  is the thrust coefficient of rotor;  $u_1$ ,  $u_2$ ,  $u_3$  and  $u_4$  are the components responsible for height, roll, pitch and yaw movement, respectively; and  $d$  is the drag coefficient.

#### B. Motors Model

We have used brushless electric motors were used in the prototype. This type of motor requires an electronic module control, called ESC (Electronic Speed Controller). The rotational speed and electrical power given to the motor is controlled by varying the duty cycle of a PWM (Pulse-Width Modulation) signal at the ESC input. The mathematical model of the motor takes as input a value of the PWM and returns the rotation speed of the propeller. In order to obtain this model we have used the ARX (Auto-Regressive with external input) estimation method [13]. The ARX estimator consists of a system that uses the input and output data of the system to obtain its mathematical model in the form of a transfer function. The mathematical model of each motor was obtained by experimentation, from a quadrotor developed at the University's Hardware / Software Laboratory, using a code developed in the Tiva C microcontroller for angular velocity sampling in motors. Figure 2 shows the quadrotor developed for the experiments.



Figure 2. Quadrotor developed for experimentation

Thus we have conducted experiments with several PWM values for each motor and measured the its rotational speed. These data were used as parameters to the ARX function available in Matlab. The transfer function identified by the ARX method of each motor are shown in Equations (3), (4), (5) and (6), with a precision of 91.28%, 88.03%, 90.37% and 89.44% for the motor/ESC/ propeller sets one, two, three and four respectively. The graph in Figure 3 shows experimental data acquired with the identified model of the motor/ESC/propeller set one.

$$\frac{\Omega_1}{PWM} = \frac{-0.03156z^{-4} + 87.37z^{-5}}{1 - 0.947z^{-1} - 0.03668z^{-2}} \quad (3)$$

$$\frac{\Omega_2}{PWM} = \frac{-1.786z^{-1} + 54.1z^{-2}}{1 - 0.9508z^{-1} - 1.273e - 5z^{-2} - 0.03065z^{-3}} \quad (4)$$

$$\frac{\Omega_3}{PWM} = \frac{0.8203z^{-4} + 75.88z^{-5}}{1 - 0.9508z^{-1} - 0.03421z^{-5}} \quad (5)$$

$$\frac{\Omega_3}{PWM} = \frac{0.8203z^{-4} + 75.88z^{-5}}{1 - 0.9508z^{-1} - 0.03421z^{-5}} \quad (6)$$

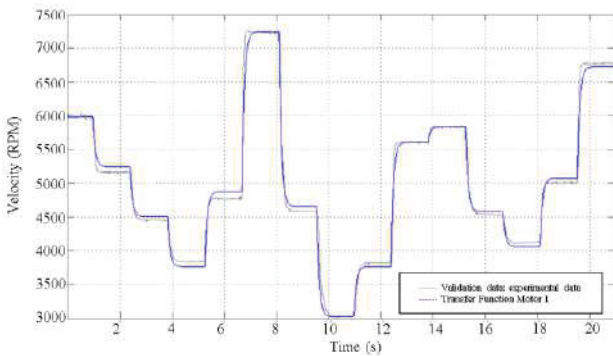


Figure 3. Comparison between the experimental data and identified model of a motor.

### III. SIMULATION

The simulation was implemented using Matlab Simulink [14]. The simulation was divided into four blocks: control; mixer; acting forces; and plant. Figure 4 shows the block diagram for the PID controller and in Figure 5 for the fuzzy controller. The following describes briefly each block.

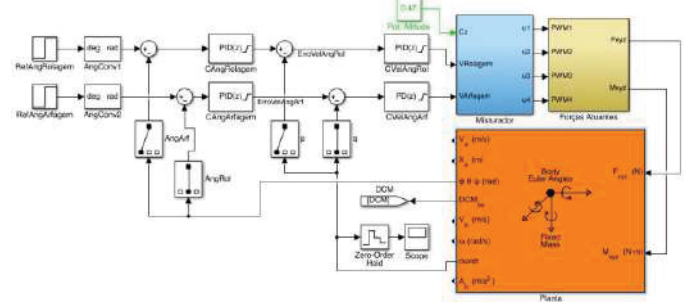


Figure 4. Block Diagram for Simulation of PID Controller

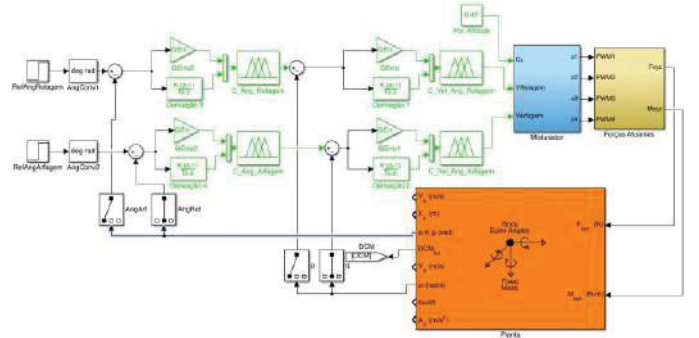


Figure 5. Block diagram for Simulation of Fuzzy Controller

#### A. Control block

The control blocks implement the control of the system. The main goal of the control is to move quadrotor from its position to a new one desired position (called reference). The control also needs to react to external disturbances. Attitude control is responsible to maintain stability during flight. In this study we proposed a fuzzy controller and compared it with the PID controller. PID control occurs by Equation (7) for continuous time, where:  $K_p$  is the proportional gain to increase or decrease the energy at the output of the controller;  $K_d$  the derivative gain that acts in abrupt variations; and  $K_i$  integral gain that decreases the final error. The PID gains were found using tools provided by Simulink tools. The final gains used were  $K_p = 1.17$ ,  $K_i = 0.0103$  and  $K_d = 0.661$ .

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int e(t) dt \quad (7)$$

Fuzzy Logic Control (FLC) emulates human-like reasoning and incorporates it into a predefined mathematical model to automate the complex control process [15]. The FLC for this work has two inputs namely error (E) and error rate (R) and one output that is the  $\Delta$  PWM (P) value. The linguistic variables used for error are negative/positive 1/2/3 and zero, for error rate negative/positive 1/2/3 and zero, for output negative/positive 1/2/3 and zero, as show in Table 2.

TABLE 2. LINGUISTIC VARIABLES

	Linguist Variables	Abbreviation
Error (E)	Negative: 1 – 2 – 3	EN1 – EN2 – EN3
	Zero	EZero
	Positive: 1 – 2 – 3	EP1 – EP2 – EP3
Error Rate (R)	Negative: 1 – 2 – 3	RN1 – RN2 – RN3
	Zero	RZero
	Positive: 1 – 2 – 3	RP1 – RP2 – RP3
PWM (P)	Negative: 1 – 2 – 3	PN1 – PN2 – PN3
	Zero	ZERO
	Positive: 1 – 2 – 3	P1 – P2 – P3

Rules are developed as if-then statements based on experience of the authors. For example, IF Error is “EP3” and Error Rate “RP3” THEN PWM is “PN3”. The interpretation of this rule is as follows: if actual altitude is much higher than desired, thus error is positive 3, i.e. velocity in Z direction is positive fast meaning a high upward velocity, then the output is negative 3, i.e. a quick decrease in PWM value for each motor thereby causing quick deceleration. A total of 42 rules have been developed for each FLC. The rule base is presented in the Table 3 below using the linguistic variables previously defined. MATLAB’s Fuzzy Logic toolbox was used to develop the fuzzy controllers.

TABLE 3. FUZZY BASE RULES

E\R	RN3	RN2	RN1	RZero	RP1	RP2	RP3
EN3	PP3	PP3	PP2	PP2	PP2	PP2	PP2
EN2	PP3	PP2	PP2	PP2	PP2	PP2	PP1
EN1	PP2	PP2	PP1	PP1	PP1	PP1	PZero
EZero	PP2	PP1	PP1	PZero	PP1	PP1	PN2
EP1	PZero	PN1	TN1	PZero	PN1	PN2	PN2
EP2	PN1	PN1	TN2	PN2	PN2	PN2	PN3
EP3	PN2	PN2	TN2	PN2	PN2	PN3	PN3

#### A. Mixer Block

In the mixing block, the control signals are associated for each angle in order to generate a PWM to control the four motors. This block is based on Equation 2.

#### B. Forces Block

In the block of forces, the response produced by the PWM in each motor is generated, calculating U1 and the rotations U2, U3 and U4 (Equation 2).

#### C. Plant Block

In order to simulate the quadrorotor model structure We have use the Simulink Aerospace Blockset in order to simulate the simulate the quadrorotor model structure (Equation 1).

## IV. RESULTS

This section is divided into Results in a Simulation Environment and Results in a Real Environment.

#### A. Results in a Simulation Environment

In the simulations we have introduced a disturbance in the rotational force on each axis. A Gaussian noise generator at a rate of 2Hz and with a variance of  $7e-3$  was used to generate the disturbance. An angle of 10 degrees is used as a reference in the roll and pitch angle. The results of the simulation are plotted in the graphs of Figures 6 and 7. Figure 6 shows the behavior of both controllers related to angular speed, while Figure 7 shows the controls related angular position. PID Control is represented by a green line and Fuzzy Control by a blue line.

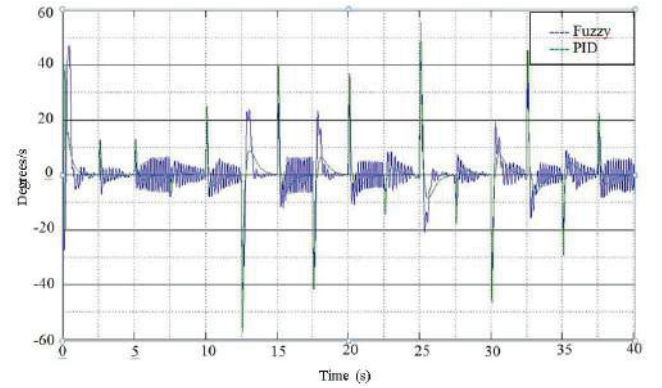


Figure 6. Comparison between fuzzy control (blue line) and PID (green line) at angular velocity

Figure 7.

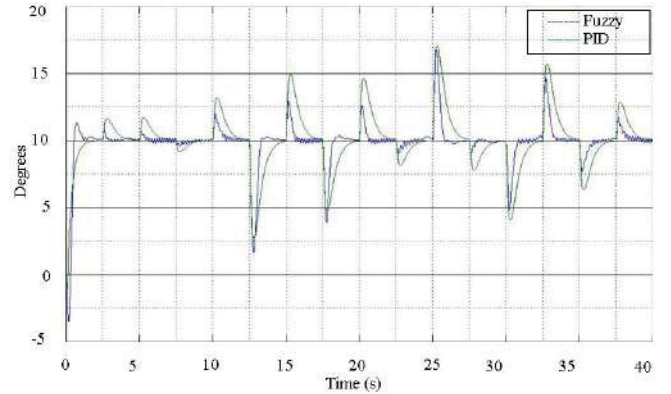


Figure 8. Comparison between fuzzy control (blue line) and PID (green line) at angle.

As can be observed in Figure 5 the PID has a more consistent response when controlling the angular velocity, that is, with less oscillation. However, when controlling the angle the fuzzy reacts faster than the PID, also decreasing the level of the maximum error reached with the disturbances, although it has small oscillations when it reaches the reference. The PID controller stabilized the errors in approximately one second, while the fuzzy stabilized in 0.6 seconds. Thus, in a simulated environment, the fuzzy behaved better, performing better than the PID.

## B. Results in a Real Environment

After the simulations with PID and Fuzzy controllers, these controls were implemented in the real quadricopter prototype and the experiments were carried out in a controlled environment (quadricopter structure trapped in an indoor test frame - Figure 2) and external flight (with a quadricopter fully assembled for flights - Figure 8).



Figure 9. External Flight of Quadrotor

The materials used to assemble the quadrotor was:

- 4 outrunner BLDC Turnigy L2215J three-phase motors;
- 4 ESCs Turnigy Plus 25 A;
- Propellers 10x4.5 weighing 11g;
- MPU-6050 sensor (gyroscope and accelerometer);
- Digital compass HMC5883;
- Barometer MS5611;
- Tiva TM4C123G development kit with ARM Cortex M4F Microcontroller;
- Zippy Compact LiPo 3S 25C 3700mAh battery.

All components were mounted on a carbon fiber structure, model Talon, from Turnigy manufacturer's. The developed quadricopter has a total weight of 1.2 kg. To obtain the angular velocity from the motors, we have use the MPU-6050 sensor, this parameter was used as an input to the controller.

The evaluation procedures in the real environment were performed using the data obtained with the inertial sensors (MPU-6050, HMC5883; MS5611) and the data were stored in the microcontroller and later plotted as shown in Figures 9 and 10. The blue line blue shows the reference and the green line the result of the controller.

In the real environment, the PID and Fuzzy presented differences compared to the simulated response. Due to the vibration generated by the motor rotation, the sensors obtained significant noise in the samplings. The noises can be softened with the implementation of filters. The noise also affected the controllers, especially the PID controller, generating some unexpected oscillations.

The results obtained from the PID controller in real environment are shown in Figure 9, and it can be observed that the controller did not present much stability and an error in relation to the reference. It was implemented exactly like in the simulation environment, without extra adjustments.

The Fuzzy controller obtained a response close to the simulation, as shown in Figure 10. Unlike the PID, the noise did not interfere significantly with the control.

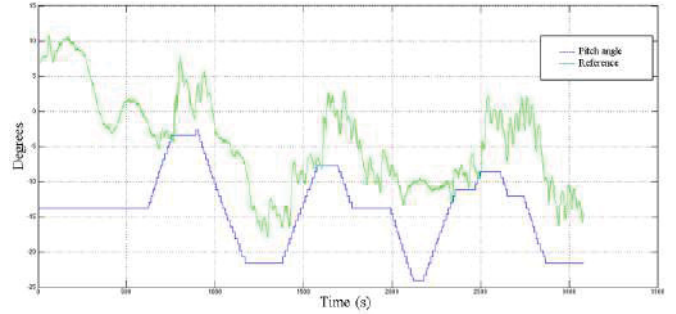


Figure 10. PID Control in a real environment

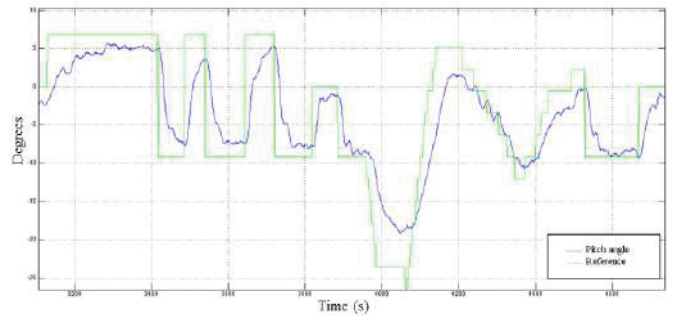


Figure 11. Fuzzy Control in a real environment

## V. CONCLUSIONS

This paper has presented a comparative analysis between a classical PID and Fuzzy Logic Controller for Quadrotor, both in a simulated and also in a real environment. In the PID controller many adjustments were necessary to the gains that resulted in a good stability of the system, whereas with the fuzzy controller the adjustments in the simulations were smaller.

The main differences when implementing the two techniques were the ease of implementation of the fuzzy controller, since it does not require a model as accurate as in the PID and the characteristics that are closest to the human-like reasoning.

The study carried out through simulations and in real environment showed that both techniques manage to control the action of the quadricopter, being that the technique using fuzzy logic presents better results in relation to the PID, due to its faster response and stability in the face of external interferences and nonlinearities. Future work intends to implement a hybrid PID controller with Neural Networks, as well as verify the feasibility of applications of other Artificial Intelligence techniques. For better precision, the implementation of a Kalman filter for inertial sensors would help prevent oscillations generated by the motor rotation.

## ACKNOWLEDGMENT

The authors thank UTFPR (Federal University of Technology – Paraná) and DAINF (Department of Informatics at UTFPR) for providing infrastructure and financial support provided.



## REFERENCES

- [1] Castillo, P., Lozano, R. and Dzul, A. (2005). *Modelling and Control of Mini-Flying Machines*, Springer, USA.
- [2] Vieira, J. C. S. (2011). *Plataforma móvel aérea quadrotor*. Dissertação (Mestrado) — Universidade do Minho - Ciclo de Estudos Integrados Conducentes ao Grau de Mestre em Engenharia Eletrónica Industrial e Computadores, <http://intranet.dei.uminho.pt/gdmi/site/arquivo/detalhearquivo.php?id=269>.
- [3] Farrel, J. A. (2008). *Aided Navigation GPS with High Rate Sensors*. McGraw-Hill Professional.
- [4] M. Santos, V. López, F. Morata, "Intelligent Fuzzy Controller of a Quadrotor" em IEEE International. Conference on Intelligent Systems and Knowledge Engineering, Hangzhou, 2010.
- [5] S. Zangenehpour, B. Dehghan, B. Asaei, "Controlling a One Degree of Freedom Arm, Using Fuzzy Controller", em International Conference on Information, Networking and Automation (ICINA), 2010.
- [6] D.C. Sales, A. Frutuoso, A.L. Santos, "Desenvolvimento de um controlador fuzzy para quadrotores", em IV Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBES), 2014.
- [7] C. D. Mattiello, B. T. Borsoi, K. C. Linares, F. Favarim, "Controle de Atitude para Veículos Aéreos Não Tripulados do Tipo Quadricóptero: PID vs Lógica Fuzzy" em Anais do VI Computer on The Beach, 2015.
- [8] Paula, J. C. de. (2012), *Desenvolvimento de um vant do tipo quadrirotor para obtenção de imagens aéreas em alta definição*. Curitiba, PR: Disponível em: <[http://dspace.c3sl.ufpr.br/dspace/bitstream/handle/1884/29886/R - D - JULIO CESAR DE PAULA-.pdf?sequence=1](http://dspace.c3sl.ufpr.br/dspace/bitstream/handle/1884/29886/R_D_JULIO_CESAR_DE_PAULA-.pdf?sequence=1)>.
- [9] Bresciani, T. (2008) "Modelling, Identification and Control of a Quadrotor Helicopter", Lund Sweden: 2008, <http://www.control.lth.se/documents/2008/5823.pdf>, julho.
- [10] Kon, J., Yamashita, Y., Tanaka, T., Tashiro, A. and Daiguji, M. (2013). Practical application of model identification based on {ARX} models with transfer functions. In *Control Engineering Practice*, v. 21, n. 2, p. 195-203.
- [11] D. A. do Nascimento, C. Z. Rufino, S. C. Oliveira and M. A. F. Feitosa, "Modeling and Design of an Autonomous Unmanned Aerial Vehicle Quadrotor Prototype for Applications in Swarm of Robots," 2012 Brazilian Robotics Symposium and Latin American Robotics Symposium, Fortaleza, 2012, pp. 325-328.
- [12] J. G. B. F. Filho, C. E. T. Dórea, W. M. Bessa and J. L. C. B. Farias, "Modeling, Test Benches and Identification of a Quadcopter," 2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR), Recife, 2016, pp. 49-54.
- [13] Kon, J., Yamashita, Y., Tanaka, T., Tashiro, A. and Daiguji, M. (2013). Practical application of model identification based on {ARX} models with transfer functions. In *Control Engineering Practice*, v. 21, n. 2, p. 195-203.
- [14] Mathworks. (2013), <http://www.mathworks.com/products/simulink/>, agosto.
- [15] Mamdani, E. H., Assilian, S. (1975) "An experiment in linguistic synthesis with a fuzzy logic controller", In: *International Journal of Man-Machine Studies*, Elsevier, v. 7, n. 1, p. 1-13.

## Sistema Web para Gerenciamento de Vôos de Aeronaves não Tripuladas

Handrey Emanuel Galon<sup>1</sup>, Robison Cris Brito<sup>1</sup>, Fábio Favarim<sup>1</sup>, Eduardo Todt<sup>2</sup>

<sup>1</sup>DAINF – Universidade Tecnológica Federal do Paraná (UTFPR)  
Pato Branco – PR – Brasil

<sup>2</sup>DInf – Universidade Federal do Paraná (UFPR)  
Curitiba – PR – Brasil

moisesmeirelles.f, diegoaddan@gmail.com, robison@utfpr.edu.br, todt@inf.ufpr.br

**Abstract.** *Drones, or unmanned aerial vehicles, beyond being a model airplane hobby gain market space by acting as an important data capture tool and helping in routine tasks, such as in agriculture management areas and goods delivery. The main reasons of its increasing popularity are the technological advancements and improvements of the hardware. Thus, the objective of this work is the development of a flight management system for unmanned aircrafts aiming agricultural applications, allowing the management of areas and the actions performed by Drones, called as missions, through geolocation technologies. Through the system, it is possible to store the mission data, allowing the analysis of the production achieved by the end user. The system integrates several technologies, among them the Java Language used to the development of the management server. Also worthy of note is the use of the Angular framework for client-side development as well as the practical adoption of the Node.js platform for the development of the middleware responsible for integration and communication to the aircraft through local network.*

**Resumo.** *Drones, ou veículos aéreos não tripulados, tem deixado de ser apenas um passatempo de aeromodelismo para ganhar espaço no mercado atuando como uma ferramenta importante de captura de dados e auxílio a tarefas cotidianas, como gerenciamento agrícola e sistemas de entregas. Os principais motivos para sua popularização são os avanços tecnológicos e aperfeiçoamentos do hardware. Neste contexto, o objetivo deste trabalho é o desenvolvimento de um sistema web de gerenciamento de vôos de aeronaves não tripuladas, aplicado à agricultura de precisão. Este sistema possibilita o controle de quais áreas poderão ser sobrevoadas e ações realizadas pelos Drones, constituindo missões, com o auxílio de técnicas de geolocalização. Ainda, o sistema permite armazenar esses dados, apresentando histórico de missões para o usuário final. Para o desenvolvimento, foram utilizadas diversas tecnologias, dentre estas a Linguagem Java, que permitiu o desenvolvimento do servidor de gerenciamento, o framework Angular para o desenvolvimento do lado do cliente, bem como a utilização prática da plataforma Node.js para o desenvolvimento do middleware responsável pela integração e comunicação com as aeronaves via rede local.*

## 1. Introdução

Nas últimas décadas, sistemas de aeronaves não tripulados (UASs) ou Drones, quadricópteros, Veículos Aéreos Não Tripulados (VANTs), Remotely Piloted Aircraft Systems (RPAS) ou Unmanned Aerial Vehicles (UAVs), surgiram em um número crescente de aplicações militares e civis. Para os menos familiarizados, Drones, termo que pode ser utilizado para generalizar todas as outras denominações, são veículos aéreos não tripulados, podendo ser controlados por um controle remoto, computador embarcado ou mesmo através de um terminal.

[Bastianelli et al. 2012] destacam que existe uma demanda muito grande para uso de Drones, seja no meio militar para reconhecimento e missões, como também no meio civil, em aplicações como monitoração ambiental, mapeamento e agricultura de precisão. As aplicações para um Drone hoje são vastas e ainda pouco exploradas, de modo que ainda existem muitas dificuldades do mapeamento, controle e precisão das mesmas; aspectos que têm motivado diversos estudos científicos para desenvolver as melhores soluções em controle dos drones. Deve ser considerado ainda o fato de ser um equipamento que precisa ser manuseado com cuidado, o que dificultou por muitos anos sua homologação para uso comercial pela Força Aérea Brasileira [FAB 2015].

Os Drones têm se destacado na área agrícola, tendo [George et al. 2013] registrado que constituem as melhores plataformas para avaliar a produção, apresentando eficiência e rapidez acima das técnicas já utilizadas atualmente via meio terrestre tradicional, permitindo a leitura de dados em qualquer ambiente e superfície de terreno.

Os Drones ainda possuem limitações importantes, como a precisão de controle, a capacidade de se adaptar a falhas, segurança e autonomia de voo. Levando em consideração essas limitações, destaca-se a importância de existir formas de gerenciar o voo e supervisionar o funcionamento adequados dos Drones para obter um melhor aproveitamento dos seus recursos, bem como a captura desses dados para, em conjunto da tecnologia, obter análises e resultados relevantes para alcançar os objetivos de otimização em resultados no meio agrícola.

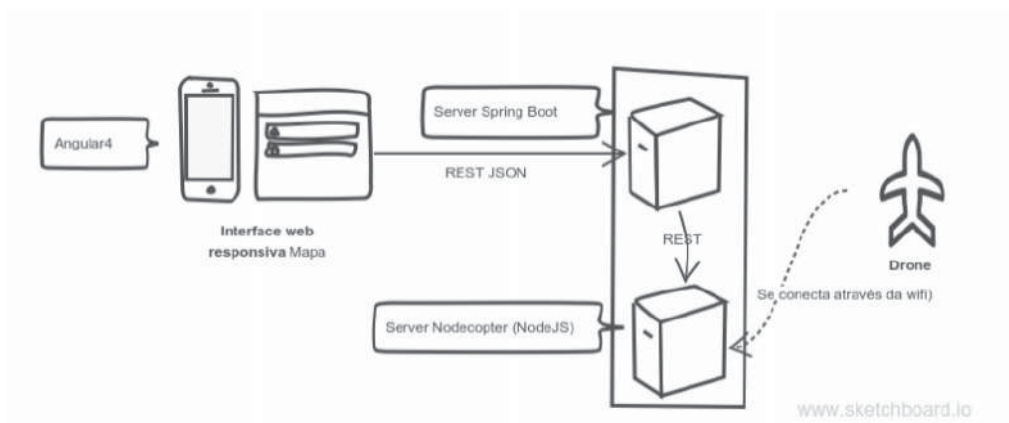
Nesse cenário, esse trabalho visa o desenvolvimento e avaliação de um sistema web para gerenciamento de missões realizadas por aeronaves não tripuladas. O sistema abrange o funcionamento do drone em campo, bem como o cadastro de dados relevantes para uma missão, desde a área até pontos de recarga para os drones. O trabalho apresenta ainda os conceitos técnicos de testes unitários e técnicas avançadas de deploy seguindo os melhores padrões do mercado.

## 2. Materiais e Métodos

Esta seção apresenta as ferramentas utilizadas para o desenvolvimento do sistema, bem como os procedimentos metodológicos adotados para tal desenvolvimento.

### 2.1. Materiais

Foram utilizadas exclusivamente ferramentas livres, visando minimizar o custo de desenvolvimento, assim como permitindo personalizar estas, caso seja necessário. Um dos objetivos do trabalho é verificar como uma grande quantidade de ferramentas distintas se comportam quando utilizadas em um mesmo projeto.



**Figura 1. Visão geral do sistema.**

Foi utilizado para o desenvolvimento o Node.JS como servidor de script e também para envio dos comandos par ao Drone. Angular como Framework Typescript. O Bootstrap para o desenvolvimento do front-end. IntelliJ IDEA como IDE de desenvolvimento. VSCode como editor Angular. PostgreSQL como banco de dados. Pgadmin para gerenciar o banco. Spring como framework servidor para segurança, configurações e autenticação e JUnit para os testes unitários.

## 2.2. Métodos

O processo de desenvolvimento do aplicativo web para gerenciamento de vôos de aeronave não tripulada foi realizado por meio de uma série de etapas, tendo como base o modelo sequencial linear como descrito em [Pressman 2005]. Inicialmente foram definidos os requisitos principais, obtendo-se uma visão geral do sistema. Esses requisitos foram complementados à medida que o sistema foi modelado e implementado. Algumas das partes consideradas mais difíceis, como a comunicação com o Drone, foram desenvolvidas por último.

A seguir estão descritas as etapas definidas para o desenvolvimento do sistema e as principais atividades de cada uma:

- Preparo e configuração do ambiente;
- Análise e projeto do sistema;
- Codificação;
- Implantação.

## 3. Resultados

Esta seção apresenta uma visão geral do sistema web desenvolvido, o Diagrama de Caso de Uso e Diagrama de Entidade e Relacionamento, assim como suas principais telas e funcionalidades.

A Figura 1 apresenta uma visão geral do sistema para facilitar o entendimento da comunicação entre as funcionalidades desenvolvidas.

O sistema web foi desenvolvida de forma responsiva em Angular 4, ou seja, pode ser acessado via navegador tradicional, como em um desktop, ou por um dispositivo

móvel. Estas informações são enviadas para um server desenvolvido em Spring utilizando a comunicação um REST JSON. Estes dados são compartilhados com um server em NodeJS, que envia a missão com os comandos que deverão ser executados para um drone.

Devido à complexidade, o sistema desenvolvido tem a função de fazer o cadastro e o gerenciamento de dados referentes às missões que poderiam ser realizados por um Drone, não sendo codificado ainda neste o envio dos comandos referente a missões propriamente dito (ex. levantar voo, vai até o ponto x no mapa, voa em linha reta, volta ao ponto de recarga e aterrisar), porém, para testar o ambiente e a comunicação, ao iniciar uma missão, é enviado apenas um comando para fazer o drone levantar voo, permanecer alguns segundos em voo e aterrisar logo na sequencia.

Após o desenvolvimento de Caso de Uso, foi definidos três atores para o sistema: Administrador, Fazendeiro e Sistema.

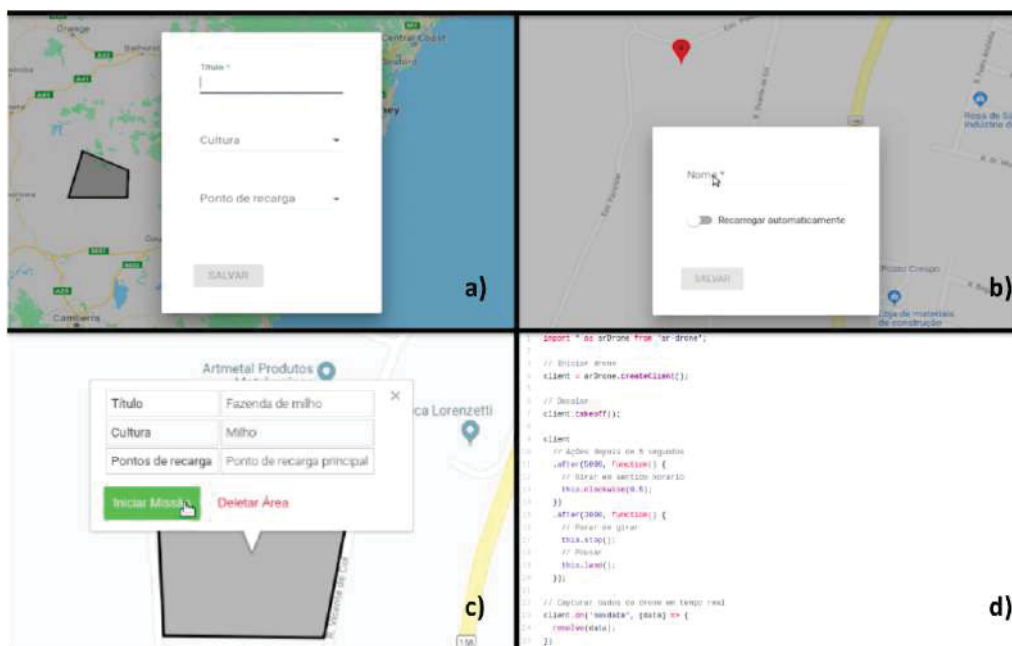
- O Fazendeiro é o principal do sistema, atuando diretamente com a maior parte das operações disponíveis, bem como a personalização e configurações relacionadas ao seu perfil.
- O Administrador é responsável pela manutenção geral do sistema, configurações, permissões e supervisão dos usuários.
- O sistema é responsável pela conexão com os Drones por meio do middleware de rede para conexão com os dispositivos do Node.JS.

Após o levantamento de requisitos e a análise realizada, o passo seguinte foi o desenvolvimento da telas do sistema, assim como o fluxo de informação entre elas.

Basicamente, o sistema se resume em cadastrar e gerenciar missões que podem ser realizadas por drones. Para isso, outros cadastros secundários são necessários, como o cadastro de drone, de áreas para atuação da missão, pontos de recargas, entre outros.

O uso do sistema pode se resumir em cinco passos, conforme apresentados:

1. **Login:** No primeiro acesso, o usuário precisa se registrar no sistema, sendo o acesso deste validado pelo administrador.
2. **Cadastro de cultura:** Um dos primeiros cadastro do usuário deve ser o cadastro de culturas possíveis (ex. milho, arroz, feijão), este cadastro será associado posteriormente a uma missão do drone.
3. **Cadastro de Drone:** Na sequência, será necessário um cadastro de Drone, informando quais os veículos aéreos que serão utilizados nas missões, assim como suas características físicas.
4. **Cadastro de bases de Recarga:** Em uma missão com drones autônomos, os mesmos podem realizar recargas sem a interação do usuário, para isso é necessário o cadastro das bases de recargas disponíveis no sistema, assim como suas características físicas.
5. **Cadastro de Área:** É necessário um cadastro das áreas nas quais os Drones poderão ser utilizados. Nesta, é utilizada uma ferramenta gráfica integrada ao Google Maps para informar onde os vôos de uma missão podem ser realizados.
6. **Cadastro de missão:** A missão é o ponto mais importante do sistema proposto, neste é solicitado em qual cultura será realizada a missão, a área onde a missão será realizada, também será selecionado o drone responsável pela missão, a base



**Figura 2. Cadastro de ponto de Recarga.**

de recarga que será utilizada assim como seu local dentro da área da missão. Por fim, o sistema está pronto para iniciar a missão, enviando as instruções para o Drone.

Dentre os inúmeros cadastros do sistema, alguns merecem destaque, como o cadastro de ponto de recarga (Figura 2.a), que permite ao usuário selecionar, por meio de um mapa, o local onde será alocado um ponto de recarga. Esta seleção acontece após um clique duplo na tela, onde é solicitado dados cadastrais do de ponto.

Da mesma forma, outra tela que merece destaque pela interação com o usuário é o cadastro de área, onde o usuário, a partir de um mapa, pode clicar nos vertices da área de atuação em que o drone irá realizar a missão. Ao completar o perímetro de ação do drone, é apresentado uma nova tela para informar novos dados sobre a área, conforme apreentado na Figura 2.b.

Por fim, selecionando uma determinada área, é possível iniciar uma nova missão, associando a esta um drone. A tela para iniciar uma missão é apresentada na Figura 2.c.

Para os testes de comunicação entre o sistema e o Drone, após iniciar uma missão, o sistema envia para o Drone, via Node.JS, instruções para o drone selecionado para a missão levante voo, permaneça no ar por alguns segundos, e na sequencia, pouse. Neste primeiro momento não foi codificada a execução da missão em si pelo drone, o objetivo foi apenas testar o sistema web para gerenciamento de voos de aeronaves não tripuladas. A única instrução enviado pelo sistema ao drone, ao iniciar a missão, é a solicitação para levantar voo, permanecer no ar por alguns segundos e pousar.

O código utilizado para o envio destes comandos é apresentado na Figura 2.d.

## 4. Conclusões

Como forma de exemplificar o uso da Drones no meio agrícola, o presente trabalho teve como objetivo a implementação de um sistema web utilizando várias ferramentas e frameworks para gerenciamento de aeronaves não tripuladas em missões para monitoramento agrícola.

O trabalho realizado apresenta a forma de uso da comunicação entre sistemas e drones utilizando a linguagem Java e o framework Node.JS. Os dados referentes as missões, como quando ocorreu, drone associado a este, assim como base de recarga, cultura e área de atuação são visualizados e manipulados por um sistema web, que possui uma interface de usuário.

O Node.js se mostrou um framework eficiente e de alto desempenho, disponibilizando o middleware de comunicação com os Drones de maneira eficaz e ainda disponibilizando uma interface de comunicação HTTP de maneira simples atendendo o objetivo de comunicação com as aeronaves.

O Spring é um framework para desenvolvimento web e mostrou-se muito eficiente por ser bem documentado e fornecer os recursos que facilitaram o desenvolvimento do sistema web. Entre os destaques do Spring pode-se citar: módulo de autenticação de usuários, rotas e controladores.

Os objetivos deste trabalho foram alcançadas com êxito. Como um dos objetivos foi verificar como diferentes tecnologias open-sources se comportavam em um projeto real, foram realizados muitos estudos e integrações, sendo muitos frameworks e ferramentas utilizadas e integradas, e todos atenderam seus objetivos.

Como sugestão de trabalho futuro, fica a possibilidade de fazer um upgrade no sistema, permitindo o cadastro e o uso de múltiplos veículos aéreos não tripulados em uma única missão, assim como várias bases de recargas. Atualmente, o sistema só permite o uso de um drone/base de recarga por missão.

O passo seguinte deste projeto é desmembrar a missão em instruções de voo para o drone, permitindo que este realize a missão à qual este foi destinado, lendo os dados da missão e armazenando após a aterrissagem em um servidor com os resultados da missão.

## Referências

- Bastianelli, G., Salamon, D., Schisano, A., and Iacobacci, A. (2012). Agent-based simulation of collaborative unmanned satellite vehicles. In *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*. Institute of Electrical & Electronics Engineers (IEEE).
- FAB (2015). Comando da aeronáutica publica nova legislação sobre aeronaves remotamente pilotadas. In <http://www.fab.mil.br/noticias/mostra/23937/>. Acessado em 12 de maio de 2016.
- George, E. A., Tiwari, G., Yadav, R. N., Peters, E., and Sadana, S. (2013). UAV systems for parameter identification in agriculture. In *2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS)*. Institute of Electrical & Electronics Engineers (IEEE).
- Pressman, J. R. (2005). *Engenharia de software*. 5ª ed. Rio de Janeiro: McGrawHill.

# A Model for landing, taking off and autonomous battery recharging of a Parrot Ar.Drone 2.0 using computational vision and GPS features

Robison Cris Brito  
Federal University  
of Technology - Paraná  
Campus Pato Branco  
robison@utfpr.edu.br

José Felipe Loureiro  
Federal University  
of Technology - Paraná  
Campus Pato Branco  
zefelippe\_0@hotmail.com

Carlos Eduardo Magrin  
Federal University  
of Paraná  
UFPR  
cmagrin@gmail.com

Eduardo Todt  
Federal University  
of Paraná  
UFPR  
todt@inf.ufpr.br

**Abstract**—A drone is a type of Unmanned Aerial Vehicles (UAV) usually equipped with four propellers. They can be used in many applications, one of those is to move through places of difficult access. Besides practicality over other aerial vehicles, its price is lower compared to large vehicles, which turns them attractive to many activities. Also it offers safety in dangerous situations, like fires or accidents, as it doesn't need an on-board pilot. In a system with autonomous flight the concern with its landing and recharging of the batteries, which doesn't last more than a few tens of minutes, arises. Using on-board devices, like its cameras and GPS modules, it is possible to implement functions to expand its capabilities. With the goal to present a solution to such problem, this essay proposes a model which utilizes image recognition to guide the landing in an autonomous fashion. This landing routine based on image turns landing into an autonomous processes, easing the landing for recharge process.

**Keywords**—Drone, Autonomous Landing and Taking Off, Computer Vision, GPS, Parrot AR. Drone

## I. INTRODUCTION

With the technological advances in electronics, unmanned aerial vehicles (UAVs) have become increasingly accessible, either due to their decreasingly cost and for the options available in the market.

The US research group Teal estimates that world production of unmanned aerial vehicles will increase from U\$ 2.6 billion in 2016 to U\$ 10.9 billion in 2025, with an annual growth rate of 15.4 %. Over the next decade the market will amount to a total of U\$ 65 billion [1].

In the civilian scope of application, some of the existing demands are environmental monitoring, such as pollution, climate or scientific applications, forest fires, border control, drug traffic, aerial surveillance, mapping, traffic monitoring, precision agriculture, search and rescue, among others [2]. There is still a very large demand for the use of UAVs in the military, either for reconnaissance, surveillance, damage assessment after an attack, or as a contingency mechanism and for communication.

Among the types of UAVs are the drones, also known as quadcopters, which are usually smaller and, most of the time, simpler aerial vehicles. According to [3], drones have gained space, among other areas, in precision farming and

livestock farming. Their versatility is worth the investment, since they can perform diverse functions in agricultural regions of difficult access and they have a relatively low cost.

Regarding the technological resources present in the drones, practically all current commercial models have video and / or photo cameras, ultrasound sensors to measure distance from the ground, GPS module for self-localization and modules for remote communication using radio-frequency, however, one can find more sophisticated drones with a greater number of sensors and features. As an example of drone with the basic resources cited is the AR.Drone 2.0 from the French company Parrot, with an approximate cost of U\$350.

Although very useful, a drone is an equipment that needs to be handled with care, a fact that made it difficult for the homologation for commercial use by the Brazilian Air Force [4]. This is due to its rotating propellers and their weight, which can vary from a few grams to several kilograms (for homologation, the drones were divided into three categories: up to 2.5 kg, from 2.5 kg to 25 kg and above 25 kg). A drone can cause serious injury in the event of an accident or fall. In all categories, by homologation, autonomous flight can only occur in non-populated areas.

To minimize the risks, the National Civil Aviation Agency (ANAC) published in late 2017 the new regulation on commercial use of drones, which promises to boost and bring greater security to its commercial use in Brazil. The Brazilian regulations follow the line of action adopted by the International Civil Aviation Organization (ICAO), based on amendments to the Annexes of the Chicago Convention. The regulation that addresses the use of Brazilian airspace by remotely piloted aircraft must be constantly revised and adjusted, given the dynamic nature of the activity and the recurrent technological advances [4].

The major disadvantage of using drones is still the limitation of autonomy in battery vehicles. Typically quadcopter drones have a flight autonomy of up to 30 minutes, however, low-cost drones such as the Ar.Drone from Parrot have just 10 minutes. In addition, after the battery has run out, the recharging process is slow and requires human supervision.

Aiming to present a feasible alternative to overcome this limitation, this work presents a model for the solution of two of



the major limitations related to the autonomous flight of drones - their landing/takeoff and battery charging, as human intervention is now necessary for both. In order to test the model, a prototype of the recharge base was developed, as well as an adaptation in the Parrot AR.Drone 2.0 to allow an autonomous recharge, and a software was also developed to communicate with the AR.Drone 2.0 during the landing/recharging/take-off process.

## II. STATE OF ART

For the development of this work, a systematic mapping was carried out on works that involved cooperation between vehicles, and part of this study selected works that involve autonomous takeoffs/landing/recharging [5]. These are detailed in the sequence.

[6] presented a cooperative approach between a UAV and an Unmanned Terrestrial Vehicle (UTV) based on monocular vision for UAV landing. Using tracing to mark a target on the UTV, the UAV can track and land autonomously on the moving UTV. Control loops based on PID controllers are employed to perform two levels of control: stabilization control and position control. The stabilization control is performed on an on-board micro controller board with the aid of an inertia measurement unit. The images captured by the on-board camera are transmitted to a central terrestrial control unit, which is present on a laptop via wireless channels. The relative position of the UAV for the land vehicle is estimated from the received images, not by GPS, so the estimated current position of the UTV is sent to the UAV via Wi-Fi. The proposed vision-based approach to detect and locate the target, as well as the height of the UAV, is robust when the UTV is easily. Practical experiments segmented from the show that the UAV can autonomously monitor the UTV and background perform landing on the moving target. This work deals exclusively with the autonomous landing, not addressing the autonomous reloading and take-off operation.

[7] has explored the UAV drive within an airport. Using the UAV front camera video, GPS information is also used for the auto-location of the UAV. The GPS, however, introduces a systematic error in the system, especially in the indoor areas. The project developed by the authors presents a new approach in which the processing of the images allows a rectification of the GPS signal, making use of the automatic recognition of airport signs and marks. The take-off paths as well as the taxiway location signs are identified and correlated with the GPS information. Several approaches were made using Hough's transformation to find the runway for landing or takeoff without cooperation between position and optical systems. The goal of the work is to create a routine that automates landing and takeoff based on images to improve the accuracy of GPS in indoor areas, but it does not address the autonomous recharging.

In another project, [8] addressed cooperation between unmanned devices by performing missions in indoor/outdoor areas. In this, the authors focus on the interaction between UTVs and UAVs to extend the autonomy of flight of a UAV by means of a landing platform. The UTV acts as a recharging station and hosts the UAV during the indoor/outdoor transition and vice versa. The platform was designed with the goal of

achieving a robust landing. Synchronization and co-ordination of co-operation are managed by a control station. This station was developed using a software tool based on the integration of Stateflow, automatic generation of C code and ROS (Robot Operating System). All software components of UAV, UTV and the station itself were developed using ROS. The results show that the UAV was able to land on the UTV with high precision ( $< 5$  cm for the  $x$  and  $y$  axes) thanks to a visual position estimation algorithm. The present study, however, did not address the autonomous recharge of the UAV.

Also in the vehicle interaction scenario, [9] presented a method of cooperation between two UAVs, one flying in high altitude and the other in low altitude, to achieve autonomous navigation and landing. In the autonomous landing based on computational vision, the accuracy of the GPS signal and the efficiency of the tracking algorithm and target detection affect the performance of the autonomous landing system. In this way, by using the comprehensive view and the high flexibility of an UAV at high altitude, it is possible to control an UAV at low altitude so that it can perform the landing procedures correctly. The flight controller can track the target and control the device in real time. This is possible through a high level control system using fuzzy logic and neural networks to calculate the positioning and perform maneuvers of the low altitude device. The present work does not present a model for autonomous recharging and takeoff.

[10] state that aerial surveillance and mapping are the main areas for developing applications that involve small UAVs, such as drones. When it comes to the mapping of large areas, for example, one can have missions that last a longer time. However, the duration of the mission is limited by the ability of the UAV battery. This requires reloading the UAV for the success and completion of the mission. Points of interest in the area to be mapped may not be accessible from a single recharge station and therefore multiple stations are required. In addition, recharging stations could not be placed anywhere in the region due to terrain restrictions. Finally, the routes of the UAVS missions should be planned according to the layout of the refueling stations. So UAVS route planning and the provision of recharging stations are connected problems, and getting optimal solutions becomes difficult. The authors developed a greedy strategy of coordination between a UTV and the UAV, with the help of a simulation platform created with the help of mathematical software Matlab to test and validate the strategy. Field tests were conducted using a single drone. The proposed model predicts the landing/reloading/autonomous takeoff, however, they were only tested with simulations.

[11] have been responsible for the Heterogeneous Autonomous Mobile Maritime Expeditionary Robots (HAMMER) project, which aims to integrate an autonomous surface platform with three different types of unmanned vehicles: aerial, surface and submarine. The HAMMER system consists of several marine vehicles working together, among them the unmanned surface vehicle (USV) acts as the central node and main transport mechanism, it can be used to transport UAVs and unmanned underwater vehicles (UUV). The system is designed to be modular and can be easily extended. For the implementation of the USV, a 16-foot catamaran (70x40-in landing area) was used. By using the surface vehicle for the base of operations of a UAV, it is possible to guarantee the

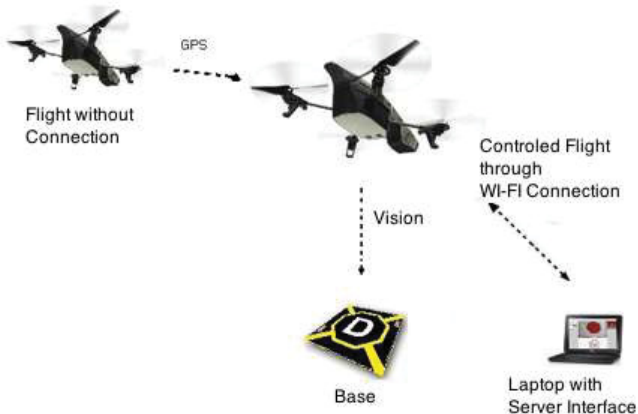


Fig. 1. Landing/take-off and recharging model using GPS and drone image processing.

interoperability, coordination and cooperation of autonomous mobile marine robots in environments in which access to the GPS is denied. The three main research areas related to the project are image processing, state estimation and autonomous cooperative control. Due to the challenges of the maritime environment, both software and hardware used must provide the level of flexibility and resources required to achieve an efficient and robust landing. The communication should also be reliable, being developed to allow the efficient sharing of data and control messages between the surface and the aerial platforms. Although it deals with landing and take-off autonomous, the work does not deal with recharging the battery.

All the works presented in this section use some form of landing and/or autonomous take-off of an UAV. All have their merits, but the recharge action, given the technical limitations, were not addressed.

The model proposed by this work aims to automate besides the landing and takeoff, the autonomous recharge of the battery, being this model implemented on a recharge base for a Parrot AR.Drone 2.0 using GPS and computer vision features.

### III. RESULTS

The aim of the work is to propose a model for landing/take-off and autonomous reloading of an Parrot AR.Drone 2.0. Its approach to the base is accomplished through the aid of a GPS and its landing happens with the aid of image processing on the images acquired by the drone's on-board camera. As soon as the drone connects to the base, the charging of the battery begins. Upon completion of the recharge, the drone is ready to take off and continue its mission. The model is shown in Fig. 1. Communication between the recharging base and the drone is done via a Wi-Fi connection. As the drone approaches the base, it establishes a Wi-Fi connection and transfers the images captured by the drone. The base is responsible for processing these images and sending maneuver commands to the drone, however, this happens only when the drone is in the process of landing, that is, near the base.

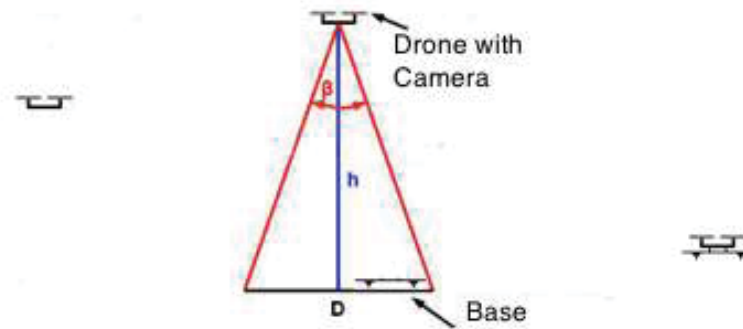


Fig. 2. Calculation of the control transition height, below this height control is transferred from the drone to the recharging base.

For the prototype, it was decided to use a drone that has stood out in the world market for its reduced price and ease of use. The Parrot AR.Drone 2.0 is equipped with a 32-bit processor, has Linux operating system and allows its control of the flight through a remote device (smartphone, computer, micro-controller, among others), and it must be connected to the drone using Wi-Fi. Its battery allows flights of 10 minutes in average. This drone can be equipped with the FreeFlight - GPS module for its self-location capability. For these reasons the AR.Drones are commonly used in scientific research, as presented in [12], [13] and [14].

The purpose of our job is to assist a drone in its landing when there is a need for autonomous charging, allowing the battery to charge and leaving it fit for subsequent take-off. For this purpose, the drone should, at the time it detects a low battery signal, find the geostationary coordinate of the nearest recharge base. This information will be present in the micro-controller of the drone, being processed by an embedded software, which constantly reads the battery level and when being lower than a threshold value, will cause the drone to fly to the recharging base.

Being near the base a Wi-Fi communication between the drone and the base is established, initiating the process of descent, which uses as a parameter the image recognition obtained the drone itself. This causes the drone to align with the base and make the landing. Charging occurs when the drone's feet touch the metallic contacts of the recharge base.

In order to identify the minimum height at which the drone needs to be for the correct recognition of the image of the recharge base, a trigonometric function was required, as shown in Fig. 2. Considering that the GPS module of the equipment has a precision of around 2 meters, that is, given a certain geostationary point, the drone will recognize it being 2 meters more or less from the point (D). The angle of vision of the lower camera is 63 degrees ( $\beta$ ). Having these two information at hand and using trigonometry it's possible to figure out the height (h) that the drone must fly to catch the base in its field of view, which is at least 3.5 meters. Thus, when the drone descends and reaches this height, its control is carried out from instructions sent from the base to the drone, which forwards instructions for an accurate landing on it, based on the images received from the drone itself.

For the landing of the drone a base is needed that is easily recognized and that facilitates the landing and recharging. In



Fig. 3. Representation of the drone approach and contact if the drone feet with metallic contacts at the base.

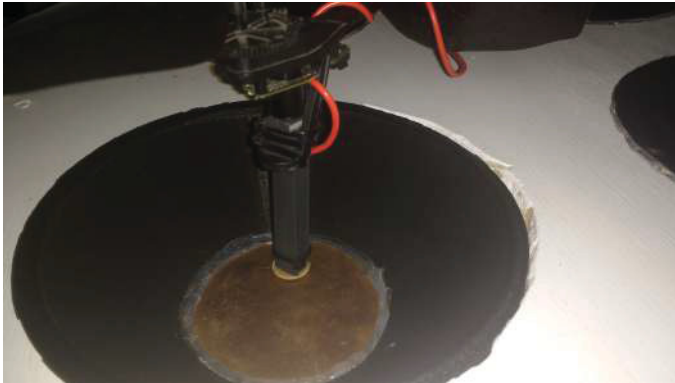


Fig. 4. Adaptation of the contacts on the drone feet with the metal contacts on the base.

this way, the base was designed as a circumference, with signs to identify the direction the drone should land (the front of the drone is facing the corresponding side of the base). It was chosen a base in the circular format to aid the identification of this by means of algorithms of image processing, since geometric images are easily identified in this type of algorithm. In the same way, the red color was chosen to identify the position in which the drone should descend. This color was chosen to have a counterpoint to the land area, which is normally an agricultural area, the background is usually green when there is vegetation, or brown of exposed soil.

Geometric objects are relatively easy to identify due to their specific characteristics such as perimeter, area and radius. Even on the basis of the image, the precision required for the contact must be high. In this way the base has 4 holes in the shape of an inverted cone, allowing the drone to land even with a small positioning error, the contact is driven by gravity to the correct position for recharging. Thus, each foot of the AR. Drone will fit into one of these orifices of the inverted cones, as shown in Fig. 3. Each cone has 9 cm in radius and 12 cm in height.

The drone battery recharge is achieved by means of an adapter in the base that must be connected to a power source. The feet of the drone were fitted with metallic tips, connected to the three cells of the battery, so that at the moment of landing, they make contact with charger. The tip of the cone contains a metal base to ensure contact with the battery poles. The Fig. 4 demonstrates the adaptations.

The base was made using a MDF board, measuring 60 centimeters in circumference, in which were drilled four circular holes 9 centimeters in radius. These holes serve



Fig. 5. Prototype of the recharge base developed for landing and recharging of an ArDrone.

as accommodation for the 4 designed cones, which were manufactured with a 3D printer. The contact plates present on the inside of the cone were made using circular-shaped electrical plates and were welded to the cables connecting to the charger. The base and cones are shown in Fig. 5.

For reasons of compatibility with the AR.Drone control environment, the server runs on a notebook connected to the recharge base. The server software was developed in Node.JS and Angular.JS using the JavaScript language. These tools facilitate the development of activity due to AR.Drone-compatible libraries, available for Node.JS, which contains functions for drone handling. This server is connected to the recharge base and is responsible for processing the information (images) sent by the drone. Based on these, images commands and maneuvers are sent to the drones.

For the application, a web server is required for message exchanges and communication with the drone. ???This system boils down to a screen for displaying the image received from the drone, as well as the result image after recognition of the desired color, and a few buttons to control the drone.??? This graphical interface was used only for the tests, and interaction with the human being in the real application is not necessary. This application is also responsible for sending the flight commands, processing the drone image, identifying the drone's position to handle the landing, and sending this command via Wi-Fi as a response to the drone. The drone can be considered a Wi-Fi router, which accepts connections from various devices, in this case the recharge base. Fig. 6 demonstrates the communication between drone and base.

As Node.JS and Angular.JS have great compatibility with JSON and both use JavaScript for development, client and server were created in such a way that they communicate via JSON. The client and the access to the information were codified using the technologies HTML and Angular.JS, since they are languages widely used in the Internet and are compatible with each other. The drone captures its images in JPEG format and they are handled by the server using the OpenCV open library.

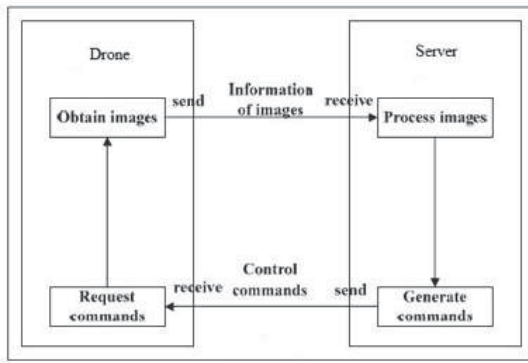


Fig. 6. Representation of communication between drone and server.



Fig. 7. Algorithm for color detection and image processing.

OpenCV is an open source library of computer vision and machine learning, which facilitates its use, as well as the modification of the code by users. The library has more than 2500 optimized algorithms, which includes a comprehensive set of computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, track moving objects, extract models of 3D objects, and more.

To identify the recharge base, an algorithm based on color detection and geometric figures was defined. Figure 7 shows the flow of the algorithm.

Generally, it is assumed that the RGB color space is more suitable for color-based segmentation, however, the HSV color space is the most appropriate. The HSV color space consists of 3 matrices, which are the matrix of hues, saturation and values. In OpenCV, the range of values for the fields of the matrices are, respectively, 0-179, 0-255 and 0-255. Hue represents the color, saturation represents the amount with which the respective color is mixed with white and value represents the amount with which the color is mixed with black (Szeliski, 2010). In this way, when the image is transformed into the HSV space, a color ends up being defined by a single parameter, instead of three as it is in the RGB, thus facilitating the processing.

The red color was chosen to be used as reference color for the drone approaching the base, and has values between 170-180, 160-255, 60-255. Here, the hue is defined exclusively for this color distribution. Already saturation and value may vary depending on the ambient lighting condition. In this way it is necessary to perform two searches, one for the lower limit of the values and one for the upper limit. Finally, adding the two images we have a third image as a result, which contains only the desired pixels.

So it is possible to extract relevant information from the result image. Since the image contains only the pixels in which the red color was recognized, a function is used to discover the center of the present area. With this we have the values of X and Y necessary to control the centering of the drone in

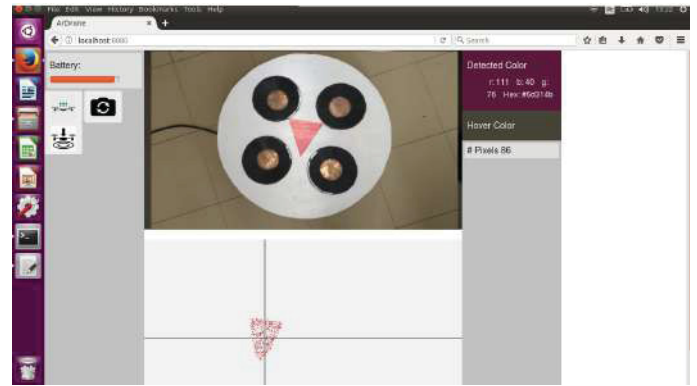


Fig. 8. Application main screen.

relation to the base.

The application consists of a single interface, containing a battery meter, three buttons, one to take off, one to land and the last to change the camera being used (since the AR.Drone has two cameras, one in the front and one in the bottom), two image fields, one displaying the original image received by the drone and the other the image already processed, and finally an interface that allows the choice of the color to be searched for. The Fig. 8 displays the application screen.

The algorithm presents satisfactory results, recognizing the red color among others tested. A major problem encountered is the intensity of illumination on the object in focus. Often a shadow or difference of illumination in an object causes the incorrect identification of colors that should be found. The image quality generated by the AR.Drone's lower camera is lower than the front camera and did not perform well. Even with the problem of illumination in uncontrolled environments, the correct identification of the base happened at about 80 % of the descents.

As the project progressed, some problems had to be addressed. According to the AR.Drone 2.0 User's Guide, the device must not have its battery charged while the device is on. This problem has been solved by making a specific circuit to cut off the drone's power at the moment of landing for a set time (40 minutes, time required to fully charge the drone). An opto-coupler drive circuit was designed to ensure insulation of the components of the drone's from the battery recharge, so that upon landing, a signal received from the base triggers the TRIAC and reverses its logic of action, cutting the energy of the micro-controller.

In this way, as the present model is projected, when an AR.Drone has the battery at critical level, it pauses the mission (the mission is present in the drone micro-controller), returns to the base of recharge, executes the landing procedure, stays recharging the battery for 40 minutes. After that the recharge base disconnects from the drone, which takes off and continues the mission from where it paused.

#### IV. CONCLUSION

Drones, here understood as quadcopters, are used very frequently by both companies and anonymous users, causing an increase in the general interest of the product. Among the

existent limitations of a drone are the low autonomy of flight, as well as landing and recharging with the need for human intervention. But the advantages are many, it has greater ease in the maneuvers compared to UAVs in the aircraft format, the data collected by drones are usually more precise since they can fly over an area at low speed and even stop at mid-flight . It also allows the increase or decrease of flight altitude at one point for capturing more detailed or larger images depending on the necessity.

Despite all these advantages, the great limitation that inhibits it in everyday use is the battery autonomy. For small flights or remotely controlled flights it ends up being enough, however, for long missions, as its application in precision agriculture, its use ends being impaired.

The present work developed a model for landing, recharging and autonomous take-off of a Parrot AR.Drone 2.0. In spite of the initial difficulties, after a series of studies and calibration, it was possible to have the drone approach the base, descend, recharge the battery autonomously, being fit for a new flight.

The use of the Node.JS language was of great value for the project, especially considering the integration of the drone with the recharge base, as well as for the development of a friendly visual interface for tests. As the language has a specific library to assist the control of the AR.Drone, it has facilitated the control of the drone. In the same way, the JavaScript language is fundamental to web application development. The project shows that it is possible to achieve the desired goals.

The drone used can be obtained online with prices starting at \$ 300.00. The cost for the base and cones of the base was R\$ 20.00. You also need a computer for the application server (this can be replaced with a Raspberry Pi).

The project is feasible and with great potential for improvement. With the help of the OpenCV library it was possible to assemble several applications based on image recognition. The biggest problem is with the image quality of the lower camera of the drone, which is of low quality and often makes processing difficult.

As an alternative to future works, the lighting control of the base could be improved, increasing the drone's descent efficiency (a faster and more accurate recognition of the base), also tests with new colors and new base formats, as well as transforming the prototype into a more professional product, already embarking in the own drone the circuit that turns off the drone when it is in the base of recharge, avoiding any damages to the drone.

## REFERENCES

- [1] PHILIP FINNEGAN. World unmanned aerial vehicle systems, market profile and forecast 2013. teal group research. In <http://tealgroup.com/index.php/about-teal-group-corporation/press-releases/129-teal-group-predicts-worldwide-civil-uas-production-will-total-65-billion-in-its-2016-uas-market-profile-and-forecast/>. Acessado em 23 de maio de 2018., jun 2013.
- [2] Gabriele Bastianelli, Diego Salamon, Alessandro Schisano, and Alessandra Iacobacci. Agent-based simulation of collaborative unmanned satellite vehicles. In *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*. Institute of Electrical & Electronics Engineers (IEEE), oct 2012.
- [3] Teresa Raquel Bastos. 15 usos de drones na agricultura e na pecuária. In <http://revistagloborural.globo.com/Noticias/Pesquisa-e-Tecnologia/noticia/2015/05/15-usos-de-drones-na-agricultura-e-na-pecuaria.html>. Acessado em 23 de maio de 2018., ago 2015.
- [4] ANAC. Resolução n429, de 02 de maio de 2017. In *Requisitos Gerais para Aeronaves não tripuladas de uso civil*. Agência Nacional de Aviação Civil., mai 2017.
- [5] Robison Cris Brito, Jose Felipe Loureiro, Eduardo Todt, and Roberto Pereira. A systematic mapping for the scenario of non-urban autonomous vehicle cooperation systems. In *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*. IEEE, nov 2017.
- [6] HUI Cheng, YOUSHENG Chen, XIAOKUN Li, and SHING Wong Wing. Autonomous takeoff, tracking and landing of a uav on a moving ugv using onboard monocular vision. In *32nd Chinese Control Conference*. Institute of Electrical and Electronics Engineers (IEEE), aug 2013.
- [7] Federico Francesco Barresi and Walter Allasia. Airport markings recognition for automatic taxiing. In *Conference: Design and Architectures for Signal and Image Processing (DASIP)*. Institute of Electrical and Electronics Engineers (IEEE), aug 2013.
- [8] Francesco Cocchioni, Valerio Pierfelice, Alessandro Benini, Adriano Mancini, Emanuele Frontoni, Primo Zingaretti, Gianluca Ippoliti, and Sauro Longhi. Unmanned ground and aerial vehicles in extended range indoor and outdoor missions. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, may 2014.
- [9] Min-Fan Ricky Lee, Shun-Feng Su, Jie-Wei Eric Yeah, Husan-Ming Huang, and Jonathan Chen. Autonomous landing system for aerial mobile robot cooperation. In *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*. IEEE, dec 2014.
- [10] Parikshit Maini and P. B. Sujit. On cooperation between a fuel constrained UAV and a refueling UGV for large scale mapping applications. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2015.
- [11] Vladimir Djapic, Christopher Prijic, and Frank Bogartz. Autonomous takeoff & landing of small UAS from the USV. In *OCEANS 2015 - MTS/IEEE Washington*. IEEE, oct 2015.
- [12] Alexandros Lioulemes, Georgios Galatas, Vangelis Metsis, Gian Luca Mariottini, and Fillia Makedon. Safety challenges in using AR.drone to collaborate with humans in indoor environments. In *Proceedings of the 7th International Conference on Pervasive Technologies Related to Assistive Environments - PETRA 14*. ACM Press, 2014.
- [13] Jonathan M. Aitken, Owen McAree, and Sandor M. Veres. Symbiotic relationship between robots — a ROS ARDrone/YouBot library. In *2016 UKACC 11th International Conference on Control (CONTROL)*. IEEE, aug 2016.
- [14] Rafael Duarte. Low cost brain computer interface system for ar.drone control. 2017.

# A Comparative Approach on the use of Unmanned Aerial Vehicles kind of Fixed-Wing and Rotative Wing Applied to the Precision Agriculture Scenario

Robison Cris Brito  
Federal University  
of Technology  
UTFPR  
robison@utfpr.edu.br

Nicollas Saque  
Federal University  
of Paraná  
UFPR  
nisaque@live.com

Jose Felipe Loureiro  
Federal University  
of Technology  
UFPR  
zefelipe\_0@hotmail.com

Eduardo Todt  
Federal University  
of Paraná  
UFPR  
todt@inf.ufpr.br

**Abstract**—To use Unmanned Aerial Vehicles in precision agriculture is a very common and efficient practice, however, the vast majority of applications make use of Fixed-Wing Aerial Vehicles due to their autonomy that is several times higher than that of the Rotating-Wing Aerial Vehicles type. Meanwhile, as its autonomy increases, so does its cost, and with the price of a single fixed-wing vehicle it is possible to purchase a swarm of several rotary-wing vehicles. The present study presents a theoretical and practical comparison to identify the most favorable scenarios for the use of Fixed-Wing and Rotating-Wing type air vehicles, making a qualitative comparison with data obtained from the vehicles Datasheet and a quantitative comparison, divided into two moments: with theoretical data calculated based on data retrieved from Datasheets, and with real data obtained from flights. At the end of the work, it is concluded that for linear flights, for example the monitoring of roads or frontiers, the superiority is held by fixed-wing vehicles. As for flights monitoring two-dimensional areas, where precision agriculture is commonly applied, the superiority is that of rotating-wings, which are able to do the same work in a lesser time than fixed-wings, with the advantages of having lower cost and greater flight dynamic, given their maneuverability, in addition to being easily applied on the autonomous flights, since the operations of takeoff and landing do not need the human interaction, since they happen vertically.

**Keywords**—charging station insert, Voronoi, Viktor Grabarchuk

## I. INTRODUCTION

Drone, Quadricopter, Unmanned Aerial Vehicle (UAV), Remotely Piloted Aircraft (RPAS) and Unmanned Aerial Vehicle (UAV) are some of the name variations for remotely controlled unmanned aerial vehicles and / or autonomous. The UAVs, to generalize the various denominations attributed to unmanned aerial vehicles, were originally a hobby, but with the evolution and accessibility of new technologies, what was called aeromodelling, today is treated with seriousness, since its applications have been growing in the most several areas.

For [1], there is a very large demand for UAVs. In the military, they can be applied in site recognition, surveillance, damage assessment after natural disaster or as a contingency mechanism for communication. In the civil area, other demands are presented, such as environmental monitoring (pollution, climate or scientific applications), fire monitoring, border

control, combating drug traffic, aerial surveillance, mapping, traffic monitoring, humanitarian aid, searches and salvage, precision agriculture, among other applications.

The importance of UAVs in precision agriculture is highlighted by authors such as [2] who register that UAVs provide the best platform to evaluate agricultural production, being fast and efficient compared to traditional terrestrial techniques, allowing reading on virtually all types of terrain. UAVs are versatile devices that can be used in a wide variety of agricultural applications, such as automatic monitoring of water and irrigation canals and channels, detection and control of agricultural and livestock pests, in the count of herd animals, for example.

From a technical point of view, currently two types of UAVs are used in precision agriculture: the fixed-wing UAVs, which have the shape of a small airplane, known as Remotely Piloted Aircraft Systems (RPAS), and the rotative wings, like a Drones.

Data from the systematic mapping developed by [3] indicate that 83% of the publications considered by the authors involve fixed-wings. This superiority is believed to be due to the fact that they are the first unmanned aerial vehicles developed, as well as being applied to precision agriculture for many years, and also because of their autonomy, which is superior to the rotative wings.

However, this second has several other advantages, such as greater accuracy in reading the data, since they can fly more slowly without the risk of falling. They have greater ease of maneuver, since unlike the fixed-wings, they can make sharp curves changing direction quickly, they also do not need the human interaction in the takeoff and landing, since the nature of these operations are vertical. There are also researches for the autonomous operations of landing, battery recharging and take-off ([4], [5] and [6]). Finally, it stands out as an advantage its cost, since a rotative wings can cost up to ten times less than a fixed-wing UAV.

The following article presents a comparative analysis between these two types of unmanned aerial vehicles: fixed-wings and rotative-wings, to identify, which is indeed the best to be applied in precision agriculture, being that historically, fixed-wings are the most used.

In the following chapters will be presented the technical characteristics of these two aerial vehicles, as well as a theoretical quantitative analysis between them, and finally, a quantitative analysis using practical experiment.

## II. FIXED-WINGS VS ROTATIVE WINGS

As there is no standardization of nomenclature for unmanned aerial vehicles, the term rotative-wing will be used in this work to refer to four-propeller (quadcopters) and fixed-wing for fixed-wing aircrafts similar to small aircrafts, as shown in Figure 1. For a general reference to unmanned aerial vehicles, the acronym UAV will be used.

In the studies involving precision agriculture with UAVs, the great majority makes use of fixed-wings in monitoring the large areas of land present in these scenarios, such as [7], [8], [9] and [10].

Fixed-wings stand out from the rotative-wings for use in precision farming because they have greater autonomy, this being the time that a UAV remains in flight without the need to recharge the battery. Because fixed-wings are usually larger, they can carry more powerful batteries, as well as having only one or two propellers, which helps to increase their autonomy.

Another feature that makes fixed-wings still very much applied in precision agriculture is that most of the time the data collected by them are images, in particular, to identify some type of disease in the monitored culture, and a macro analysis of the images is performed. With this, fixed-wing flights can happen faster and at higher altitudes, producing less detailed images.

However, even with less autonomy, rotative-wings stand out in other important aspects, such as:

- 1) **Precision in reading the data:** As the rotative-wings have a stability system formed by four propellers, which allows them to fly at low speed, a detailed reading of the information coming from the sensors or the cameras in them is possible. Another advantage over fixed-wings is in the dynamism of flight altitude. Rotative-wings can fly at lower altitudes for detailed reading of the data, or, if necessary, climb with high vertical velocity to larger heights, for a wider view of the area;
- 2) **Ease of maneuver:** Sharp maneuvers can be performed with rotative-wings, unlike fixed-wings, which usually make great turns to change direction. The steep incline of a fixed-wings during a curve is also a disadvantage, since at this moment, it can not get pictures or read with its sensors, because the data would be distorted. Rotative-wings often fly without the need for sharp slopes, allowing for more accurate readings all the time;
- 3) **Does not require human interaction in takeoff and landing:** As takeoffs and landings happen vertically, they can occur without human interaction, requiring only a flat base so that these procedures can be performed. In the fixed-wings, the nature of such operations is horizontal, requiring a human operator to launch them on the flight and rescue them after landing;

TABLE I. COMPARISON BETWEEN THE FEATURES OF A DJI PHANTOM 4 AND AEROVIRONMENT QUANTIX.

	DJI Phantom 4	AeroVironment Quantix	Comparison Rotative x Fixed Wings (%)
Velocity Horizontal	20m/s (80km/h)	15m/s (57km/h)	25% faster
Autonomy	25 to 30 minutes	45 to 55 minutes	45% less
Time of recharge	60 minutes	75 minutes	20% faster
Value	\$1200	\$ 12000	90% cheaper

- 4) **Cost:** Another advantage of rotative-wings is its cost that is often lower than fixed-wings. In some cases, at the cost of a single fixed-wing it is possible to purchase a set of rotative-wings.

As características citadas anteriormente são comparadas na Tabela I, sendo escolhido um modelo de Rotative Wings, o Phantom 4 da fabricante DJI, e um modelo de Fixed Wings, o Quantix da empresa AeroVironment. Para a comparação, os dados da Tabela foram coletados dos *DataSheets* dos equipamentos. The characteristics cited above are compared in Table I, with a rotative-wing model, the Phantom 4 from the DJI manufacturer, and a model of fixed-wings, the Quantix of the company AeroVironment, chosen. For the comparison, the data in the Table were collected from the DataSheets of the equipment.

Within its segments, DJI Phantom 4 and Quantix of AeroVironment are the most modern models of the largest manufacturers of their categories, as shown in the graph of Figure 2.

The graph presents a study developed by Drone Industry Insight [11]. For the research, the company used parameters such as number of references in web search sites, number of new models of UAVs launched in the last 24 months, number of company employees, among other parameters. The survey resulted in a score, with the largest manufacturers being highlighted. The first four companies (DJI, Parrot, Xiaomi and Hover) are manufacturers of rotative-wings. DJI that manufactures the Phantom 4 is the leader in its segment. The same is true for AeroVironment, the maker of Quantix, which is the fifth-largest manufacturer, but leader in the fixed-wings segment.

As noted in the table, it is possible to verify the superiority of the fixed-wings over the rotative-wings in relation to their autonomy. However, in terms of speed, both can be considered as equivalent, however, the biggest advantage of rotative-wings is its cost, about ten times lower than the cost of fixed-wings.

In the next chapter we will present a quantitative comparison between rotative-wings and fixed-wings.

## III. RESULTS

In order to compare performance between Rotative-Wing and Fixed-Wing, two distinct scenarios will be used: the first considering linear flights and the second considering flights over two-dimensional areas. For both tests, the comparisons will be theoretical based on the technical information of both vehicles. At the end, a practical flight comparison of a Rotative-Wing and a Fixed-Wing is performed to see how they behave in practice.

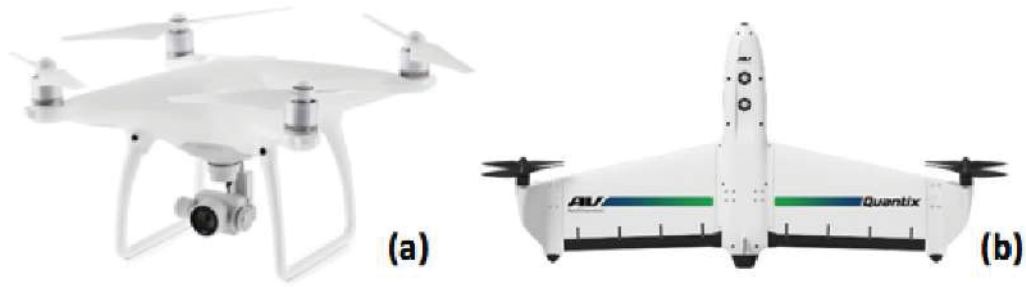


Fig. 1. Model of a rotative-wing DJI Phantom 4 (a) and a fixed-wing AeroVironment Quantix (b).

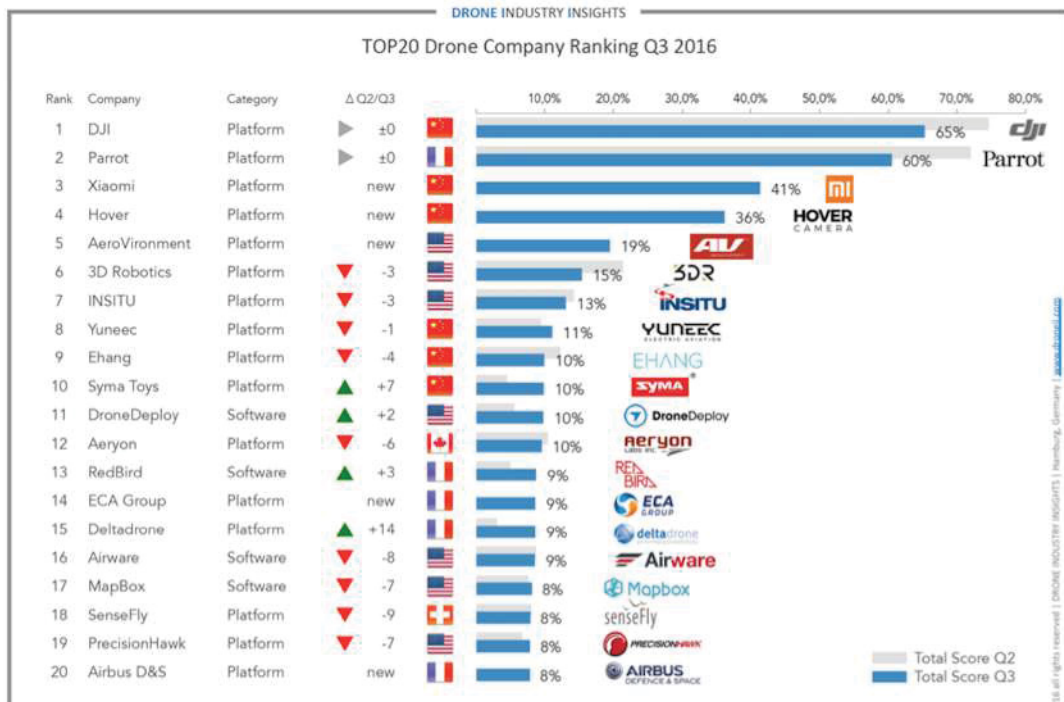


Fig. 2. Largest world manufacturers of UAVs.

Initially, calculations are performed to compare both vehicles in a linear flight scenario, that is, following a straight line. Considering that the Quantix travels at a speed of 15 m/s, with an autonomy of 45 minutes, it travels a linear distance of approximately 40.5 Km. The Phantom 4, considering a velocity of 20m/s, with an autonomy of 25 minutes, can travel 30.0 Km.

In this way, it is concluded that for linear flights, such as road monitoring, border monitoring, among others, fixed-wings performs better, traveling a distance 25 % higher. In this scenario, the disadvantage of fixed-wings is in its cost, which for the models compared is 10 times higher.

However, in precision agriculture, the monitoring takes place in two-dimensional areas, with a high cost of maneuvering for the adopted UAV.

To compare the performance of a rotative-wing and a fixed-wing in a two-dimensional area, a scenario for comparison in

the arbitrary size of 170m x 170m will be used, with a total area of 28,900 square meters, or approximately 2.9 ha(hectare). Considering that this area should be flown with a reading accuracy of 10 meters, the readings should be obtained at points 10 meters apart.

For this area, the rotative-wing would cover it as shown by Figure 3, since it would need to travel 170 meters for 17 times, totaling 2,890 linear meters (blue lines), in addition, there would be the cost for displacement between one line and another, that are 17 movements of 10 meters each (red lines), totaling 170 meters of displacement, making a total of 3,060 linear meters to fly over the area. In this calculation the costs of acceleration and deceleration of rotative-wings were neglected.

For a fixed-wing it is more difficult to present an exact perspective, since it has a very high cost of maneuver and



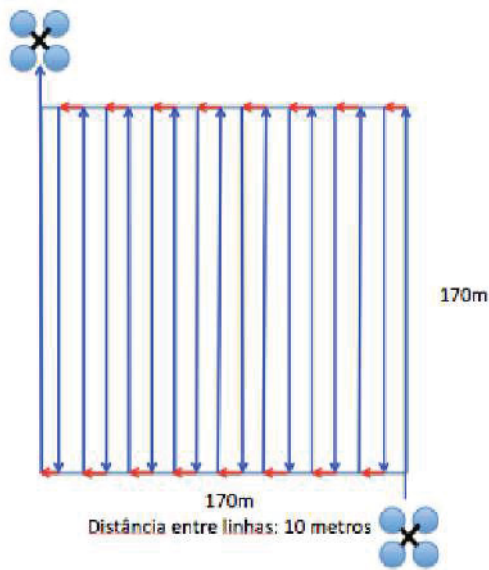


Fig. 3. Rotative-wings course for monitoring a two-dimensional area.

this cost depends on the model of the fixed-wing, but for comparison, it is considered that the fixed-wing travels the same area of 170m x 170m, with readings every 10 meters, we have, thus, a course similar to that presented in Figure 4. The spiral flight is the most efficient for a fixed-wing to cover a two-dimensional area, since it does not have to make a big curve outside the area to be explored to return and perform a flight parallel to the line that has just been flown.

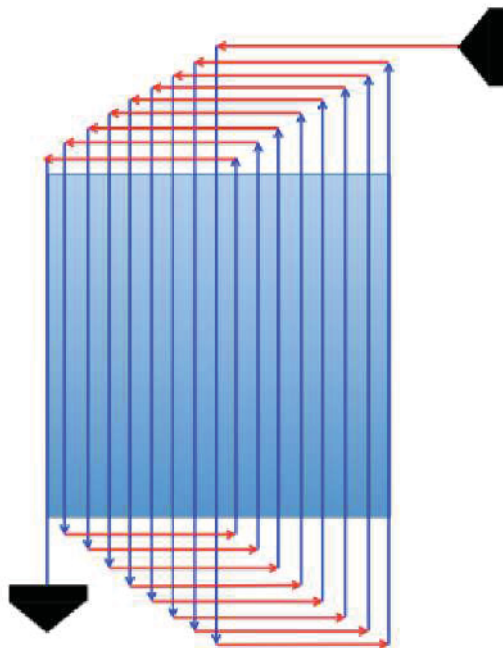


Fig. 4. Path of a fixed-wing for monitoring a two-dimensional area.

It is observed that the cost of line change is great, since the

fixed-wing can not make sudden movements, nor even decrease its speed of navigation to not risk falling.

Thus, to go through the 170 meters x 17 (blue lines), amounting 2,890 meters, a greater displacement is needed to change lines. There are 9 routes of 90 meters to the left and 8 routes of 80 meters to the right (red lines), which results in 1,450 meters of flights only for these exchanges.

For this scenario, while the rotative-wing would only use 170 meters for the displacement between one line and another, the Fixed Wing would use 1,450 meters for this type of displacement. This extra displacement does not result in monitoring and, in practice, it is known that the path for the inter-row displacement is even greater in fixed-wings in real situations, because for the calculations presented in the previous paragraph, it was considered that fixed-wings did not leave the perimeter of the area to be monitored, making 90 degree turns. This cost was not treated in this calculation because it depends on the RPAS used, the larger the fixed-wings, the higher the maneuver cost. The acceleration and deceleration costs were also disregarded.

[12] states in his work that when using fixed-wings, paths with few curves are more efficient considering the distance traveled, duration of flight and energy expenditure, and for this reason the use of fixed-wings is not interesting for overflight of small two-dimensional terrain.

The following is the calculation of the area that can be flown by a single Phantom 4, considering its autonomy.

Considering a reading accuracy of 10 meters, precision used in the previous examples, the Phantom 4 could reach 3,000 points, since it has a flight autonomy of 30,000m linear, and the readings will occur every 10 meters. Calculating the square root of 3,000 (number of points), one has a square area of approximately 54x54 points, that is, 540x540 meters, which represents 291,600 square meters, or approximately 29 ha. Thus, this entire area can be flown over with a single Phantom 4 without the need to recharge the battery. For this estimate the cost of acceleration and deceleration was also disregarded.

With a fixed-wing, if we compare the theoretical dimension that it can travel with a battery charge, the area would be 520x520, totaling since, with precision of 10 meters we would have  $520 \times 52 = 27040$ , however, there is a high cost of displacement between lines. It would be 52 flights of 260 to move the lines, which makes a total of 13520 meters, totaling almost the 40500 linear meters that the fixed-wing can fly with a single battery charge. This cost is theoretical, once we know that the curves of an RPAS must be executed outside the perimeter of the area to be monitored, but even so, with this calculation it is possible to verify that, theoretically, even with a greater autonomy, a Fixed Wing can not monitor a two-dimensional area with the same efficiency as a Drone.

For comparison purposes, all tests carried out earlier are presented in Table II.

To obtain more precision in the presented data, two real tests were also performed using a rotative-wing of the Phantom 4 model of the company DJI, the one presented at the beginning of the section, and a fixed-wing Arator 5a of the Brazilian company XMobots. For technical reasons, it was not possible

TABLE II. THE QUANTITATIVE COMPARISON OF THEORETICAL DISTANCE OF A FIXED-WING AND A ROTATIVE-WING.

	DJI Phantom 4	Aero Vironment Quantix	Comparison Fixed wing x Quadcopter(%)
Linear Distance Overflown with a Battery Charge	30000 m	40500 m	25,92% lower
Traveled Distance to Monitoring 170mx170m area	3060 m	4340 m	41,83% higher
Two-Dimensional area Overflown with a Battery Charge	291600 m <sup>2</sup>	270400 m <sup>2</sup>	8% higher

to test flights with Quantix from AeroVironment, but XMrobot, according to its Datasheet, has its technical characteristics, such as speed and autonomy, very similar to Quantix.

Both tests were performed in favorable weather conditions, with no wind.

The DroneDeploy online tool was used to make the Phantom 4 flight plan. With this, a flight was carried out equivalent to the theoretical tests presented previously, that is, by having the Drone collect information, for example, pictures were used, with precision of 10 meters.

While performing the flight, it was observed that the Drone's battery reached critical level after 15 minutes and 49 seconds, requiring Drone landing. Re-sizing the flight plan, on the second attempt it flew over in 15:50 seconds, flying an area of 37,403 square meters (37ha), reaching a maximum speed of 54km/h (34 miles per hour). The result was the monitoring of a square area of 193m x 193m, as picture taken from the DroneDeploy - Figure 5.

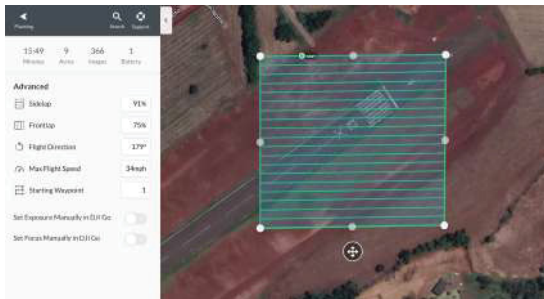


Fig. 5. Real Flight Scenario of a Phantom 4.

The linear distance traveled was  $193\text{m} \times 20 = 3860\text{m}$ , plus the costs of changing the line, 190m, which totals 4,050 linear meters.

Also the same test was done with a fixed-wing of the Arator 5b model of the Brazilian company XMrobot, which has technical characteristics very close to the fixed-wing of the AeroVironment Quantix model used in the comparisons at the beginning of this section.

This Fixed Wing managed to cover an area of 180 x 200 meters, making a total of 36,000 square meters with a battery charge. As it requires a more complex landing process, depending on wind conditions and terrain suitable for its landing, it overflowed the area until its battery reached a level of 15 %, after the monitoring flight was closed to start the landing process.

TABLE III. QUANTITATIVE COMPARISON OF PRACTICAL DISTANCES OF A RPAS AND A DRONE.

	DJI Phantom 4	Aero Vironment Quantix	Comparison Fixed wing x Quadcopter(%)
Linear Distance Overflown	4050	4810	15,8% lower
Area Overflown	37403	36000	3,9% higher
Fligh Time	15m49s	19m07	17,26% lower

Sobre a distância linear percorrida pelo Fixed Wing na cobertura da área citada anteriormente, esta foi de 4810 metros, sendo 3600 metros de área útil e 1210 metros para o deslocamento entre as linhas.

Regarding flight autonomy, the fixed-wing reached the critical battery level at 19 minutes and 07 seconds of flight, and required landing after the low battery warning.

The flight plan of Arator 5b is shown in Figure 6. This screen was obtained from the Mission Planner software, which is used for the planning and flight of the fixed-wing Arator 5b.



Fig. 6. Real flight scenario of an Arator 5b.

### III.

A summary of the data from the two tests is given in Table III.

In general, it was possible to observe the superiority of the rotative-wing for monitoring flights of two-dimensional areas, which commonly happens in precision agriculture.

Even with greater autonomy, fixed-wings have a very high yaw cost, which compromises their use in the monitoring of agricultural areas, due to the great need to change direction.

## IV. CONCLUSION

The objective of the present work is to provide a theoretical and practical comparison of the use of two types of unmanned aerial vehicles: rotative-wing vehicles (quadricopters) and fixed-wing vehicles (similar to small airplanes) to identify in which scenarios the use of each type is justified.

For the tests, two vehicles were used that stand out in their segments: the Phantom 4 being the rotative-wing and the Quantix being the fixed-wing.

The comparisons took place in three moments: first a qualitative evaluation given the information obtained from the datasheets of the two vehicles, in a second moment a theoretical evaluation, considering two scenarios (linear flights

and monitoring flights of two-dimensional areas) and, finally, using the vehicles on real flights.

It was observed that fixed-wing vehicles stood out in scenarios of linear flights, for example, in the monitoring of borders or routes, since that in these scenarios there isn't a great change of directions. As fixed-wings have a longer range, and their speed is equivalent to the speed of a rotative-wing, this causes their superiority to be highlighted (25.92 % greater).

However, in flights for monitoring two-dimensional areas, which are commonly used in precision agriculture, the superiority is held by the fixed-wing vehicles. Although they have a 45 % lower autonomy compared to the autonomy of fixed-wing vehicles, the rotative-wings have a small cost for the change of direction, as well as maneuvers, which makes it possible to cross an area 8 % greater with a single battery charge.

Finally, tests with actual vehicles were performed, and similarly, rotative-wing vehicles achieved higher monitoring (3.9 % greater).

Thus, it can be concluded that, even though they have a lower autonomy compared to fixed-wing vehicles, the advantages of a rotative-wing applied to precision agriculture are many, among them: they cost much less than a fixed-wing, cover practically the same area in a shorter time given its low maneuverability, allows for more dynamic flights, allowing the rotative-wing to move up or down rapidly, searching for more detailed or larger images of the area to be monitored, and especially, it can perform autonomous flights without the need for human interaction for either take-off, landing or battery recharging, as presented by authors, such as [13], [14] and [15].

As a guide for this study, a graphical analysis will be carried out to identify the scenarios in which it is no longer interesting to use a rotative-wing vehicle and it is again more attractive to use the fixed-wing vehicle, imagining that at regular shapes (square or circle, for example) the superiority is of the rotative-wings, but for linear flights (or more elongated areas, such as rectangles or ellipses), the superiority returns to the fixed-wing, so a quantitative study is necessary to identify at which point this division occurs.

## REFERENCES

- [1] Gabriele Bastianelli, Diego Salamon, Alessandro Schisano, and Alessandra Iacobacci. Agent-based simulation of collaborative unmanned satellite vehicles. In *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*. Institute of Electrical & Electronics Engineers (IEEE), oct 2012.
- [2] Eobin Alex George, Gaurav Tiwari, R N Yadav, Edward Peters, and Srishiti Sadana. UAV systems for parameter identification in agriculture. In *2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS)*. Institute of Electrical & Electronics Engineers (IEEE), aug 2013.
- [3] Robison Cris Brito, Jose Felipe Loureiro, Eduardo Todt, and Roberto Pereira. A systematic mapping for the scenario of non-urban autonomous vehicle cooperation systems. In *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*. IEEE, nov 2017.
- [4] Min-Fan Ricky Lee, Shun-Feng Su, Jie-Wei Eric Yeah, Husan-Ming Huang, and Jonathan Chen. Autonomous landing system for aerial mobile robot cooperation. In *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*. IEEE, dec 2014.
- [5] Parikshit Maini and P. B. Sujit. On cooperation between a fuel constrained UAV and a refueling UGV for large scale mapping applications. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, jun 2015.
- [6] Bing Yan and Chengfu Wu. Research on taxi modeling and taking-off control for UAV. In *2014 Seventh International Symposium on Computational Intelligence and Design*. IEEE, dec 2014.
- [7] Jose Polo, Gemma Hornero, Coen Duijneveld, Alberto García, and Oscar Casas. Design of a low-cost wireless sensor network with UAV mobile node for agricultural applications. *Computers and Electronics in Agriculture*, 119:19–32, nov 2015.
- [8] Chunhua Zhang, Dan Walters, and John M. Kovacs. Applications of low altitude remote sensing in agriculture upon farmers requests— a case study in northeastern ontario, canada. *PLoS ONE*, 9(11):e112894, nov 2014.
- [9] Dionisio Doering, Arthur Benenmann, Rafael Lerm, Edison Pignaton de Freitas, Ivan Muller, Jean M. Winter, and Carlos Eduardo Pereira. Design and optimization of a heterogeneous platform for multiple UAV use in precision agriculture applications. *IFAC Proceedings Volumes*, 47(3):12272–12277, 2014.
- [10] Pablo Menendez-Aponte, Christian Garcia, Douglas Freese, Sinem Defterli, and Yunjun Xu. Software and hardware architectures in cooperative aerial and ground robots for agricultural disease detection. In *2016 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, oct 2016.
- [11] Drone Industry Insights. Top20 drone company ranking q3 2016. In *Requisitos Gerais para Aeronaves não tripuladas de uso civil*. Drone Industry Insights., dez 2016.
- [12] Yan Li, Hai Chen, Meng Joo Er, and Xinmin Wang. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. *Mechatronics*, 21(5):876–885, aug 2011.
- [13] HUI Cheng, YOUSHENG Chen, XIAOKUN Li, and SHING Wong Wing. Autonomous takeoff, tracking and landing of a uav on a moving ugv using onboard monocular vision. In *32nd Chinese Control Conference*. Institute of Electrical and Electronics Engineers (IEEE), aug 2013.
- [14] Federico Francesco Barresi and Walter Allasia. Airport markings recognition for automatic taxiing. In *Conference: Design and Architectures for Signal and Image Processing (DASIP)*. Institute of Electrical and Electronics Engineers (IEEE), aug 2013.
- [15] Francesco Cocchioni, Valerio Pierfelice, Alessandro Benini, Adriano Mancini, Emanuele Frontoni, Primo Zingaretti, Gianluca Ippoliti, and Sauro Longhi. Unmanned ground and aerial vehicles in extended range indoor and outdoor missions. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, may 2014.

# Optimized division of exploration areas in multi-robot systems considering static and dynamic charging stations

Robison Cris Brito  
Federal University  
of Technology  
UTFPR  
robison@utfpr.edu.br

Nicollas Saque  
Federal University  
of Paraná  
UFPR  
nisaque@live.com

Diego Addan Gonçalves  
Federal University  
of Paraná  
UFPR  
diegoaddan@gmail.com

Fabio Favarim  
Federal University  
of Technology  
UTFPR  
favarim@utfpr.edu.br

Eduardo Todt  
Federal University  
of Paraná  
UFPR  
todt@inf.ufpr.br

**Abstract**—The present work shows a Java Desktop graphical application that makes use of the JST library that requests data to the user, such as the map of the place to be explored, quantity of charging stations, if these stations must be in static places, perhaps due to electrical outlet locations restriction, or if they could be dynamically placed. Based on these data an optimal solution to insert the charging stations as well as the definition of each robot activity area are presented. Algorithms such as Voronoi, Viktor Grabarchuk and Centroid position are used in this process. The Voronoi Algorithm allowed the balanced division of the action area into a group of robots considering static recharge positions. The combination of the Viktor Grabarchuk and Centroid Position algorithms allowed a balanced division of the operating area for different robots, and also, the definition of a central position to allocate the bases of recharges, which reduces the time of displacement of the robot to the base when necessary.

**Keywords**—charging station, Voronoi, Viktor Grabarchuk

## I. INTRODUCTION

With the industrial robot advent in 1954 [1] several human being tasks became easier. An example is the automobile industry, where currently much of the work is done exclusively by machines.

In the Hannover Fair 2011 edition has the 4.0 industry concept also called fourth industrial revolution was popularized as it was exposed to the general public. This has led to a growing concern about intelligent devices interconnected within the industrial production and logistics [2]. In this context, the use of intelligent and autonomous robots stands out, making them work cooperatively to achieve a common goal through a software specification, without the need for direct human intervention in their work.

Besides industrial robots, there are other types of robots such as domestic, military, space, entertainment, competition and especially vehicles that can move on their own [3] in terrestrial, aerial and aquatic environment. As examples of terrestrial vehicles we have domestic cleaning robots, grass mowing robots and autonomous combined harvesters. In the aerial robots category stand out the UAVs (Unmanned Aerial Vehicles) and Drones. Among the watercraft we have submarines in the oil industry applied to pipelines inspection and supervision [4].

In this work we will use the term robot to refer to any vehicle that can move autonomously, this being a terrestrial vehicle, aquatic, and especially aerial vehicles, like a drone.

Often a task is complex to be performed only by one robot, requiring two or more of them, working together, to achieve a common goal [5]. This states that a group of robots manages to guarantee the coverage of all the places with greater robustness in the patrolling.

In addition, overloading work on a single robot can cause equipment damage and activity failures, so it is interesting to use a multi-robot system. In addition to share responsibilities over each one, when there is a failure in one of the robots, the others manage to control the situation and continue the work [5].

Precision farming can be greatly benefited by the use of numerous aerial vehicles for the monitoring large production area. In the domestic environment, there could also be glass-cleaning robots in skyscrapers, or even garbage-picking robots in a city or park.

When working with small robots, the major limitation lies in the autonomy of their batteries. The quadricopters, for example, have an average autonomy of 10 minutes. Autonomous charging stations are indispensable for this type of vehicle when they are covering a large area.

In this context there is a concern about the best places to locate the charging stations for the robots, as well as to define each robot area of action. In some cases the location of the charging station must be informed by the user, as if it should be static, since there may be a power point or some other limitation that makes it difficult to change the location.

In other cases, the charging stations can be dynamic, being necessary to establish an algorithm to find the best position for the charging stations in order to optimize the robots work, considering time of displacement and battery usage. In outdoor environment situations where drones are used, charging stations can be placed almost anywhere, if the stations have solar power plates. At other times, drone charging is performed by land vehicles with loaded batteries, so there is the also freedom to position the charging stations arbitrarily, considering terrain limitations.

## II. RELATED WORKS

There are studies that, in order to optimize the work of a multi-robot system, divides the operational area. Virtual Door algorithm [6] aims to divide an indoor area into parts delimited by the rooms of that region, in which each environment is a sub-region. That algorithm is limited to closed environment since it was designed to be used by autonomous vacuum cleaners. For the algorithm application, it is necessary to pre-set the rooms number in the environment.

Another strategy proposes a division of the area into polygons from a checkered mesh overlapping the operational area, where the edges of the subdivisions match those of the meshes. As the work presented by Lee [6], it also considers only indoor areas, since it will be used by autonomous vacuum cleaners. Thus, there may be more rooms than robots, demanding a control for the sub-areas distribution.

Hert and Lumelsky [7] proposes the division of areas according to the number of robots in the system. For this, is considered other variables such as the initial position of each robot. To solve the problem, the author proposes two algorithms, the first for convex polygons, and the second for non-convex polygons.

The position of the charging stations can be used, optimizing its location based on the proposed divisions [8]. The authors present an approach in which it is considered only the base of mobile charging stations, that can be best allocated according to the robots work division.

This work proposes a solution for the division of areas based on the main characteristics presented in the previous four works: Enable the algorithm application in places where the charging stations for both cases, fixed or mobile. Also, allow a division of the area regardless of the number of robots allocated to work on them. Was used, in general, concepts that have provided simplicity in the solution proposed in [6] and [9], as well as the concepts for convex and non-convex polygons as proposed in [7], also considering the dynamic charging stations, as presented in [8].

## III. METODOLOGY

Aiming to optimize the work of autonomous robots and considering the existence of charging stations, it is very important to choose a good policy to define the area of operation of each robot, as well as identify, when possible, the best location for the charging station. This decrease considerably the energy use and the displacement time for the exploration of a large area.

The first parameter required for division of charging station is the available number of them. In the present work it was considered the existence of a charging stations for each robot in the system.

Afterwards, it is necessary to decide whether the system will use static or dynamic charging stations.

### A. Static Charging Stations

When there is a limitation concerning the positioning of the charging station, we have to consider a strategy in which the

station placement is static, and thus just the area of operation of the robots should be optimized.

In this situation, algorithms that divide the area based on fixed points, the charging stations, must be used to identify the best area for each robot. The algorithms based on the Voronoi diagram stand out in this type of case. Among the many of them, the Fortune algorithm was chosen.

In 1987, Steve Fortune proposed a new algorithm for constructing the Voronoi diagram, called Scanning Algorithm, it has a complexity of  $O(n \log n)$ , considered optimal [10].

The algorithm is based on the passage of a scan line in the plane containing the fixed points. First, when the line encounters a point, the algorithm creates a zero-width parabola with vertex at that point. With the scan line displacement, this width is increased and new parabolas are created by passing through other points. From the intersection of all these parabolas and some other considerations, the edges of the Voronoi polygons are constructed.

An important consideration is that for the use of the Voronoi algorithm, it is necessary to have fixed initial points, which are the places where the stations are found. Based on these, spaces of action are created for each robot.

### B. Dynamic Charging Stations

At certain cases, especially in outdoor environments, it is of great importance to have dynamic placement of the charging stations, in special, when we are using autonomous drone. The choice of location should be made in a way that optimizes the use of robot battery. For the proposed model, a space partition will be used for each robot considering each charging station.

The space partition, in principle, is important to restrict the field of action for each robot, facilitating the accomplishment of the assigned task. The partition of the space to be explored allow the robot to traverse the smallest possible area, reduces the possibility of going through the same place several times and the occurrence of overlapping. The objective is saving energy and time for complete accomplishment of the work.

A model that stands out for the space partition into equal parts is the decomposition from the polygon division algorithm of Viktor Grabarchuk. Based on an integer 'n', which represents the number of partition, the algorithm divides a polygon identified by the coordinates of its vertices. The cuts in the geometric figure are made with straight lines and the algorithm accepts convex and concave polygons. After division, all space partition have the same area measure [11].

After dividing the space that delimits where each robot will work, it is necessary to define the position of the charging stations within each space partition. Considering a simplifield premise that the best point would be the central position of each division, which optimizes the work of moving the robot within the space, an algorithm was used to calculate the centroid of the figure. Thus the distance from any point to the center will be approximately the same for any point chosen within the subarea.

The chosen algorithm to calculate the centroid is one present in the JST library, which has many functions for handling 2D geometric figures.

#### IV. RESULTS

In order to build of an intelligent system that identifies the best operation space in a multi-robot environment, considering that the charging station is fixed or dynamic, a software was developed in the Java programming language using Canvas [12] for plotting the data on the screen. Java language was chosen for the flexibility that it presents, used in multiple platforms, in which the same generated application (file with JAR extension) can be executed in Windows, Linux and MAC OS platforms, and can be executed on web pages, like an Applet.

For 2D geometric data manipulation, the Java Topology Suite (JTS) library was used. JTS is an open-source JAVA library used to treat geometric elements for spatial analysis, advanced query and construction of 2D spatial applications. The library provides a set of functions that facilitate the processing of geometric information.

Considering the use of the this system, first the user needs to define which will be the operation space of the robots/drones. The developed system could be use for indoor/outdoor/hybrid environment. It is possible through some questions asked by the system at the beginning of the user's use. .

Concerning the system use, as first step, a mapping of the robots action area must be done. The environment developed works for both indoor, outdoor, or hybrid environments, and to ensure this flexibility, generic information is requested and used in the logic process in order to define the area of operation for each robot.

The first requested information is the background environment image, which can be a blueprint or a map downloaded from some map application. The application accepts JPEG and PNG images.

In the sequence, the user is asked which base unit of length to use. This can be in meters, mainly used for indoor environments, or geocodes that contains longitude and latitude, used especially in outdoors environments.

Afterwards, a form is presented with the selected image referring to the area of the robot. This image can be the a site blueprint (for indoor robots) or a map snapshot of some map site (for outdoor robots). For the second case is performed a calibration process, identifying the robot operation area extremities associating them to a point using the base unit. For example, in indoor areas can be associated the unit of measurement in meters on the x and y axes, whereas for outdoor areas, can be associated with longitude and latitude coordinates.

This conversion becomes necessary, since the computational system in Canvas consider the points of the screen in pixel, using the nomenclature position in pixel for x and y axis, being x corresponding a one point to the width of the screen, starting at 0 left, and y corresponding a one point to the height of the screen, starting at 0 top of the screen.

Once this computational result reflects in a real environment, it is indexed according to the type of environment, using meters (commonly used for indoor environments) and

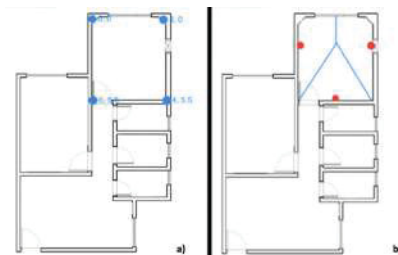


Fig. 1. a) Definition of the robots action area perimeter. b) Example of usage of application in order to define robot operating subareas for static charging station

geocodes (frequently used in outdoor environments), easing the interpretation by the user.

In order to calibrate the measures, the user must click on all vertices of the robot operation area, informing the reference value for these points. For indoor environments 0 and 0 (x and y) can be used in the vertex related to the upper left corner, and the width and length of the environment to the area in the lower right corner. For outdoor environments, latitude/longitude information can be used from each point - Figura 1.a).

After the calibration of the data related to the action area vertices, the user is asked the number of charging stations that will be used in the processing of the Voronoi algorithm (static charging station) or Viktor Grabarchuk (dynamic charge).

Finally, the system asks the user if it will use static or dynamic charging stations in the environment.

##### A. Application for Static Charging Stations

When static charging stations are used, the user informs the system where these points will be in relation to blueprint or map. Indoor environments

The user can inform a charging station location on the map by clicking on the point where it will be inserted, or informing this via inbox, indexing this station in meters from the upper left corner of the image or geocode.

Then, the result of the best robot action space considering the static position of the stations is presented in Figure 1.b).

As can be seen in this example, the space will be explored by three robots, and have three static charging station (circle filled). By means of the Voronoi algorithm, the action area of each robot is established. This information can be exported.

This is not a deterministic algorithm, showing to the user in this experiment only the first solution found.

##### B. Application for Dynamic Charging Stations

In environments where the charging station location can be defined by the software, the user does not need to inform the station location.

In many cases, this type of approach is used for outdoor environments, where a map is loaded into the application by an online map system (such as Google Maps). The greatest challenge when using maps is in the referencing activity for

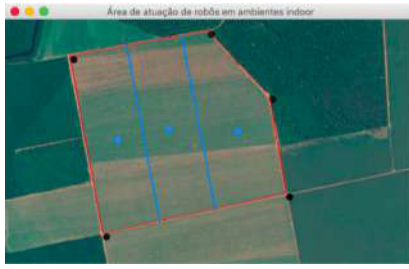


Fig. 2. Example of application using for outdoor environments.

all the vertices of the aimed area to be explored. The geocode is used in order to refer the vertices, obtained in some online map application, or even, in loco, using a GPS device. In some situations, even some of the edges of this area are circular, and in this case, as the system was developed to treat only vertices, the user needs to use an interpolation, placing points at a smaller distance from each other until the edge be drawn.

However, it is in this scenario that the greatest possibilities are found, especially when autonomous vehicles are used for surveillance/search and rescue. It is possible for these vehicles to return for the base and recharge their batteries autonomously. The same can be pointed about unmanned aerial or aquatic vehicles, where there are minor obstacles to movement, thus there is a great potential for exploitation of this type of service, mainly for precision farming and search/rescue.

For all these vehicles, the battery is still the major limitation, increasing the need to have scattered charging stations for an autonomous recharging. Algorithms or tools to identify the best places of these charging stations are of great importance for the collective optimization of the vehicle batteries use.

The developed application, in the dynamic charging stations environment, first defines the areas of action subdivision for the vehicles, using the Viktor Grabarchuk algorithm. After defining a subarea division of the exploration area, associated to each charging station, the JTS algorithm is executed for the sub-area centroid calculation, allowing an optimization in the vehicle movement on this subarea, considering that it can return several times to recharge the battery, given its autonomy or the area size - Figure 2.

In the Figure 2, it is possible to observe a mapped area of five vertices. This area was divided considering five charging stations (filled circles).

## V. CONCLUSION

The objective of this work aims to explore the optimized division of an exploration and operation area by robots in sub-areas, considering that the robots used are autonomous. Besides the exploration, they have the characteristics of returning to a charging stations for the recharge of their battery autonomously. As the great limitation of these robots is usually their autonomy, it is important to optimize their action areas, as well as, when possible, identify the best place to insert the station.

Two approaches were studied for this work: the first, where the charging station is static. In this case, the position of the

station is fixed, being a restriction that can not be changed, so what is optimized is the area of action for each robot, subdividing the total area in sub-areas of the same size. This is an approach commonly used by indoor robots, such as vacuum cleaners or internal monitoring vehicles, since in these places the charging stations are close to the electrical outlet. The second approach refers to the situations in which the charging stations can assume any place, used especially in outdoor environments, since these bases usually present solar panels and do not depends on a source of wired electric power. In this cases, besides sub-dividing the exploration area of the same size, the best position is also pointed for the best position for the charging stations, often being the mass center of the sub-area.

For the static charging station scenario, the Voronoi algorithm was used, in which static points are established (places where the charging station is located), and from these, the total area is divided into subareas.

In dynamic vehicle scenario, initially the exploration area is sub-divided, one for each charging station. The algorithm of Viktor Grabarchuk was used, identifying the center point of each sub-area in sequence, where the charging station will be placed.

Among the identified problems, there is the choice of algorithms that cover the requirements of the proposed system, since there are many area division algorithms available and different central point definitions.

## REFERENCES

- [1] RIA. Unimate. In <https://www.robotics.org/josephengelberger/unimate.cfm>. Acessado em 21 de maio de 2018., dec 2015.
- [2] SEBRAE. Saiba o que é a indústria 4.0 e descubra as oportunidades que ela gera. In <http://www.sebrae.com.br/sites/PortalSebrae/artigos/saiba-o-que-e-a-industria-40-e-descubra-as-oportunidades-que-ela-gera.11e01bc9c86f8510VgnVCM1000004c00210aRCRD/>. Acessado em 29 de maio de 2018., dez 2016.
- [3] Todd Litman. Autonomous vehicle implementation predictions. In <https://www.vtpi.org/avip.pdf>. Acessado em 21 de maio de 2018., apr 2018.
- [4] Cécile Huet and Franco Mastroddi. Autonomy for underwater robots—a european perspective. In *Autonomous Robots*. Springer US, oct 2016.
- [5] Daniela Filipa Aires Nobre. RoboCops II – Patrulhamento de Edifícios com Equipas de Robôs Móveis. 2016.
- [6] Jeong H. Lee, Jeong S. Choi, Beom H. Lee, and Kong W. Lee. Complete coverage path planning for cleaning task using multiple robots. In *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, oct 2009.
- [7] Susan Hert and Vladimir Lumelsky. Polygon area decomposition for multiple-robot workspace division. *International Journal of Computational Geometry & Applications*, 08(04):437–466, aug 1998.
- [8] Bingxi Li, Barzin Moridian, Anurag Kamal, Sharvil Patankar, and Nina Mahmoudian. Multi-robot mission planning with static energy replenishment. *Journal of Intelligent & Robotic Systems*, jul 2018.
- [9] 2002.
- [10] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry*. Springer-Verlag Berlin Heidelberg, 2008.
- [11] Sumit Khetarpal. Dividing a polygon in any given number of equal areas. In <http://www.khetarpal.org/polygon-splitting/>. Acessado em 24 de maio de 2018., maio 2015.
- [12] Robison Cris Brito and Ricardo Ogliari. Desenvolvendo um jogo para desktop controlado pelo celular via bluetooth com canvas. In *Mobile Magazine ed. 21*. Neoficia, nov 2009.

# Optimization System for Dynamic Flight Planning for Groups of Drones using Cooperation with Mobile Recharge Bases by means of Multiagent System and Recursive Auctions

Robison Cris Brito  
Federal University  
of Technology  
UTFPR  
robison@utfpr.edu.br

Jose Felipe Loureiro  
Federal University  
of Technology  
UTFPR  
zefelippe\_0@hotmail.com

Andre Guedes  
Federal University  
of Paraná  
UFPR  
alpguedes@gmail.com

Eduardo Todt  
Federal University  
of Paraná  
UFPR  
todt@inf.ufpr.br

**Abstract**—This work presents a proposal for a cooperation system aimed to optimize flights of unmanned aerial vehicle like a quadricopter, applied to precision agriculture. The system uses technologies that allow the opening, which is the property of inserting and removing system elements at any time, and dynamicity, allowing the system to recover itself from adverse events or failures. It is also proposed a distributed optimization algorithm, that optimizes the number of points visited by the quadricopter, considering the limitation of it's autonomy. This work starts by presenting the techniques used to define the research problem, such as Problem Solving, Stakeholder Diagram, Evaluation Frame, Value Pie and Building Blocks of Culture. Next, it presents the Systematic Mapping and Systematic Review. These studies allowed to define the research problem, and propose a system to solve it, as to define the technologies used, such as Multiagent Systems, cognitive agents considering mental states, as beliefs, desires, and intentions, the negotiation among agents using FIPA Contract-Net protocol, and optimization using the proposed recursive auction algorithm. Finally, tests were developed to evaluate the proposed Multiagent System and the algorithm used to perform the recursive auctions. The Multiagent System guaranteed the opening of the system in tests with inclusion and exclusion of elements, the cognitive agents considering mental states allows the dynamicity of the system. The optimization using recursive auctions was tested in scenarios with 4, 9 and 16 points, and in all of these the optimal result was found. To minimize the processing time, as the number of message exchanges among the agents, two heuristics were proposed. After applying the heuristics, a reduction of up to 99% was achieved in the number of messages exchanged between agents in complex scenarios, like the one with 16 points.

**Keywords**—Drone, Multiagent System, BDI, Recursive Auction

## I. INTRODUCTION

Drone, Quadricopter, Unmanned Aerial Vehicle (UAV) and Remotely Piloted Aircraft (RPAS) are some of the name variations for remotely controlled unmanned aerial vehicles and / or autonomous. The UAVs, to generalize the various denominations attributed to unmanned aerial vehicles, were originally a hobby, but with the evolution and accessibility of new technologies, what was called aeromodelling, today

is treated with seriousness, since its applications have been growing in the most several areas.

There is a very large demand for UAVs. In the military area, they can be applied in site recognition, surveillance, damage assessment after natural disaster or as a contingency mechanism for communication. In the civil area, other demands are presented, such as environmental monitoring (pollution, climate or scientific applications), fire monitoring, border control, combating drug traffic, aerial surveillance, mapping, traffic monitoring, humanitarian aid, searches and salvage, precision agriculture, among other applications.

The importance of UAVs in precision agriculture is highlighted by authors such as [1] who register that UAVs provide the best platform to evaluate agricultural production, being fast and efficient compared to traditional terrestrial techniques, allowing reading on virtually all types of terrain. UAVs are versatile devices that can be used in a wide variety of agricultural applications, such as automatic monitoring of water and irrigation canals and channels, detection and control of agricultural and livestock pests, in the count of herd animals, for example.

From a technical point of view, currently two types of UAVs are used in precision agriculture: the fixed-wing UAVs, which have the shape of a small airplane, known as Remotely Piloted Aircraft Systems (RPAS), and the rotative wings, like a Drones.

Data from the systematic mapping developed by [2] indicate that 83% of the publications considered by the authors involve fixed-wings. This superiority is believed to be due to the fact that they are the first unmanned aerial vehicles developed, as well as being applied to precision agriculture for many years, and also because of their autonomy, which is superior to the rotative wings.

However, this second has several other advantages, such as greater accuracy in reading the data, since they can fly more slowly without the risk of falling. They have greater ease of maneuver, since unlike the fixed-wings, they can make sharp curves changing direction quickly, they also do not need the



human interaction in the takeoff and landing, since the nature of these operations are vertical. Finally, it stands out as an advantage its cost, since a rotative wings can cost up to ten times less than a fixed-wing UAV.

The following article presents a comparative analysis between these two types of unmanned aerial vehicles: fixed-wings and rotative-wings, to identify, which is indeed the best to be applied in precision agriculture, being that historically, fixed-wings are the most used.

## II. RELATED WORKS

After the definition of the research scenario, a Systematic Mapping was carried out to identify the possible research topics related to flights of groups of UAVs.

Of the 48 papers selected from the systematic mapping, seven of them were the ones that contributed the most to the elaboration of the proposed system. These articles are presented below.

The work of [3] is considered the closest to what is being proposed, since it works with multi-RPAS flight optimization applied to precision agriculture. In addition to being among those selected for a full reading, is the most cited work in the literature. For this reason, the thesis that resulted in this work was also read and analyzed.

[3] works with the cooperation of two RPAS applied in precision agriculture. The author divides the problem into two parts: first, it uses the particle flow principle to make the RPAS attracted by points through which they must pass. It then uses an optimization algorithm developed in Mixed Integer Linear Programming (MILP) to perform the flight optimization of the two UAVs, knowing the waypoints and the place of landing and takeoff. Among the limitations of the work is the fact of being developed for only two UAVs, besides being a static model. Before the flight, knowing the waypoints, a algorithm is executed in MILP that generates the flight plans, that later will be executed by the UAVs. According to the author, the number of two RPAS is ideal for cooperative flights, because the larger the number of UAVs, the greater the complexity and the total cost of the system. Flights with RPAS have a high cost for setup of the aircraft, besides the need for the intervention for takeoff and landing. The number of system operators is proportional to the number of UAVs.

The work of [4] presents a flight model for multiple RPAS. This model was tested only in a simulation environment. Although simulated, issues such as angles and minimum speed are considered so that the RPAS can perform maneuvers.

[4] also defines the points that must be visited by the UAV, but includes the parameter flight height, in addition to introducing the concept of prohibited points, which are points for which the UAV can not pass. The result is a three-dimensional flight plan. This work aims to find a feasible result, in which all the points informed must be visited. Genetic Algorithm (GA) was used to define the flight plan. The system is trained and optimized prior to the aircraft flights. In this model it's not considered the dynamics existing in the real models, such as wind gusts or the early depletion of the battery of some UAVs.

[5] presents in his paper a model to optimize RPAS flights applied to the data collection of wireless sensor networks. To minimize the cost of data transmission over a wireless network, a system is proposed for flight optimization of a RPAS, the sensors being distributed in cluster and sending the information to a node called head of the cluster. This node centralizes all information and forwards it to the RPAS as it flies over that node (head node of the cluster). Battery costs for information exchange within the network are compared to the traditional approach, called Low Energy Adaptive Clustering Hierarchy (LEACH), and the model using fixed-wing UAVs has presented system battery savings overall.

The work of [6] stands out for the organization of dynamic flight environments, without necessarily addressing optimization. This author embarked on an entire infrastructure in a RPAS to behave as a smart agent in a Multiagent System. The author of the work considers that the RPAS are connected to each other all the time and codify the logic of this as being a cognitive agent using for this the Belief, Desire and Intention.

In [6] it is assumed that these RPAS are part of a dynamic and open Multiagent System, allowing new RPAS to be inserted or removed dynamically. In order to guarantee this possibility, the organizational model of agent called Model of Organization for Multiagent System (MOISE+) is used. In order to share information between the UAVs, we use the tool Common ARTifact Infrastructure for AGents Open Environments (Cartago), which acts as an artifact repository. In this repository, the beliefs, desires, and intentions of the Multiagent System are shared.

The work of [7] also made use of Multiagent Systems to optimize group flight of RPAS. The author proposes a simulation system to optimize the flight of RPAS groups. For the system, the author addresses the problem as a Traveling Salesman Problem (TSP), considering the multi-RPAS flight, although it is not taken into account that the RPAS can return to the base to recharge their battery. For the optimization, the author created a hybrid algorithm from the combination of heuristics with stochastic techniques.

[8] present a RPAS group flight optimization model from a hybrid PSO and GA algorithm. This work considers the optimization of RPAS paths in a three-dimensional scenario. As contributions of this work stands out: the processing that is executed in parallel, optimizing the use of multi-core processors and the use of multi-objective function, allowing to optimize more than one parameter of the system. This work was also tested only in a simulation environment.

Finally, [9] presents a system of cooperation between Drones and Unmanned Ground Vehicles (UGV) in order to calculate the best Drone flight route and to identify the ideal number of UGVs to provide support to air vehicles. It is considered that after the flight, the Drone can land on the UGV that moves to another strategic point, while the quadricopter recharges its battery. In the proposed model, a UGV can carry up to two Drones at the same time. However, the system is not online, so the optimization algorithm is executed before the start of the mission, identifying the ideal number of Drones and UGV to complete the mission and defining the movement plans of the involved vehicles. From this point on, the system loses its dynamicity, only what has been previously processed

TABLE I. CHARACTERISTICS OF THE WORKS THAT CONTRIBUTED THE MOST TO THE PROPOSED SYSTEM.

Work	WP	FP	3D	Type UAV	Numb. UAV	Tec. Optim.	Op.	Din.
[3]	S	N	N	F. Wing	2	MILP	N	N
[4]	S	S	S	F. Wing	n	GA	N	N
[5]	S	N	N	F. Wing	1	PSO	N	N
[6]	N	N	N	F. Wing	n	N/A	S	N
[7]	S	N	N	N/A	n	Hybrid	S	N
[8]	S	N	S	F. Wing	n	PSO/GA	S	N
[9]	S	N	S	Drone	n	GA	S	N

is performed.

A summary of the main features of these 7 papers, which most contributed to the proposal of this paper, are presented in Table I.

The table shows the related works in each row, with their characteristics in the columns. The WP column (WayPoint) indicates whether the respective work handles required ticket points, and the FP column (Forbidden Point) indicates if the work handles forbidden points. The 3D column informs whether the work considers the height at the waypoints. In type of UAV it is possible to observe that the great majority of the works makes use of RPAS. The UAV Quantity column shows how many air vehicles are supported by the proposed solution, with "n" indicating that there is no limit. The column Tec. Latin. presents the optimization algorithm used by the works. Openness means that the work supports the inclusion and exclusion of new elements and dynamicity indicates whether the work manages to recover from adverse events that occur during execution.

From the works cited in this section it is possible to observe that few scientific works deal with flight optimization with Drones, although having several advantages over RPAS, as presented. It is also possible to verify that no article studied deals with the dynamicity of the environment, a very common problem faced in real flight environments. It is believed that this is because most of the works treat previous optimization and not optimization during the execution of the system.

### III. PROPOSED SYSTEM

Although several preliminary studies have been developed and showed the viability of a system that deals with the execution of monitoring missions formed by several Drones, a research topic is still little explored: flight optimization considering the opening of the system and dynamicity of the environment.

In this way, the proposed system makes a hybrid of the characteristics presented in the 7 works detailed in the previous section, however, with a focus on the distributed optimization. An overview of the system is presented in Figure 1.

As shown in Figure, the system proposes the use of multiple Drones working cooperatively, applied to precision farming to perform missions that escapes the Drone's individual capabilities. Precision farming was chosen because of its characteristics, since the monitored areas are two-dimensional. And in these scenarios it is important that flights are slow for a detailed reading of the terrain.

It is proposed the cooperation with land vehicles to recharge the battery of air vehicles, as well as an event-based

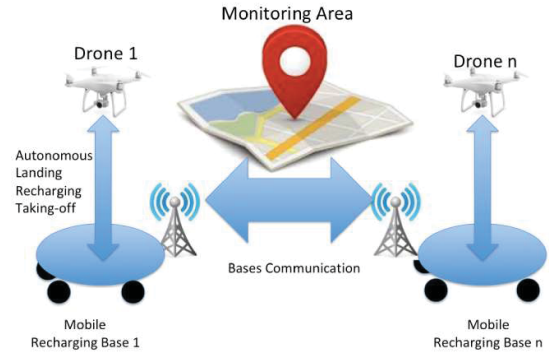


Fig. 1. Overview of a flight optimization system for multiple Drones.

communication model. To optimize Drone battery usage, the biggest limitation of the system, communication only happens between the charging bases. The Drone battery is only used for the flight. When the Drone returns to the base it feeds the system with the information retrieved during the flight, and the system defines the new flight plans based on this information for all other Drones. As the Drone recharges its battery, the land vehicle moves to a new strategic position for the Drone to begin the new mission.

To guarantee the opening of the system, it was chosen to model it using the Multiagent System architecture. For communication and negotiation the auction technique was used. Already to ensure the dynamic optimization, an optimization system was created based on recursive auctions.

The choice of the Multiagent System occurred because it behaves very well in dynamic environments, with the inclusion and exclusion of agents during the execution of the system. In the proposed system, each element is modeled as a software agent. Agents negotiate with each other to achieve a global goal, which is to optimize resources such as flight time and / or reduction of battery consumption. For the proposed system, the MOISE+ tool was chosen for the management of the Multiagent System.

As it is a Multiagent System, being an agent for each Drone, one for each recharge base and one for each point to be visited, the ideal is that the information exchange does not happen in an online form, in order to minimize UAV battery costs. Thus, it was assumed that communication will only take place between the recharge bases, and the system will be fed with information at each Drone landing event.

The chosen method of negotiation was the auction. This method is relatively simple to apply in computational systems from the FIPA Contract-Net protocol and allows agents to communicate within the system regardless of the amount of elements that may vary over time. According to [10], auction-based trading methods are often used to perform the distribution of tasks assigned to teams with dynamic elements.

Analyzing the problem in which the proposed system applies, it is possible to observe an analogy to a traditional auction - Figure 2, in which the Drones would be the consumers, and have as their trading currency their autonomy. They share this information with the bases of refills, which have the role of auctioneer that offer these resources to those interested by

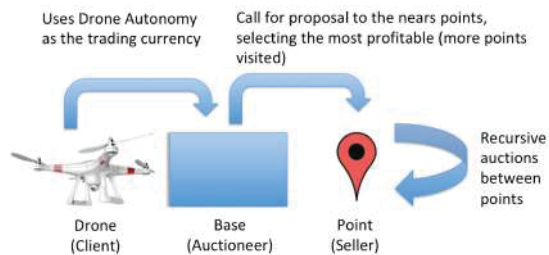


Fig. 2. Analogy of the system based on recursive auctions to a traditional auction.

the auction, which in this analogy are the points that wish to be visited. Each point receives the proposal for the Drone's visit, and the applicant responds to the cost to be visited, as well as how many unique points will be visited on the same flight. The best proposal is chosen based on the desired objective function. In the proposed system, the objective function is to maximize the number of visit points in a single flight.

Information about the system, such as points already explored, information on wind speed and direction, among others, will be fed by the quadcopters at each landing event. This will form a base of artifacts and information will be treated as beliefs within the system, shared with all agents. For the sharing of this information, the Cartago tool was used. For the codification of cognitive agents, which consider beliefs, desires and intentions, the JASON tool was used.

Regarding the flight optimization model used, it should allow for the creation of flight plans with great agility, since with each landing of quadcopters the system is fed with information and new flight plans must be created, optimizing the system as a whole. In this way, the system does not need to find the best global solution, but rather a feasible solution in a short time, given the dynamicity of the environment.

A characteristic of the optimization model is that it is a greedy model executed in an exhaustive way, which guarantees to find the optimal solution for any scenario. As a disadvantage, there is the high computational cost for the processing and the exchange of messages, but for this, specific heuristics can be applied based on the objective function. Details about the heuristics used for the present work are presented in the next section.

#### IV. RESULTS

Simulation tests were carried out to verify the feasibility of the system, as well as to identify its behavior in a future application in a real problem involving quadcopters.

This paper presents the results involving the application of the proposed system, which was modeled to be applied to the precision agriculture scenario, but using an applied architecture perspective, identifying the feasibility of using Multiagent Systems and cognitive agents to guarantee the openness and dynamicity of the system. These characteristics are paramount for cooperative flights in open and unstable environments, as climate changes can occur, such as change of direction and wind speed, as well as early depletion of drones batteries. Given this dynamic nature, the optimization method must be

fast and flexible enough to find feasible solutions quickly and decentrally.

Thus, in this section, in a first moment, will be presented the tests carried out referring to the Multiagent System, in which agents were added and removed dynamically. In this test, we also tested the behavior of the cognitive agents coded in the JASON language, verifying the feasibility of using the beliefs, desires and intentions in relation to the dynamicity of the environment.

In a second moment, the algorithm of optimization based on recursive auctions was codified and tested to identify if it works for the proposed problem.

##### A. Organization and Development of Cognitive Agents

Tests were performed with the MOISE+ tool, and the structural, functional and deontic specification of the proposed Multiagent System was defined. In the visual environment of the MOISE+ test, agents were dynamically inserted and excluded to test the opening of the system. Tests were carried out in scenarios with one and two quadcopters, as well as one and two recharge bases and a different number of points to be visited. All elements are represented in the environment as cognitive software agents.

For the execution of the system, a script was first executed in which an agent was instantiated for each type of system element: quadcopters, recharge bases and points to be visited. This script inserts the elements into the multiagent system, representing the initial state of the environment.

The codes of the agents by category are the same, thus, the agents of the type quadcopter have the same logic, varying only on some characteristics, like autonomy. The same happens for the recharge bases and the points to be visited, and for the latter there is a table of cost of displacement between one point and another. This cost is variable, since it depends on the perception of the environment, such as wind speed and direction.

After the initial script is executed and the agents are inserted into the environment with their initial attributes, the system is started, with all agents sharing their beliefs, desires, and intentions. The JASON language was used to encode such mental states and the CarTAGo tool for sharing artifacts between agents. Thus, in order of inclusion in the environment, each quadcopter type agent initiates an optimization process based on recursive auctions (detailed in the next section). In this process, a flight plan is defined for each vehicle. This plan is executed in a simulation environment developed in Java SE, being possible to verify the message exchange between the agents, as well as the accomplishment of the operations.

In order to test the opening of the system, during the execution of the simulation, quadcopter type agents are excluded in the environment to see how the system behaves in a possible mechanical failure of one of the quadcopters, as well as new vehicles, charging bases, and points for visitation, all this with the running system.

For the present test, a script was created that, during the execution of a simulated mission, inserted and removed quadcopter type agents from the simulation environment, done

randomly. In total, 5 agents were inserted and removed, in an intercalated fashion. Even with these random events, the system was able to execute until the end of the mission, generating flight plans that were executed by the agents, which covered all the area previously registered.

Thus, it was verified that the MOISE+ tool allows the system to be opened safely, allowing the system to continue running independently if new agents are added or removed, as each agent included in the system receives a mission to execute. Upon being withdrawn, the mission performed by him is released so that new agents can take over.

The CARTAGo and JASON tools were critical to the system to maintain openness and dynamism, since the JASON-coded beliefs, desires and intentions are shared with all agents through CarTAGo. Thus, regardless of when, when agents are included or removed, this information remains available to all others, who are aware of the state of the environment and can undertake new activities if necessary.

### B. Optimization with Recursive Auctions

A great potential of the proposed system lies in the technique of distributed optimization based on recursive auctions. This optimization model is designed to be used in this system, but can be adapted to other dynamic scenarios that require optimization.

The choice of the recursive auction is given by the nature of the environment, in which initially the allocation of Drone agent resources to the recharging base agent, and the recharging base agent to the points to be visited agents already take place in the format as shown in Figure 2.

Unlike bioinfo optimization algorithms, such as genetic algorithm, PSO, neural networks and ant colonies, the optimization algorithm based on recursive auctions is already designed to work in distributed environments, requiring no centralized processing to perform the optimization. The processing is done in the various elements of the system, being dynamic, since the optimization continues existing independently of the amount of elements in the system.

The number of auctions required to optimize the system depends on the autonomy of the quadricopter, the longer the autonomy, the more points it can visit, and consequently more auctions will be needed.

To better represent all the messages exchanged for an auction with the four-point scenario, the Sequence Diagram of Figure 3 is shown.

It is possible to observe in the figure that the first point to be tested as a base, Point 1, sends an auction request to its two nearest neighbors, Points 2 and 3. This message is represented by the green color. These two points, respectively, begin a new round of auctions for their nearest neighbors, represented by the red arrow. The third round of auction is represented by the yellow color and the fourth round of auctions by the purple color. In total, 30 messages were exchanged, with 30 simultaneous auctions to reach the best result for the scenario, considering that the base is in Point 1.

With the recharge base at Point 1, you have two optimal results, one going through Points 2, 4, 3 and returning to base

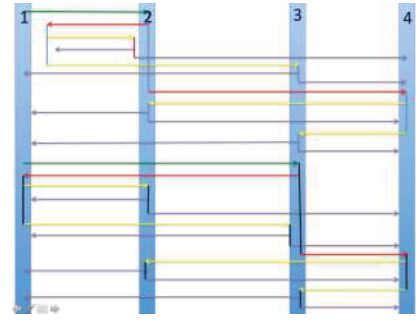


Fig. 3. Sequence diagram with the auctions flow for a scenario with 4 points, considering two neighbors closer and autonomy of 4 u.m. for the Drone.

TABLE II. RESULTS OF THE EXECUTION OF THE ALGORITHM OF OPTIMIZATION WITH RECURSIVE AUCTIONS FOR SCENARIOS WITH 4, 9 AND 16 POINTS.

Scenario	N. of Opt. Sol. Found	N. Auctions Original	N. Auctions after h1	N. Auctions after h2
4 points	8	120	30	8
9 points	234	9198	1022	18
16 points	2.432	2242784	140174	683

in Point 1, and the other passing through Points 3, 4, 2 and returning to base in Point 1. In the sequence, the algorithm considers the base in Point 2 and a new auction round is started.

To show the exponential increase in the quantity of auctions in relation to the increase in the number of points, the Table II is presented, which presents the data for the 4, 9 and 16 point scenarios. In the Original Auctions Qty column the number of auctions is displayed, as well as their exponential increase.

For all scenarios all possible positions for the base are tested for the best placement of the recharge base. For the 4-point scenario, there are 8 possible optimal possibilities. As it was presented, for the scenario with 4 points, for each point 30 auctions were realized, finding two possibilities of optimal path. As the base was tested in the four points of the scenario, 8 optimal possibilities were obtained, with a total of 120 auctions required, with a processing time of 32 ms.

The same test was performed for scenarios with 9 and 16 points. Noting that the number of auctions increased exponentially, as did the processing time.

Thus, to know the amount of auctions that will happen in the system, Equation 6.1 is used, and the quantity of auctions depends on the number of neighbors that the points will have (k), as well as the autonomy of the quadricopter (n).

$$\sum_{i=1}^n k^i \quad (1)$$

To improve the algorithm, two heuristics were proposed that guarantee the reduction of the number of auctions, without significantly compromising the result of the algorithm. These heuristics were designed given the characteristics of the flight of multiple Drones with the application of precision agriculture.

The first heuristic, called h1, defines an initial point for the recharge base, so it is not necessary that all points be tested as possible initial. This point can be set using the most central position of the area, which facilitates the movement to all other points.

Another applied heuristic, called h2, also considerably reduces the number of auctions. It defines that points already visited in an itinerary can not be visited again, since much of the auctions happen with points that have already been visited and receive new requests of visit in the itinerary.

After applying the heuristics h1 and h2, the number of auctions has decreased considerably, as can be seen in the Table II.

For large scenarios with 16 visit points, for example, the reduction in the number of auctions surpassed 99%.

In relation to the global optimum, for the scenario with 4 and 16 points, this optimum was found. For the 9-point scenario, the optimal solution was not found, but a feasible solution.

In this way, it was verified that the optimization algorithm proposed for the present work allows the optimization for simple scenarios and also for complex scenarios, formed by several Drones and points of passage. It is a simple algorithm to be implemented, it is customizable, being possible to be customized through heuristics, given the characteristics of the scenario in which it will be applied. Depending on the heuristic used, one can limit the algorithm to not find the best overall solution, however, the algorithm always guarantees a feasible solution, in distributed environments, in a short time, being this a characteristic for the scenario of flights of Drones applied to precision agriculture.

## V. CONCLUSION

The present work proposes a cooperation system to optimize Drone flights applied to precision agriculture. The scope of this system aims to solve existing limitations in proposals that use Drones to monitor large areas.

For the proposed Multiagent System, it was decided to use three open source tools for the development of the simulation environment: JASON, for the coding of cognitive agents, Cartago, as a repository of beliefs, desires and intentions of the system, and MOISE+ for the organization of the Multiagent System. For the algorithm of negotiation, FIPA Contract-Net was chosen for the development of auctions among the agents in the system.

The proposed Multiagent System was tested and structured based on the problem. The groups of agents, the roles and the functionalities were defined, guaranteeing openness and dynamicity of the system in a simulation environment.

The algorithm of optimization based on recursive auctions was tested in an application developed in Java SE, with graphical interface in Canvas, in which it was possible to test the behavior of this algorithm in different scenarios. The algorithm was stable, obtaining optimum results for all the scenarios tested, regardless of the number of points to be visited and the Drone's autonomy.

As the number of auctions tends to increase exponentially as the number of points to be visited increases, two distinct but simple heuristics have been proposed. They have proved to be very efficient, as they have reduced the number of auctions carried out by up to 99% in scenarios with more points of passage. However, it was observed that in some scenarios, it was not possible to go through all the proposed points, which are believed to be solved with the inclusion of new heuristics.

As referral to this work, once validated in a simulation environment for the actual tests, the system will be coded in 4 Drone Phantom 4 of the company DJI, with the development of the mobile autonomous recharge bases and the communication process between them. The actual tests will serve to compare with the results already obtained in a digital elevation model performed by a RPAS. Quantitative information was obtained, such as flight time to acquire the images, time to organize the images, for the digital processing of the images, as well as the hit and error rate achieved. These parameters will also be measured on flights with 4 Drones for a quantitative comparison.

## REFERENCES

- [1] Eobin Alex George, Gaurav Tiwari, R N Yadav, Edward Peters, and Srishti Sadana. UAV systems for parameter identification in agriculture. In *2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS)*. Institute of Electrical & Electronics Engineers (IEEE), aug 2013.
- [2] Robison Cris Brito, Jose Felipe Loureiro, Eduardo Todt, and Roberto Pereira. A systematic mapping for the scenario of non-urban autonomous vehicle cooperation systems. In *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*. IEEE, nov 2017.
- [3] Gustavo Avellar, Guilherme Pereira, Luciano Pimenta, and Paulo Iscold. Multi-UAV routing for area coverage and remote sensing with minimum time. *Sensors*, 15(11):27783–27803, nov 2015.
- [4] Halit Ergezer and Kemal Leblebicioğlu. 3d path planning for multiple UAVs for maximum information collection. *Journal of Intelligent & Robotic Systems*, 73(1-4):737–762, oct 2013.
- [5] Dac-Tu Ho, Esten Ingar Grotli, P. B. Sujit, Tor Arne Johansen, and Joao Borges Sousa. Cluster-based communication topology selection and UAV path planning in wireless sensor networks. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, may 2013.
- [6] Marcelo S. Menegol, Jomi F. Hübner, and Leandro B. Becker. Evaluation of multi-agent coordination on embedded systems. In *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*, pages 212–223. Springer International Publishing, 2018.
- [7] Nicholas D. Ernest, Kelly Cohen, and Corey J. Schumacher. UAV swarm routing through genetic fuzzy learning methods. In *AIAA Infotech@Aerospace (I@A) Conference*. American Institute of Aeronautics and Astronautics, aug 2013.
- [8] Vincent Roberge, Mohammed Tarbouchi, and François Allaire. Parallel Hybrid Metaheuristic on shared Memory System for real-time UAV Path Planning. *International Journal of Computational Intelligence and Applications*, 13(02):1450008, jun 2014.
- [9] Serkan Caska and Ahmet Gayretli. An algorithm for collaborative patrolling systems with unmanned air vehicles and unmanned ground vehicles. In *2015 7th International Conference on Recent Advances in Space Technologies (RAST)*. IEEE, jun 2015.
- [10] Charles E. Pippin and Henrik Christensen. Cooperation based dynamic team formation in multi-agent auctions. In *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III*. SPIE, may 2012.

# Choosing the best k-neighbor parameters for optimizing a flight of a group of drone using recursive auction algorithm applied to the precision agriculture

Robison Cris Brito\*  
Federal University of  
Technology - Parana  
Pato Branco, Parana  
robison@utfpr.edu.br

José Felipe Loureiro†  
Federal University of  
Technology - Parana  
Pato Branco, Parana  
josefvloureiro@gmail.com

Eduardo Todt‡  
Federal University of  
Parana  
Curitiba, Parana  
todt@inf.ufpr.br

## ABSTRACT

This paper presents the choice of the value for the constant  $k$ , which represents the number of neighbors used in a drone group flight optimization algorithm based on recursive auctions. For the test, we used the algorithm after applying the heuristic named  $h_2$ , which allows us to find feasible solutions in an acceptable processing time. From this point, an analysis was made of the types of points and the shape of the areas that should be overflown in order to identify a proportion between points that need more neighbors than those that need fewer neighbors. In the end, a test was performed in a real scenario with 16,000 meters, or 16 hectares. It has been identified that the number of auctions grows linearly as the number of  $k$  increases, but from four points or more, all runs found the same optimal result, even with a longer processing time.

## CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

## KEYWORDS

Drone, Optimization, Recursive Auction, Precision Agriculture

### ACM Reference Format:

Robison Cris Brito, José Felipe Loureiro, and Eduardo Todt. 2020. Choosing the best k-neighbor parameters for optimizing a flight of a group of drone using recursive auction algorithm applied to the precision agriculture. In *Proceedings of* . ACM, New York, NY, USA, Article 4, 5 pages.

## 1 INTRODUCTION

The use of unmanned aerial vehicles (UAVs) is very common in many areas, especially in civil and military environments. In recent years, with the growth of techniques applied to precision agriculture, the use of UAVs in this scenario tends to be more constant,

\*Professor of the Computer Department - UTFPR-Câmpus Pato Branco

†Egress from the Computer Engineering Course - UTFPR-Câmpus Pato Branco

‡Professor of the Computer Pos-Graduated Program - UFPR-Curitiba

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© Copyright held by the owner/author(s).

allowing to the identification of pests, soil analysis, spraying and sample collection of what is planted.

From a technical point of view, currently two types of UAVs are used in precision agriculture: the fixed-wing, which have the shape of a small airplane, known as Remotely Piloted Aircraft Systems (RPAS), and the rotative wings, a vehicle like a helicopter with several propellers, most known as Drones.

A study presented in [2] did a comparison to identify the most favorable scenarios for the use of Fixed-Wing and Rotating-Wing type air vehicles. It is concluded that for linear flights, for example the monitoring of roads or frontiers, the superiority is held by fixed-wing vehicles. As for flights monitoring two-dimensional areas, where precision agriculture is commonly applied, the superiority is that of rotating-wings, which are able to do the same work in a lesser time than fixed-wings due the flight dynamic.

In this scenario, for the monitoring of large terrain, which is common in precision farming, you can use a set consisting of several Rotative Wings, such as DJI Phantom, in place of a single Fixed Wing, like Arator 5b by XMobot. The first one cost around \$1,200.00 dollars, and the second one cost 12,000.00 dollars, ten times more. If the rotative wings work in a coordinated and optimal manner, the area explored by these vehicles can be much larger than the area exploited by a single fixed wing.

Still, we have enough technology that allows these Rotative Wings to autonomously fly, and human interaction with them is unnecessary, even in the most delicate operations such as landing, battery recharging and takeoff. Also, while the Rotative Wings is on the ground recharging its battery, it may be informed of its next mission, which can be centrally processed by a modern personal computer with high processing power that calculates flight plans for all Rotatives Wings and send to them.

Thus, the biggest challenge lies in optimizing flight plans. Other tasks such as autonomous flight, communication between UAVs, landing, reloading and autonomous takeoff are operational activities that already present solutions in the literature. The challenge lies in optimizing these flight plans when it comes to a set of autonomously coordinated UAVs flying in a dynamic environment where climate change can occur at any time, such as wind direction and speed, operational unforeseen events such as early battery termination or short battery life, for example, or by the dynamics of the environment, as UAVs or recharging bases can be included / removed during the execution of a mission.

A work developed by [1] presents a Drone group flight optimization algorithm, which is a dynamic algorithm based on recursive auctions. This algorithm presents good results in an acceptable processing time, and the quality of its result depends on the amount of neighbors that will be considered in the execution of the algorithm, represented by the constant  $k$ .

Thus, the present work presents an analysis of the value of the constant  $k$  applied to the optimization algorithm based on recursive auctions. This considering a real two-dimensional area, which is an experimental rural area of the agronomy course of a Brazilian university.

## 2 RESEARCH PROBLEM

Although several preliminary studies have been developed and showed the viability of a system that deals with the execution of monitoring missions formed by several Drones, a research topic is still little explored: a faster flight optimization considering the opening of the system and dynamicity of the environment.

Opening is the term used to inform that the system can receive or lose components in the execution time, these components could be UAV or Bases. Dynamicity means that the cost between points can change over time, with the change of wind speed and direction

To be possible for such optimization in a dynamic environment in which UAV group flight is applied to precision agriculture, it is necessary to choose and test a fast, dynamic optimization algorithm that can be easily distributed on different processors. The proposed algorithm for this work is the recursive auction based optimization algorithm.

This algorithm is relatively simple to apply in distributed computational systems from the FIPA Contract-Net protocol and allows the communicate within the system regardless of the amount of elements that may vary over time. According to [3], auction-based trading methods are often used to perform the distribution of tasks assigned to teams with dynamic elements.

Analyzing the problem in which the proposed system applies, it is possible to observe an analogy to a traditional auction - Figure 3, in which the UAV would be the consumers, and have as their trading currency their autonomy. They share this information with the bases of refills, which have the role of auctioneer that offer these resources to those interested by the auction, which in this analogy are the points that wish to be visited. Each point receives the proposal for the Drone's visit, and the applicant responds to the cost to be visited, as well as how many unique points will be visited on the same flight. The best proposal is chosen based on the desired objective function. In the proposed system, the objective function is to maximize the number of visit points in a single flight.

If we run the auctions exhaustively, testing all possible possibilities, we can find the best result for the scenario, but in this situation we have the same problem as the classic algorithms proposed to find solutions to the TSP (Traveling Salesman Problem), for example: time Processing is disproportional, so in a scenario with a few dozen points, a normal computer cannot find results in a feasible time.

The problem presented where a UAV should visit a known number of points only once and return to base is similar to the resolution of the Hamiltonian circuit, where we have a graph in which the



Figure 1: Analogy of the system based on recursive auctions to a traditional auction.

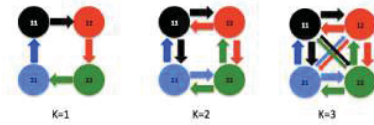


Figure 2: 4 point scenario, with  $k$  ranging from 1 to 3.

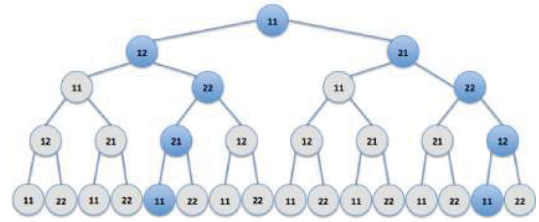


Figure 3: Graph with the possibilities of displacement of a drone considering the initial position the vertex 11,  $k = 2$  and drone autonomy of 4

vertices are the points to be visited. Edges are the paths between one point and another, and the UAV, in turn, has to make the decision as to which points it can visit, optimizing flight time, and consequently the use of its battery. The Drone has to go through all points only once and return to the recharge base.

A system constant that directly influences response quality as well as processing time is the  $k$  constant, which represents the number of neighboring points that a given point can visit and, consequently, the number of auctions that the point will make. If we use  $k$  with small values, results are found at a shorter processing time, but if we use  $k$  larger, better results tend to be found, but with longer processing time. Figure 2 presents examples using  $k = 1$ ,  $k = 2$  and  $k = 3$  for a 4-point scenario.

Considering this same scenario with four points to be visited, and with a value of  $k$  equal to two, and considering that the base of the UAV is in point 11 of the figure, we will have a graph representing the possible visits of the UAV according to Figure 3.

Note that the height of the tree is the autonomy of the drone, that is, the amount of points it can visit. The opening factor of this is 2, which is the constant  $k$ . For a scenario with 4 points to be visited: 11, 12, 21 and 22, 30 auctions (graph edges) were required, in which case 2 global optimal solutions (darker points) were found,

Choosing the best k-neighbor parameters for optimizing a flight of a group of drone using recursive auction algorithm applied to the precision agriculture

which ensure that the UAV passes through all points and return to the starting point.

For the recursive auction of the presented graph, it is considered that the base is at point 11, just like the UAV, thus, it requests the two nearest neighboring points proposals for the UAV visit. These points respond with the Drone's travel cost to them, as well as the Drone's return cost to base. Drone being left autonomous, each of these neighboring points start a new round of auctions, asking their neighbors for new proposals for Drone's visit, with the most lucrative bid being chosen (lowest cost with the largest number of unique points visited) and returned to the point you requested. This sequence of auctions happens exhaustively as long as there is autonomy in the Drone, which ensures that the optimal solution is found.

This process is done by considering the starting point of each base point of the scenario, finding the best solution for each point, as well as the best position from point to base. Regarding the number of auctions executed, for the scenario presented, 30 auctions are held for each point chosen as the starting point of the Drone, as there are 4 points, in total there will be 120 auctions for this scenario.

Thus, one of the heuristics adopted to decrease the number of auctions is to choose, by another algorithm or technique, a point to be the starting point of the Drone, and there is no need to test all possible possibilities, for this purpose. In the 4-point scenario, for example, there is a 75 % decrease in the number of auctions, and this decrease difference is even greater for larger scenarios.

Since the goal is to make an optimization in which the UAV must visit each point once, another time-optimizing heuristic can be adopted: to make each visited point not receive new requests for visits, that is, a great pruning using this one heuristic happens, as shown in dark circle - Figure 3.

Initially, before the heuristic that prevents a point from being visited twice there were 30 auctions, after the application of this heuristic, the number of auctions fell to 8, a reduction of 73.33 %, and for larger scenarios, the reduction in number of auctions is even higher.

The next session will present the result of applying the auction-based optimization algorithm features for some pre-established scenarios, with performance testing.

### 3 RESULTS

In this chapter we will present an analysis of how to find the best value of k (number of neighbors), allowing the algorithm to achieve feasible results in a short time.

For two-dimensional areas, which must be flown over by the Drone group, we have three distinct situations about the optimal number of neighbors.

For vertices the number of neighbors should be small, since these are more isolated points in the two-dimensional area. In the internal points, the number of neighbors should be greater, since each point has connection with other points around it. Finally, at the edges the neighbor number must be a number between the number of neighbors of the vertices (most isolated points) and the internal points (points with connectivity on all sides).

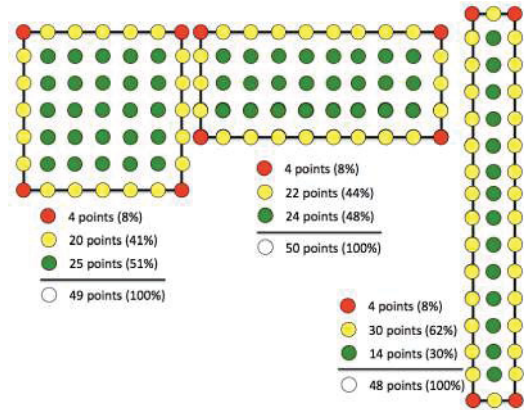


Figure 4: Representation of vertex points (red), edges (yellow) and internal points (green), as well as their proportions in different quadrilateral formats.

In Figure 4 we can see the points with the least connectivity (in red), the points with the most connectivity (in green) and the edge points (in yellow) in different quadrilaterals.

As can be seen in the Figure, all quadrilaterals have the same amount of red dots, which are the edges. In an area formed by a perfect quadrilateral (all equal sides), we have a proportion of the number of inner points larger than edge points, as can be seen in the left figure. The greater the disproportion between the sides, the more edge points relative to the inner points.

In linear areas, similar to the right figure, the number of inner points is smaller than the edge points. According to [2], these scenarios perform better if surveyed by fixed wing air vehicles, as they need to do few maneuvers. The rotating-wing type vehicles perform better when flying over isometric areas, that is, they have the same side measurements, as they usually need more maneuvers to fly over.

For points in red, an interesting amount of neighbors can be two (blue) or three (orange). For the yellow dots, three (blue) or five (orange) neighbors are suggested. Finally, for internal points, four (blue) or eight (orange) neighbors are considered, as noted in Figure 5.

In precision agriculture we have a large number of areas that need to be overflowed, most often being similar to the left area shown in Figure 4. In these, as the proportion of internal points are larger, there must be a greater number of neighbors to obtain good results.

To identify the best proportion of the number of points, tests were made in a real scenario of 340m x 340m, which totals 16,000 square meters, or 16 hectares. This area has an amount of 1.156 points, as can be seen on the simulator screen - Figure 6. For easier viewing, these points were also plotted on an interface without the satellite image - Figure 7.

In the algorithm, the number of neighbors was changed, ranging from 1 to 8, identifying how many unique points were passed. To choose the neighbor, the one with the shortest distance was



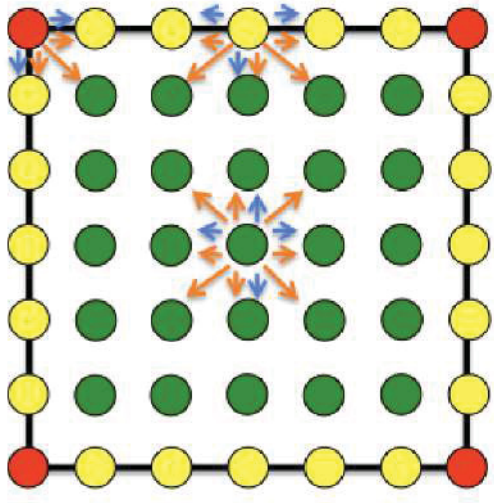


Figura 5: Graphical representation of possible neighbor number representations for each point type in a two-dimensional scenario.

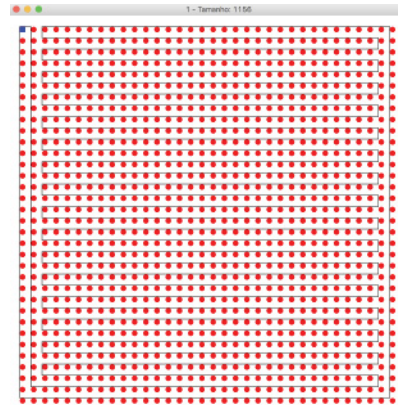


Figura 7: Application Screen that selected a way to pass one time in each point in the selected area.

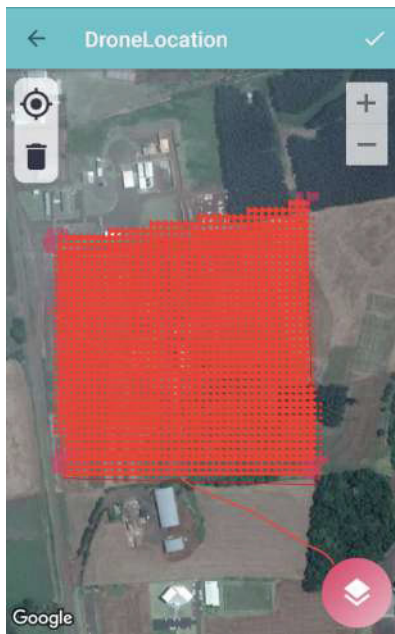


Figura 6: App Screen with all points.

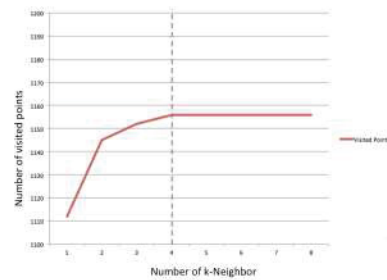


Figura 8: Graph show the increment of number of visited point when has the k variation.

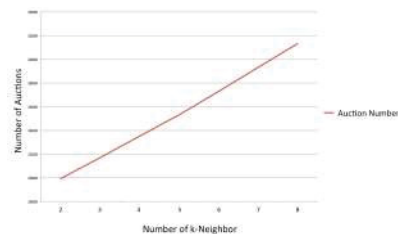


Figura 9: Graph show the increment of number of auctions when has the k variation.

identified. In situations where two or more points had the same distance, a draw was used for the choice.

The result can be seen in the graph in Figure 8. As this is a two-dimensional figure with the same side proportions, with  $k$  equal to four, all points were flown over, from this point on, the result remains the same, although the number of auctions increases, and consequently the time of processing.

The graph in Figure 9 shows the increase in the quantity of auctions as well as the processing time in a linear manner.

Thus, it can be concluded that in common scenarios flown in precision agriculture, using the recursive auction based optimization algorithm, the use of 4 neighbors is sufficient to find feasible results.

Choosing the best k-neighbor parameters for optimizing a flight of a group of drone using recursive auction algorithm applied to the precision agriculture

#### 4 CONCLUSION

The present work proposes choosing a value for constant k in an algorithm for optimization Drones flights applied to precision agriculture. The scope of this system aims to solve existing limitations in proposals that use Fixed Wings to monitor large areas.

The algorithm was tested and presented in [1], however, in the opportunity was highlighted the importance of finding a reference value for k, which is a very important parameter for the algorithm, since varying it allows to explore larger result possibilities, however, in contrast, there is a proportional increase in processing time.

For the tests, it was possible to identify possible values of k, as well as in which situations each value would be more interesting (vertices, edges or internal points). At the end of the work, it was observed that in two-dimensional areas, the value four allows to find feasible results.

In the future, the proposed model and the k value used in recursive auctions will also be tested in other scenarios, such as the optimization of the use of urban electric vehicle batteries, in order to test its instanciability.

#### REFERÊNCIAS

- [1] Robison Cris Brito, Jose Felipe Loureiro, Andre Guedes, and Eduardo Todt. 2019. Optimization System for Dynamic Flight Planning for Groups of Drones using Cooperation with Mobile Recharge Bases by Means of Multiagent System and Recursive Auctions. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE. <https://doi.org/10.1109/compsac.2019.10262>
- [2] Mainara Cristina Lorencena, Robison Cris Brito, Jose Felipe Loureiro, Fabio Favarrim, and Eduardo Todt. 2019. A Comparative Approach on the use of Unmanned Aerial Vehicles kind of Fixed-Wing and Rotative Wing Applied to the Precision Agriculture Scenario. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE. <https://doi.org/10.1109/compsac.2019.10259>
- [3] Charles E. Pippin and Henrik Christensen. 2012. Cooperation based dynamic team formation in multi-agent auctions. In *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III*. SPIE. <https://doi.org/10.1117/12.919551>

# Mobile System for Optimized Planning to Drone Flight applied to the Precision Agriculture

<p>Andre Rabello Federal University of Technology UTFPR rabello_andre@hotmail.com</p>	<p>Robison Cris Brito Federal University of Technology UTFPR robison@utfpr.edu.br</p>	<p>Fabio Favarim Federal University of Technology UTFPR favarim@utfpr.edu.br</p>	<p>Alfredo Weitzenfeld University of South Florida USF aweitzenfeld@usf.edu</p>	<p>Eduardo Todt Federal University of Paraná UFPR todt@inf.ufpr.br</p>
---	---	--	---	--

**Abstract**—This paper presents a mobile app developed to optimize the drone flight in a precision agriculture scenario. The Android platform was chosen, once it have free tools for development and there are many different API that could be used to solve this problem. For map presentation, as well as geocoding manipulation, Google tools were used. For the optimization, an algorithm based on recursive auctions was used, which has the characteristic of finding feasible solutions even in complex scenarios. The app has been tested and achieved feasible results for large scenarios with over a thousand waypoints in just few minutes, even running on a mobile device. It highlights the mobile app, and the recursive auction algorithm, it is an important solution for drone flight optimization in rural areas, where there is usually no possibility to run the application on traditional computers, as usually there is no access to the Internet.

**Keywords**—Drone, Android, Flight Optimization, Precision Agriculture, Recursive Auction

## I. INTRODUCTION

In recent decades, Unmanned Aircraft Systems (UASs) have been used in a growing number in military and civilian applications. These are characterized as Unmanned Aerial Vehicles and can be controlled by a remote control, embedded computer or even via a terminal. In an UAS, two types of Unmanned Aerial Vehicles (UAV) are commonly used, which can be vehicles known as fixed wings, similar to small aircrafts, the other one is named rotary wings, which have pairs of propellers. The last one is popularly known as Drones.

[1] highlights that there is a very high demand for the use of Drones, in the military field for reconnaissance and missions, as well as in the civilian field, in applications such as environmental monitoring, mapping and precision agriculture.

According to [2], the concept of precision agriculture is usually associated with the use of high-tech equipment to evaluate or monitor conditions on a given part of land, and based on these data resources are applied to the land according to its real needs for it. These resources can be seeds, fertilizer, water, among others. Drones have been highlighted in the agriculture scenario, having [3] reported that they are the best platforms for evaluating production, presenting efficiency and speed above the techniques already in use today, allowing the reading of data in any kind of environment or ground surface.

But there are some limitations in the drone handling, such as control accuracy, ability to adapt to faults, safety and

flight range. It should also be considered the fact that must be carefully handled, which has, for many years, making it difficult to be approved for commercial use in many countries, such as Brazil [4], where this research was conducted.

Considering these limitations, it is important to have ways for a good management of the flight and supervising how it is working to get the best results, as well as capturing this data for further processing and analysis, obtaining relevant results in order to achieve the optimization goals regarding results in the agricultural environment.

The application of fixed wing UAVs in the precision agriculture is already common, however, authors as [5] report that this type of vehicle has a high setup cost, and need an exclusive human operator per aircraft, in operations such as taking-off and landing, which take place horizontally and can hardly occur autonomously given the characteristics of the terrain in the scenarios where precision agriculture applies.

On the other hand, Drones have vertical take-off/landing operations, making it easier for these operations to be autonomous [6], as the process of autonomous recharging of its battery [7]. Another advantage in using drones in precision farming is that your flight can take place at slow speeds, and it can stop in the air, going up or down, allowing reading more detailed or more generics, depending on the needs. The advantages of Drones over fixed wing vehicles are varied and are highlighted in [8].

However, probably the biggest advantage when using drones over fixer-wing is in the possibility of using multiple drones, all flying simultaneously, with little to none human interaction. In order for this to happen in an optimized way, there is a need of a flight optimization platform for groups of drones, which must be fast in order to recover from adverse events, and light, so it can be executed in smartphones this being the best computing platform for executing the software at the rural area, even without the processing power of some personal computers, and it has an user-friendly interface along with bluetooth or wifi connectivity, which allows sending data to external devices such as robots and drones.

Thus, the present work presents a platform developed for smartphone, which runs to optimize the flight for a Drone in a precision agriculture scenario. The focus of this work is present an Android Application that is able to run a optimization algorithm and show on feasible plan of flight for Drone. The

focus is not on its usability, but on the algorithms used for it. The application allows to inform the area of land that will be monitored by Drone, as well as the accuracy of the readings, which is used to create the ways-points that will be overflowed for drones. This path plan is done using a based recursive auction algorithm for the optimization. This path plan could be sent to the Drone through WIFI or Bluetooth connection.

## II. RELATED WORKS

In the literature there are many works that use autonomous vehicles, cooperating one with each other to develop a mutual activity, working together and exchanging information about the mission.

It can be highlighted the systematic mapping developed in [9], which informs that 70% of the researched works that use vehicle cooperation are using Drones or other air vehicles for cooperation. Some of these works, in particular, are focusing on the cooperation algorithm., but no one presents the use of a mobile device, such as a smartphone, to perform the processing and optimization for the system.

One of the most significant researches was developed by [5]. This author worked with the cooperation of two Fixed Wing UAVs applied in precision agriculture. The author divides the problem into two parts: first, he uses the principle of particle flow to make Fixed Wings attracted to points they need to pass - waypoints. It then uses an optimization algorithm developed in Mixed Integer Linear Programming (PLIM) to perform the flight optimization of the two UAVs, since the waypoints and the landing and takeoff locations are known. Among the limitations of the work is the fact that it is developed for only two Fixed Wings, besides being a static model, so, before the flight, is executed a PLIM algorithm that generates the flight plans, which will later be executed by the UAVs, so, the processing is done in the computer, with no graphic interface, using only programming language, and after the processing, the result are programmed in the UAV.

According to the author, the number of two fixed wings is ideal for cooperative flights in precision agriculture, because the higher the number of fixed wings, the greater the complexity and the total cost of the system. Fixed Wing Flights have a high cost to setup the aircraft, as well as the need for takeoff and landing intervention. Increasing the number of Fixed Wing increases the number of system operators.

According to the author himself, in a research conducted for the development of his work, the justification for using Fixed Wings on cooperative flights is because most of the scientific work developed between 2005 and 2015 used this type of vehicle.

Although the author uses Fixed Wings, he highlights its limitations, such as restrictions related to the minimum speed of the aircraft that cannot be below a certain limit to avoid the risk of falling and the non-detailed reading of data by high flight speed.

In general, it is possible to observe that few scientific papers deal with rotative wing flight optimization, either, all of this works show presents the algorithm for optimization, but, not how this optimization could be done in a rural environment, with no computers or networks. Also, it is possible

to verify that no article studied deals with the dynamics of the environment, a very common problem faced in real flight environments. It is believed to be because most works deal with pre-optimization rather than optimization during system execution.

## III. SYSTEM DESCRIPTION

According to [10], smartphones are classified through the operating system installed on the device, being the most popular Apple iOS, Google's Android OS, Blackberry's RIM OS and Microsoft's Windows Mobile OS. Android is prevalent in the market, and projections point out that for many years this will remain the leader, due to its openness and ease of customization, putting it ahead of other operating systems [11].

Thus, for the development of the app, we used the Android platform, because it has free tools for development and has numerous libraries for use and manipulation of geographic data.

The main objective of this mobile application is develop an app like a Wizard, that allow the user inform data about the area that will be overflowed, as the feature about the how will be this optimization. For this reason, we opted for the development of several screens, however, each one requesting few information from the user, and at the end of the process, a flight plan will be generated for the Drone, which can be sent over a wireless connection, most often the ad hoc WIFI network created between the Drones and the Android Device.

### A. App Development Methodology

The blocks for generate the flight plans are presented in Figure 1.

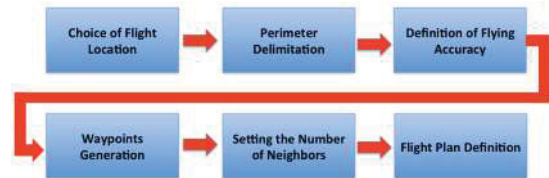


Fig. 1. Flow Diagram among Screens.

How presented in the Figure, to define an optimal flight plan are required six steps on the mobile app. To improve the Graphic User Interface for this app, map features as Google Maps version 16.0.0 were used.

The first step of the wizard is define the location where the monitoring will be performed. The user can enter a specific address, or enter the name of a city or town. For map manipulation we use geocodes, which represent any position of the globe using latitude and longitude. To transform textual data, such as an address, into geocode, we used in the app an API called Google Location and Activity Recognition,

After, the map is displayed on the app with the position entered in the first screen on the center. The user also can navigate on the map, using features of the Google Maps API itself, such as zoom and navigation. Next step will be inform the perimeter of the location that will be overflowed. For it, the user will click on the vertices of the area to be mapped,

forming a geometric figure on the map. To visual mark of these vertices on the map, a non-dragging marker feature of Google Maps was used, and for the perimeter line drawing, the library's Polyline feature was used.

After we define the perimeter, the user have to click at the button located in the lower right, after that, the mobile application calculates the internal points for the selected area. This points is called waypoints. To calculate the waypoints the mobile application needs to respect the distance among the points, for it, we using the Android class Location . At the end, this points are displayed on the screen and it is validated through the Maps Utils API.

These points are drawn on the screen from a pivot point, which consists of the lowest longitude and lowest latitude added to a coefficient between the user-entered distance and the radius of the Earth, where the next point will be the previous longitude, plus this coefficient. This process is repeated until the longest edge is reached. At this point, the coefficient is added to latitude and the process is repeated from the shortest to the largest, always adding the coefficient to longitude, performing the process to the value generated. be larger than the highest latitude edge.

After completing the point calculation step, the user have to click the right aligned button in the application Toolbar which will present a dialog box asking to the user how many nearby points (k) should be calculated by the application. After informed the k value, the application persists all of this data: the points and its neighbors. To persist, we used SQLiteDatabase feature, that initially save the information on the smartphone, and after, share this information using wireless with the Drone. The user can see the mapping again, or another old one, through the SideMenuBar; Mappings, which presents the list of mappings already made and stored in the local database.

To choose the k nearest neighbors, one of the most costly parts of the processing, we needed to calculate the distance between points using the mathematical formula:

$$\text{acos}(\sin(\text{lat} / 180.0 * \text{PI}) * \sin(\text{lat} / 180.0 * \text{PI}) + \cos(\text{lat} / 180.0 * \text{PI}) * \cos(\text{lat} / 180.0 * \text{PI}) * \cos((\text{lat} - \text{lat}) / 180.0 * \text{PI})) / 180.0 / \text{PI} * 60 * 1.1515 * 1.609344.$$

This formula will return the distance in meters between the points. Having calculate the distance between one point with all of the other points in the scenario, it necessary to do a simple sort on the list to identify the closest points.

Finally, based on this information, the recursive auction algorithm will be applied, which will run until its found a optimal result or until a timeout is reached. This timeout can be informed by the user on the mobile app. This algorithm will return a path, similar to the Hamiltonian Circuit, which basically consists of passing through all points in the map only once and return to the starting point.

Algorithms for finding optimal solutions for a Hamiltonian circuit usually are very slow, incompatible for mobile processing in a smartphone, and do not consider dynamics in the environment, such changing the value of the Hamiltonian circuit edges, that we have in the scenario of Drone flight in the Precision Agriculture. This edge values could be wind speed and direction, which is often dynamic.

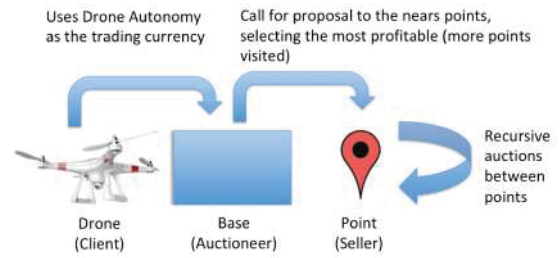


Fig. 2. Analogy of the system based on recursive auctions to a traditional auction.

The recursive auction algorithm adopted by this scenario works independently of the number of points to be visited, and works even when have changing in the scenario, once it is a dynamic algorithm. With the recursive auction algorithm, it is possible, for example, to increase or decrease the number of waypoints, as well as the costs of moving from one point to another.

### B. Methodology for Optimization Algorithm Development

To be possible for such optimization in a dynamic environment in which UAV group flight is applied to precision agriculture, it is necessary to choose and test a fast, dynamic optimization algorithm that can be easily distributed on different processors. The proposed algorithm for this work is the recursive auction based optimization algorithm, presented in detail in this section and evaluated in the following session.

This algorithm is relatively simple to apply in distributed computational systems from the FIPA Contract-Net protocol and allows the communicate within the system regardless of the amount of elements that may vary over time. According to [12], auction-based trading methods are often used to perform the distribution of tasks assigned to teams with dynamic elements.

Analyzing the problem in which the proposed system applies, it is possible to observe an analogy to a traditional auction - Figure 4, in which the UAV would be the consumers, and have as their trading currency their autonomy. They share this information with the bases of refills, which have the role of auctioneer that offer these resources to those interested by the auction, which in this analogy are the points that wish to be visited. Each point receives the proposal for the Drone's visit, and the applicant responds to the cost to be visited, as well as how many unique points will be visited on the same flight. The best proposal is chosen based on the desired objective function. In the proposed system, the objective function is to maximize the number of visit points in a single flight.

If we run the auctions exhaustively, testing all possible possibilities, we can find the best result for the scenario, but in this situation we have the same problem as the classic algorithms proposed to find solutions to the TPS problem, for example: time Processing is disproportional, so in a scenario with a few dozen points, a normal computer cannot find results in a feasible time.

To decrease computational complexity, heuristics can be used to decrease processing time without significantly compromising the result found, as well as meta-heuristics can also be

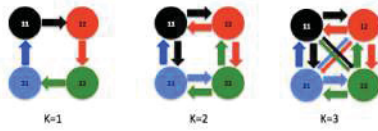


Fig. 3. 4 point scenario, with  $k$  ranging from 1 to 3.

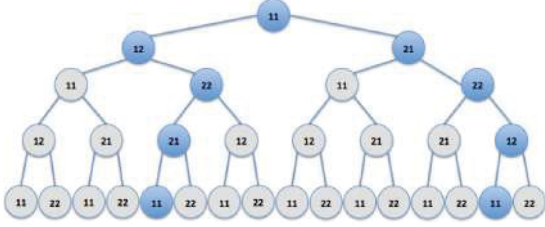


Fig. 4. Graph with the possibilities of displacement of a drone considering the initial position the vertex 11,  $k = 2$  and drone autonomy of 4

applied to heuristics, allowing to considerably improve results without significantly compromising processing time.

The problem presented where a UAV should visit a known number of points only once and return to base is similar to the resolution of the Hamiltonian circuit, where we have a graph in which the vertices are the points to be visited. Edges are the paths between one point and another, and the UAV, in turn, has to make the decision as to which points it can visit, optimizing flight time, and consequently the use of its battery. The Drone has to go through all points only once and return to the recharge base.

A system constant that directly influences response quality as well as processing time is the  $k$  constant, which represents the number of neighboring points that a given point can visit and, consequently, the number of auctions that the point will make. If we use  $k$  with small values, results are found at a shorter processing time, but if we use  $k$  larger, better results tend to be found, but with longer processing time. Figure 3 presents examples using  $k = 1$ ,  $k = 2$  and  $k = 3$  for a 4-point scenario.

Considering this same scenario with four points to be visited, and with a value of  $k$  equal to two, and considering that the base of the UAV is in point 11 of the figure, we will have a graph representing the possible visits of the UAV according to Figure 4.

Note that the height of the tree is the autonomy of the drone, that is, the amount of points it can visit. The opening factor of this is 2, which is the constant  $k$ . For a scenario with 4 points to be visited: 11, 12, 21 and 22, 30 auctions (graph edges) were required, in which case 2 global optimal solutions (darker points) were found, which ensure that the UAV passes through all points and return to the starting point.

For the recursive auction of the presented graph, it is considered that the base is at point 11, just like the UAV, thus, it requests the two nearest neighboring points proposals for the UAV visit. These points respond with the Drone's travel cost to them, as well as the Drone's return cost to base. Drone being left autonomous, each of these neighboring points start a new round of auctions, asking their neighbors for new proposals

for Drone's visit, with the most lucrative bid being chosen (lowest cost with the largest number of unique points visited) and returned to the point you requested. This sequence of auctions happens exhaustively as long as there is autonomy in the Drone, which ensures that the optimal solution is found.

This process is done by considering the starting point of each base point of the scenario, finding the best solution for each point, as well as the best position from point to base. Regarding the number of auctions executed, for the scenario presented, 30 auctions are held for each point chosen as the starting point of the Drone, as there are 4 points, in total there will be 120 auctions for this scenario.

Thus, one of the heuristics adopted to decrease the number of auctions is to choose, by another algorithm or technique, a point to be the starting point of the Drone, and there is no need to test all possible possibilities, for this purpose. In the 4-point scenario, for example, there is a 75 % decrease in the number of auctions, and this decrease difference is even greater for larger scenarios.

Since the goal is to make an optimization in which the UAV must visit each point once, another time-optimizing heuristic can be adopted: to make each visited point not receive new requests for visits, that is, a great pruning using this one heuristic happens.

Initially, before the heuristic that prevents a point from being visited twice there were 30 auctions, after the application of this heuristic, the number of auctions fell to 8, a reduction of 73.33 %, and for larger scenarios, the reduction in number of auctions is even higher.

The next session will present the result of applying the auction-based optimization algorithm features for some pre-established scenarios, with performance testing.

#### IV. RESULTS

As a result of this work, we have an Android mobile app that is able, through a simple and interactive interface, to request data for a rural area that needs to be overflowed by a Drone, and then generates a feasible flight plan, not necessarily optimal, that allow the drone flight above all the waypoints, returning to base at the end of the mission.

The application consists in a sequence of screen, these shown in Figure 5.

As we can see, the Screen a) requests a textual identification of where the Drone mapping will be performed. This can be the name of a city or an address. A name for the location is also requested, which will be used for persistence in the cell phone database. In the first screen is requested the accuracy of reading data in meters. On the next step - screen b) the map with the location entered on the screen a) is shown. In this screen, the user can navigate and use the zoom. In c) the perimeter of the area where it will be overflowed is defined, and in d) the waypoints are presented considering the reading accuracy informed on the screen a). In e) is requested the number of neighbors that will be considered in the processing of the recursive auction optimization algorithm, and at finally, in f) is presented part of the screen with the flight plan, showing the points where the Drone should pass in the area that will be monitored.



Fig. 5. App Screen.

Due to the small size of the mobile screen, it is not possible to see the flight route for the example presented above, which is a 340x340m, ie it has 34x34 waypoints, totalizing 1.156 point at all.

We used a program to delimit this area and put all the point with 10 meters of precision. Considering these 1,156 points, and using 4 neighbor for each point, we used the optimization algorithm based in recursive auction to return the best way to pass in all of this 1,156 point only one time. After 176 seconds of processing the algorithm found the first optimal situation.

The be easy to see the way, we put in a simulation environment, without background map, all the points and was traced the UAV path. How we have a lot of point in the map, probably we have a lot of option of way to the drone pass in all points, and probably the algorithm will spend a lot of time to returns all of the results, but, for our problem, we need quicker answer, because it, we use only the first best result.

After the processing, this flight planning could be send to Drone using some specific SDK. For example, and DJI Phantom allow exchange information with other devices, like Smartphone Android, using DJI Developer SDK through WIFI.

## V. CONCLUSION

This paper presented the development of a mobile application to aid the optimized planning of a drone's flight plan, applied to precision agriculture.

It was only used to develop free tools, that allow the app to run on Androids devices. The application's graphical user interface allowed the user to inform the required information using screens with few graphics, in order to make the application easier to use.

Additional tools and API, such as Google Maps, Google Places, and Geocoding API were used for the project development, allowing a better usability as well as greater accuracy in the generated data. The result of the processing was the

optimized flight plan, which can be sent to Drone through specific communication APIs.

Even when ran on a cheaper Android Device, the result of a solution happened in minutes, even for a scenario of more than 1,000 points. For the development of the heuristic used in the recursive auction algorithm, it was used features that allow the algorithm to find faster results in quadratic areas, but even in complex figures, formed by several vertices, such algorithm also allows to find feasible solutions, but in this case, the processing time could be longer.

As future work, we suggest improving the application as well as the algorithm to handle the optimization of multiple drone flight.

## REFERENCES

- [1] Gabriele Bastianelli, Diego Salamon, Alessandro Schisano, and Alessandra Iacobacci. Agent-based simulation of collaborative unmanned satellite vehicles. In *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*. Institute of Electrical & Electronics Engineers (IEEE), oct 2012.
- [2] Miguel Castro Neto, Pedro A. Pinto, and José Paulo P. Coelho. *Tecnologias de informação e comunicação e a agricultura*. Porto: Sociedade Portuguesa de Inovação, 2005.
- [3] Eobin Alex George, Gaurav Tiwari, R N Yadav, Edward Peters, and Srishti Sadana. UAV systems for parameter identification in agriculture. In *2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS)*. Institute of Electrical & Electronics Engineers (IEEE), aug 2013.
- [4] FAB. Comando da aeronáutica publica nova legislação sobre aeronaves remotamente pilotadas. In <http://www.fab.mil.br/noticias/mostra/23937/>. Acessado em 12 de maio de 2016., dez 2015.
- [5] Gustavo Avellar, Guilherme Pereira, Luciano Pimenta, and Paulo Iscold. Multi-UAV routing for area coverage and remote sensing with minimum time. *Sensors*, 15(11):27783–27803, nov 2015.
- [6] Min-Fan Ricky Lee, Shun-Feng Su, Jie-Wei Eric Yeah, Husan-Ming Huang, and Jonathan Chen. Autonomous landing system for aerial mobile robot cooperation. In *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*. IEEE, dec 2014.
- [7] Jose Loureiro, Robison Cris Brito, Carlos Magrin, and Eduardo Todt. A model for landing, taking off and autonomous battery recharging of a parrot ar.drone 2.0 using computational vision and gps features. In *Computer on the Beach 2019*. COTB, fev 2019.
- [8] Mainara Cristina Lorencena, Robison Cris Brito, Jose Felipe Loureiro, Fabio Favarim, and Eduardo Todt. A comparative approach on the use of unmanned aerial vehicles kind of fixed-wing and rotative wing applied to the precision agriculture scenario. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, July 2019.
- [9] Robison Cris Brito, Jose Felipe Loureiro, Eduardo Todt, and Roberto Pereira. A systematic mapping for the scenario of non-urban autonomous vehicle cooperation systems. In *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*. IEEE, November 2017.
- [10] Heloise Pieterse and Martin S Olivier. Android botnets on the rise: Trends and characteristics. In *2012 Information Security for South Africa*. IEEE, aug 2012.
- [11] Ya-Lin Liu and Fang-Mei Tseng. Combining scenario analysis with the diffusion model and the competitive model for analyzing the development of the smartphone operating system. In *PROCEEDINGS OF PICMET '12: TECHNOLOGY MANAGEMENT FOR EMERGING TECHNOLOGIE*. IEEE., dez 2012.
- [12] Charles E. Pippin and Henrik Christensen. Cooperation based dynamic team formation in multi-agent auctions. In *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III*. SPIE, may 2012.

# Implementing a Communication Network between Bases Station applied for Group of Drones

Julio Opolski Netto  
Federal University  
of Technology  
UTFPR

julionetto@alunos.utfpr.edu.br

Robison Cris Brito  
Federal University  
of Technology  
UTFPR

robison@utfpr.edu.br

Fabio Favarim  
Federal University  
of Technology  
UTFPR

favarim@gmail.com

Luis Felipe Priester  
Federal University  
of Technology  
UTFPR

luisf.priester@gmail.com

Eduardo Todt  
Federal University  
of Parana  
UFPR

todt@inf.ufpr.br

**Abstract**—The use of Unmanned Aerial Vehicle (UAV) have been shown to be increasingly frequent for a diversity of applications, mainly in agriculture. The mapping of large areas for analysis purposes is common and it is considered a challenge due to the short range of the UAVs. The base stations utilization for drone recharge and important information obtainment is a relevant proposal. This paper features a low energy cost long range communication system between in base stations. Using IoT concepts and the possibility of utilizing a diversity of communication protocols in just a single device, this paper shows the integration between microcontroller, server and operator interface. The developed system is capable of indentifying a drone that just landed in a base station through RFID technology, and send this and other informations in real time through the command line “gateway” to the server using LoRa technology and MQTT protocol.

**Keywords**—IoT, Drone, Mapping, MQTT, LoRa

## I. INTRODUCTION

The disponibility of natural resources, the competence of the farmers and the chain production organization, along with Science, Technology and Innovation, all together contributed significantly to Brazil’s economical development, making it one of the biggest producers and exporters of agricultural products during the last decades. [1].

The use of new technologies in the primary sectors of the economy, mostly in agriculture, aided the increasing data collection from the productive areas. In order to assure the sucess of all the production and to avoid several issues faced in the planting stage such as plagues, low productivity, controlling nutrients and soil acidity and obtaining information about the factors which interact inside the culture and the other necessary information to maximize productivity, all of extreme relevance. The union of both technology and agricultural practices is named Precision Agriculture (PA) and aims to aid the farmer in obtaining higher levels of control and better productivity results.

Technologies which use Unmanned Aerial Vehicles (UAV), Radiofrequency Devices (RD), apps using IoT and agricultural machines can be named as some of the main responsible devices for this transformation. According to [2], up to 2020 the PA Market is expected to grow from \$730 million up to \$2.4 billion dollars.

Authors such as [3] name the importance of using unmanned aerial vehicles in the precision agriculture. Fast and

efficiently, UAVs are able to perform the most varied forms of tasks in basically all types of terrain, such as mapping areas in order to obtain data using the Elevation Digital Model (EDM), analyzing and the automatic monitoring of the soil and the plantations, along with plague control added to that.

One of the evolutions in the use of UAVs in the agricultural environment is the practice of autonomous flights. Such practice opens ground to inspecting and monitoring great areas of plantations without the need of an actual operator for each UAV. However, using UAVs for great areas still present a few limitations such as its battery consumption, affecting directly the size of the area to be analyzed [4].

One possible solution for the coverage of a great area according to [4] is the use of a group of UAVs. In order to further improve the results, these UAV may work connected to base stations. Such stations would have the capacity to recharge the drones, collecting information out of it and out of the soil as well, and also transmit new missions/tasks to the drones. This way, the integration of the base stations to the drones increases substantially the amount of data obtained through the whole system.

A great number of devices interconnected and sharing information among themselves demand the implementation of a communication network fault tolerant and of high reliability, in order for the data exchange to occur under the planned design. However, due to the distance, it is not always that the base station can communicate to the central base. One way to treat this issue is using one dynamic network which adapts itself autonomously in a way to maintain connectivity.

This paper aims to present an offer of communication among multiple bases distantly located, which will behave in a dynamic fashion in order to transmit data to a gateway. Such gateway must send the information to a server or a broker MQTT and at last all the received/sent information may be verified through a web interface.

## II. RESEARCH SCENARIO

The current project is part of a bigger one, this being proposed by [5], which presents a pattern of drone group’s flight applied into precision agriculture, this using cooperation with a mobile recharge base.

The architecture that the current project inserts is exhibited in the figure 1. This posses a group of drones, which were



programmed to autonomously land and take off. Their recharge base are automatic *off-road* vehicles, which can rearrange on the terrain to be monitored while the drones standby recharging. The system's communication, being the focus of this paper, happens only in between recharge bases because energy is not limited in those, differently from drones, thus, each base posses a mechanism that allows its communication with each other, ensuring the well distributed optimization algorithm's execution proposed. For the mission's management by a user, there will be a web system of mission management, where the drones are assigned to the mission, as well as their recharge bases.

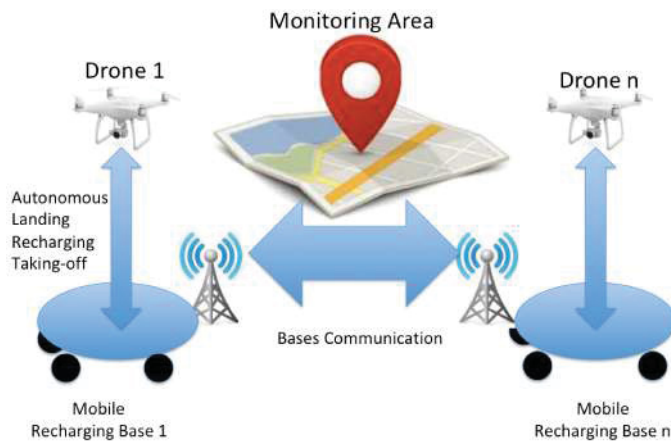


Fig. 1. Model overview applied to optimize the flight of several Drones.

Following will be presented the challenges that will be faced when implementing the MAOVGD to the presented problem, as well as previous works developed or collaborated with by this author, aiming the application of the MAOVGD.

#### Explicar a sigla

- 1) **Autonomous Landing/Recharging/Taking-off** : This subject was already explored before by authors such as [6] and is feasibly possible. Plenty of systems allow that the drone returns to its dock using its own GPS resources, thus when it is flying over the dock it is possible for it to land using computer vision.
- 2) **Mobile Recharging Dock:** Although not obligatory, the recharging dock in rural environments may be mobile, such as a vehicle for example, fed through solar panels. Thus, when the drone lands and starts its charging process, which may take several minutes, the charging dock is free to move to a different strategical location, aiming to optimize the flight time regarding the vehicle movement. Some papers propose such vehicle, as presented in [7]. Referring to the positioning of the autonomous vehicle in a new strategical position in the area to be explored, there are specific techniques for the choice of the best positioning of the mobile recharging dock, one of these techniques is presented [8].
- 3) **Area to be monitored:** for the execution, the area to be monitored must be known and registered in a mission management system. In this system, every point of view must be informed, as well as out-

of-bounds points, starting dock position, number of drones needed for said mission, amount of available recharging docks, dock positioning, distance between the data points, among others. As the management system involves controlling multiple drones, it must be as simple as possible for the user, in a way that it informs only the necessary data for the mission start, and after that the control must be carried on autonomously through an intelligent system. The web system for the multiple drone flight management was developed in [9].

- 4) **Communication between the charging docks:** the optimization of the drone's autonomy is a crucial factor for the proposed model, since this is the great limitation of it use in precision agriculture. For such optimization, an important point to be considered is to limit the information exchange between the drones during the mission execution, i.e., after they took off. This way, it is suggested that the communication occurs mainly in between the docks once stated that said docks have solar-based charging mechanisms, with no battery limitations. This being, once the drones are back to the charging dock, it updates the whole system with the updated model, sharing the areas it mapped, updating as well information such as sensor data and camera, wind speed and weather conditions. The model processes such information and define a new flight plan for each drone. This communication process is named "half-line" and it is event-based.

In the presented scenario, this paper offers a communication mechanism among the base-station/docks. This mechanism has low energy consumption, once the docks/stations have no constant energy feed needed. Also, it must allow a long range communication in a transfer rate that is enough to share data such as if one specific drone landed in a specific station base, obtaining such information through sensors.

### III. MATERIALS

For the development of the base stations, the gateway and the server, the following electronic components were used:

- Heltec ESP32 LoRa: The Heltec module was used along with the ESP32 microcontroller to identify the drone using a RFID tag coupled to the drone, as well as a RFID RC522 reader integrated to the module. The module, located in the base station, is responsible for the communication with the gateway using the LoRaWAN technology, already integrated to the module through a LoRa SX1278 chip. The Heltec ESP32 LoRa module (Figure 2.a) has the following specifications: Tensilica Processor LX6 ESP32 240MHz Dual-Core; Chip LoRa SX1278; 3x UART; 2x SPI; 2x I2C; 1x I2S; 4MB(32M-bits) SPI FLASH;
- Antena Steelbras DUAL VHF/UHF AP0188: A AP0188 (Figure 2.b) is a dual band antenna which operates in the VHF and 5/8 UHF range used along with Dual band transceivers. The antenna has the following specifications: Frequency VHF: 144 - 148 MHz; Frequency UHF: 430 - 440 MHz; Max Potency:

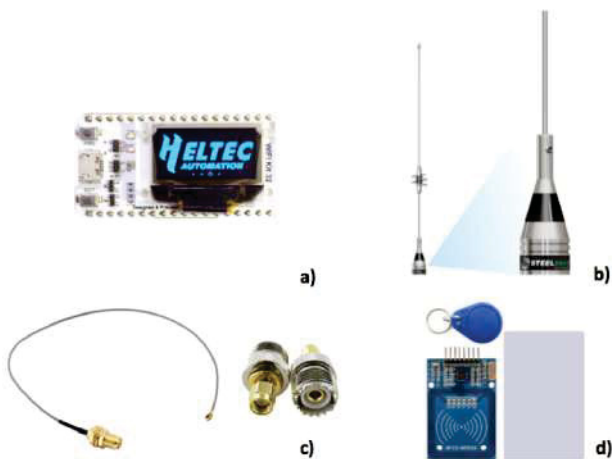


Fig. 2. a) Heltec ESP32 LoRa with integrated display. b) AP0188 Antenna. c) Pigtail and UHF adapter. d) RFID Reader RC522, tag e Card.

150 Watts; Impedance: 50 Ohms; VHF Gain: 0 dB - 2,15 dBi; UHF Gain: 3 dB - 5,15 dBi; Height: 440mm; Weight: 105g. Two adapters were coupled to the antenna as shown in the Figure 2.c).

- Reader module and RFID tag: This module (Figure 2.d) is able to obtain the identification from a tag or card, with a reading range up to 5 centimeters high. The tag must be coupled to the drone, allowing it to be identified through the reading device at the base station

From the application development point of view, the following tools were used:

- Visual Studio Code and PlatformIO: Visual Studio Code is a free and multiplatform text editor distributed by Microsoft for app developing. This editor allows extensions and plugins to be added, amplifying the range of programming languages supported. For this Project, the PlatformIO ambient was used, installed through the Visual Studio Code and used in its own IDE. The PlatformIO is an ambient aimed to IoT developments, allowing the creation of projects towards several embedded boards and installations along with several supports, containing an internal terminal along with a serial port monitor.
- Docker: the Docker is an open-source platform written in Go, a programming language developed by Google which facilitates the creation and administration of isolated environments. Different from a traditional virtualization system, in which an Operating System (OS) is virtualized complete and isolated, Docker has isolated resources, making it possible for the packaging of an application or a whole environment inside a container, making such environment portable to any other host which has Docker installed. Docker was used in this paper for the creation and maintenance of the database in MongoDB.

- Studio 3T: The Studio 3T is a multiplatform IDE for the MongoDB. Counting with several functionalities, such as the ability to perform searches and explore directly the information from the database using basic queries in SQL. The Studio 3T was used in this project to manage the database which will store the information in the base station.
- Node-RED: Aiming to connect the hardware devices, APIs and online services, Node-RED is used as a visual programming tool. Created by IBM Emerging Technology focused in IoT, Node-RED allows the simplified connection of devices and counts with several functionalities such as creating routines, user interfaces, receiving event ins HTTP and MQTT protocols and making the integration to the database. Allows to create nodes and programming in a native way in Java Script or several other languages.
- Mosca MQTT: It is an open-source MQTT broker light and flexible to be used in servers as well as in devices which require low energy consumption such as microcontrollers. It may be executed either independently or embedded in Node.JS applications using Java Script programming language, its installation and execution is quite straightforward.
- Mongo DB: Instead of using the table concept as used in most of the database, MongoDB uses NoSQL data structure, using the collections concept. Both concepts are quite similar, however the collections are not as relational and do not have a fixed structure, meaning that the data might be stored as documents, graphs, keys/values or columns in the SQL. The data is stored in documents using the JSON format.

#### IV. METHODOLOGY

This work was built based in the IoT architecture. It has four main components, being: base station, gateway, server and final user, according to the Figure 3.

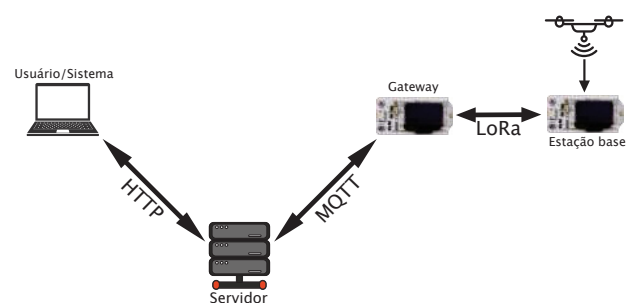


Fig. 3. Work architecture.

As soon as the drone lands on the base station, the station will communicate to the gateway using the LoRa communication technology, transmitting the needed information. The gateway will then communicate to the server using the MQTT communication protocol, making it easier to manipulate the information and its visualization by the operator down in the ground.

Find below the main components of this work, and the scheme of how they communicate

### A. Base Station

The base stations have the Heltec modules, ESP32 LoRA model coupled with a RFID reader and an external charger to feed the module.

The base station has as its main purpose to obtain information through the RFID tag that is coupled to the drone, transmitting this and other relevant information into the gateway.

As soon as the drone approaches the base station, the microcontroller must perform the RFID tag reading. A previously developed routine performs this reading, as well as retrieving other information, transmitting it to the gateway using the LoRaWAN protocol. The diagram presented in Figure 4 represents the landing process for the drone on the base station.

### B. Gateway

The gateway is represented by the Heltec module itself, however the module is able to exchange information with the base stations using the LoRaWAN protocol, and with the server using the MQTT protocol. However, the gateway is a “bridge” in between the base stations and the servers. The diagram presented in Figure 5 shows the gateway operation process.

The gateway is responsible to verify periodically, during a predetermined time window, the base stations status and pass on this information to the server.

### C. Server

The server is a computer or a microcontroller able to store and manage in a database all the information collected by the base stations which were received through the gateway. Besides maintaining the web interface that is used by the operator and allows him to receive and send commands to the base stations using the gateway.

For this work, a laptop was used and installed the Docker application, which is able to run the MongoDB database and the Node-RED application. After running the database and Node-RED, an application developed in NodeJS is ran, which implements the connection to the database, the broker MQTT Mosca initialization, the subscription to the predefined topics and all the routines for the events received in the broker MQTT topics.

With the Node-RED tool, a flow was created capable to manage all the communication from the broker MQTT, allowing to check the gateway and base station status, providing a user interface for the operator to visualize the information.

### D. Operator Graphic User Interface

Application developed and kept by the server with the main goal to allow the control and monitoring of the base station and drones by the ground operator. The application was developed using Node-RED.

The application is capable of managing the information received through the broker MQTT and allows the creation of a user interface which will be used to show in real time the

status of all base stations and gateways, informing any landing from the drones on the base stations.

Node-RED allows to insert all components (labels, buttons, graphs, notifications) needed for the visualization and use by the user according to the figure 6.

As the base stations and gateway were fed through an external charging device, the value for the battery level is merely illustrative. The values that will be received and updated on the operator’s interface is the gateway and base stations IDs, the time of the last received information, base station status, if it is online or offline and if any drone has landed. In case there is a drone landed on the station, its ID is shown on the interface.

## V. RESULTS AND DISCUSSION

In order to verify the distance and data loss under a real use environment, several scenarios were used for the test. For the distance tests, three scenarios were ran, two rural and one urban environment.

On the first rural environment scenario, tests were performed in several points with different elevations, and it was observed that in points with Non-Line of Sight (NLoS) between the fixed module that was sending the information and the mobile module which was receiving the information, the data package loss and loss of signal was considerable, however in points with Line of Sight (LoS) the loss was minimal.

Using this information and moving forward to the next test scenario, it was found a point where the Line of Sight is 9700 meters. The communication between the devices ran without any significant data loss.

The third test scenario was an urban environment, which contains all kinds of possible interferences such as buildings, communications, among others. The data loss occurred quickly in points with NLoS and the communication occurred successfully without any significant data loss in a point 5 km far on the LoS.

In every test scenario, data loss tests were performed, where a data package was created containing 52 bytes of random information along with a counter to verify if any loss occurred. The CayenneLPP library was used in order to optimize the data that is sent through low transfer rates, organizing the information by type and coding/decoding them into the JavaScript Object Notation (JSON).

In the developed code, 200 packages were sent each test, following the interval of 2 seconds each. With the receptor receiving the information into JSON to check if the package received was the exactly one supposed to be received, or if there was any data loss.

The Table I illustrates the obtained results regarding the distance and data loss.

Distance	1m	900m	2000m	3000m	5000m	10000m
Data Loss (%)	0.225	0.5	1	0.5	0.5	0.7

TABLE I. OBTAINED RESULTS FROM THE PACKET LOSS TEST

According to the obtained results, it was possible to observe that the communication in between the modules is satisfactory

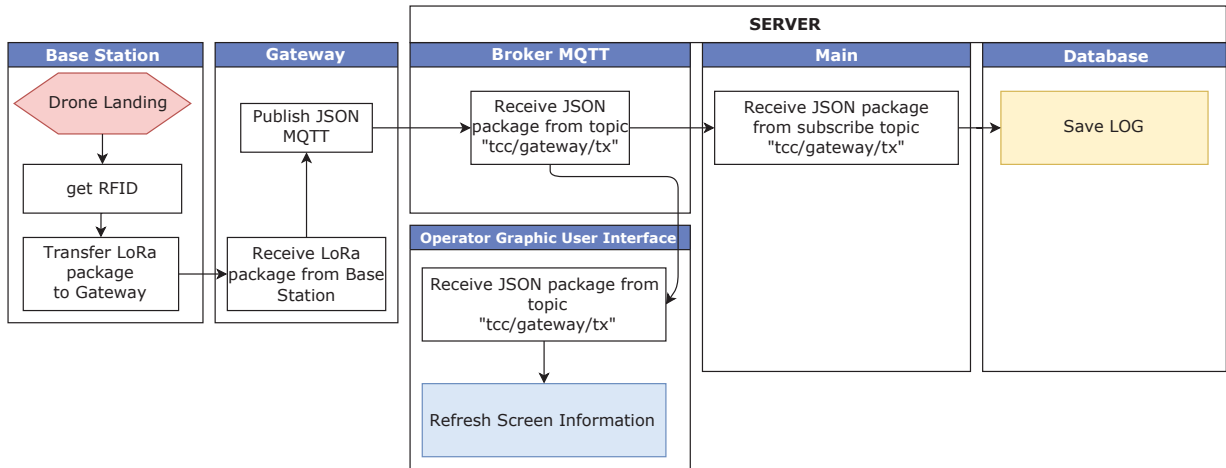


Fig. 4. This Diagram show the landing of a drone in the base Station.

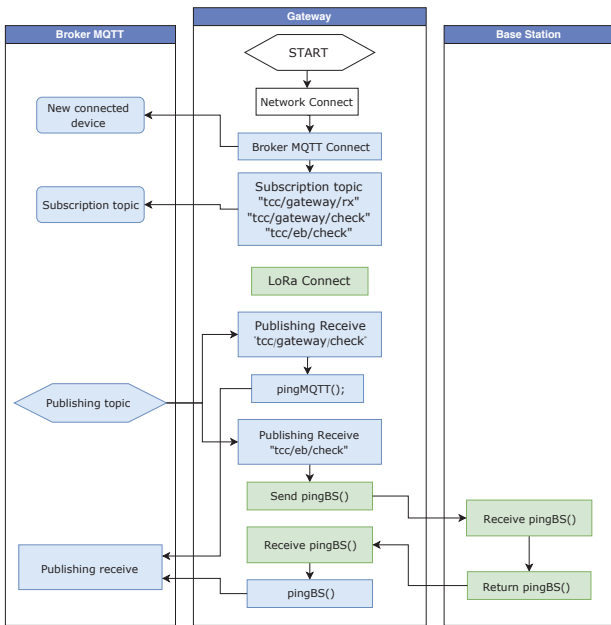


Fig. 5. Diagram that show the gateway process.

as in short distances as it is in long distances, as long as there is one Line of Sight.

## VI. CONCLUSION

This paper aimed developing a communication system capable of acting, sensing, processing and communicating in a efficient way with a long range and low energy cost in between base stations. By possessing a huge diversity of applications, mainly when it comes to the Internet of Things or “Internet of Things (IoT)” the LPWAN networks were shown ideal, and are becoming a option that is more commercially chosen.

Initially, the objective was focused in obtain a 5 kilometers range between base station and the command line “gateway”, enough distance to meet the requirements proposed by the

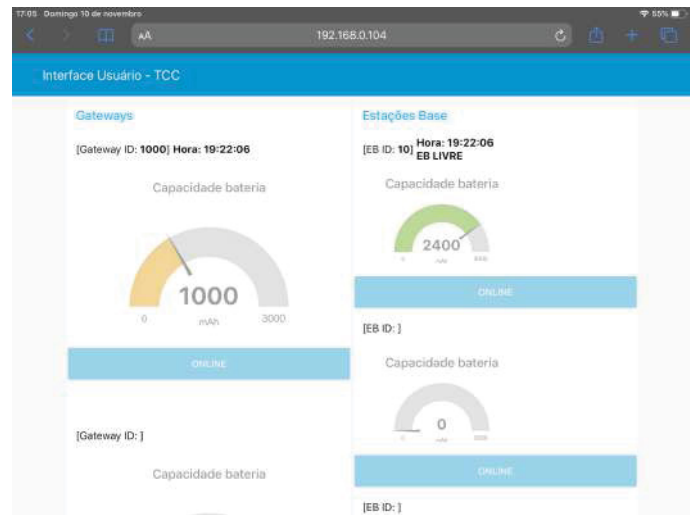


Fig. 6. Operator GUI.

paper. The LoRa technology has shown capable of going beyond the requirements needed, making the information exchange possible even on a range of 10 kilometers without compromising performance. The tests were performed on the real environment, in other words, in the rural area of the city of Renascença and in the urban area of the city of Francisco Beltrao.

The Heltec module was chosen for the tests because it posses all needed technology for promoting communication by using both LoRa protocol and MQTT, besides having a low energy cost [10]. The system was developed for countless base stations operations, being necessary the individual configuration of each base station via *gateways* coding, in order to define a ID for each station according to the user interface developed.

The implementation of the paper was facilitated due to many sources of information such as libraries for the utilized technology, like the Heltec and RFID modules. The use of the Node-RED was fundamental for the success of the project,

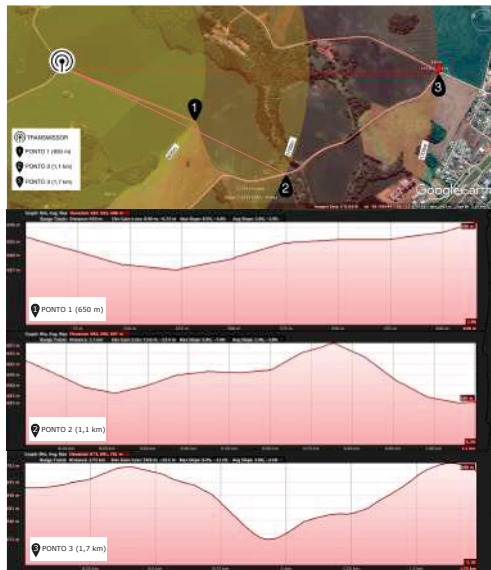


Fig. 7. Distance Test 1.

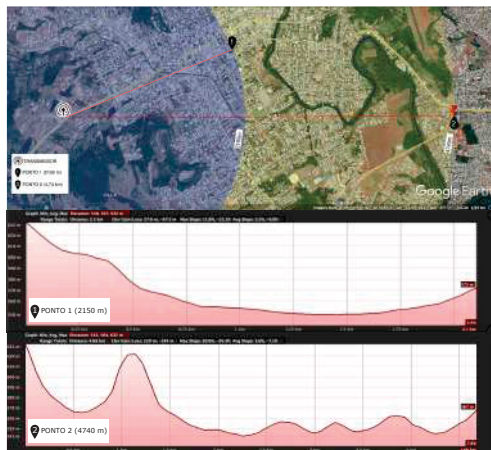


Fig. 8. Distance Test 2.

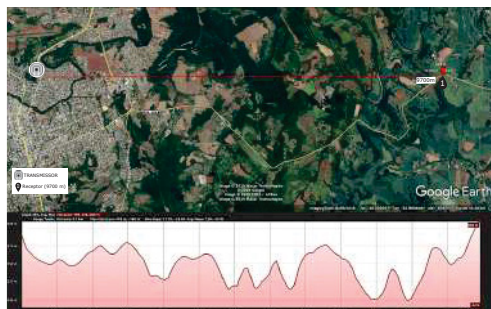


Fig. 9. Distance Test 3.

since the application allows the configuration of the MQTT communication between operator interface and devices, as well as implementing the own interface directly on the flow.

On the rate that the technology is advancing, in no time new technologies, libraries and tools will appear to smooth even more the way that LoRa protocol communicates, allowing

communication between devices optimize in a increasingly way, enabling reaching even longer distances.

The project developed proved promising, although some implementations need to be added in order to obtain the final product, such as: use of solar energy on base stations to supply both the microcontroller and the drone; if the base stations does not posses a Line of Sight (LoS), it may be necessary the use of **use a better antenna** or increase antenna's height; use of scheduled outages on the base stations; code aiming decrease their energy consumption; implementation of additional cryptography/security in all communication.

Besides the utility of the base stations for groups of drones, the project execution's result allows utilization beyond the propose initially given to the project, for instance, security area sensing, agricultural, from houses to a industry's facility.

#### REFERENCES

- [1] Estratégia nacional de ciência, tecnologia e inovação, 2016.
- [2] Jennifer Kite-Powell. Why precision agriculture will change how food is produced, 2018.
- [3] Eobin Alex George, Gaurav Tiwari, Rama N. Yadav, Edward Peters, and Srishti Sadana. Uav systems for parameter identification in agriculture. *2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS)*, pages 270–273, 2013.
- [4] Robison Cris Brito. Um modelo de otimização para planejamento dinâmico de voo para grupos de drones utilizando cooperação com bases de recargas móveis por meio de sistema multiagente e leilões recursivos. Doutorado em informática - ciências exatas, Universidade Federal do Paraná, Curitiba, 2018.
- [5] Robison Cris Brito, Jose Felipe Loureiro, Andre Guedes, and Eduardo Todt. Optimization system for dynamic flight planning for groups of drones using cooperation with mobile recharge bases by means of multiagent system and recursive auctions. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, July 2019.
- [6] Robison Cris Brito, Jose Felipe Loureiro, Carlos Eduardo Magrin, and Eduardo Todt. A model for landing, taking off and autonomous battery recharging of a parrot ar.drone 2.0 using computational vision and gps features. In *2019 Computer on the Beach*. COTB, fev 2019.
- [7] Lucas Puhl, Fabio Favarim, Vinicius Pegorini, and Carlos Torrico. Desenvolvimento de robô trekking autônomo integrando sensoriamento inercial, geoespacial e visual. In *2018 Computer on the Beach*. Meditec, Outubro 2018.
- [8] Robison Cris Brito, Nicollas Saque, Diego Addan Goncalves, Fabio Favarim, and Eduardo Todt. Optimized division of exploration areas in multi-robot systems considering static and dynamic charging stations. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, July 2019.
- [9] Moisés Meirelles Filho, Diego Addan Gonçalves, Robison Cris Brito, and Eduardo Todt. Sistema web para gerenciamento de vôos de aeronaves não tripuladas. In *Submitted to Medianeira in Technology 2018*. MEDITEC, nov 2018.
- [10] K. Mikhaylov, . Juha Petaejaejaervi, and T. Haenninen. Analysis of capacity and scalability of the lora low power wide area network technology. In *European Wireless 2016; 22th European Wireless Conference*, pages 1–6, May 2016.

# Evaluation of a Dynamic Optimization Method based on Recursive Auctions for Use in Flight planning of a Group of Drones applied to Precision Agriculture

Robison Cris Brito Federal University of Technology UTFPR robison@utfpr.edu.br	Beatriz Borsoi Federal University of Technology UTFPR beatriz@utfpr.edu.br	José Ricardo da Rocha Campos Federal University of Technology UTFPR jricardo28@gmail.com	Alfredo Weitzenfeld Univeristy of South Florida USF aweitzenfeld@usf.edu	Eduardo Todt Federal University of Paraná UFPR todt@inf.ufpr.br
--	--	--	--	---

**Abstract**—This study presents an algorithm to optimize flights of the unmanned aerial vehicle like a quadcopter, applied to precision agriculture. The proposed optimization algorithm can be distributed and can be used to maximize the number of points visited by the quadcopter, considering the limitation of its autonomy. This work starts by presenting how complex it is to solve the flight optimization of Drones, as its scenario is dynamic. Next, it shows some related papers found in the literature as the advantages of the proposed solution. These studies allowed us to define the research problem and offer an algorithm to solve it. Finally, tests were developed to evaluate the proposed algorithm in a simulation environment. The optimization using recursive auctions was tested in scenarios with 4, 9, and 16 points, and all of these, the optimal result was found. To minimize the processing time and the number of message exchanges among the agents, two heuristics were proposed. After applying the heuristics, the reduction of 99 % was achieved in the number of auctions in complex scenarios. In the end, it was realized tests with a real area with 16,000 square meters, with 1,156 points to be visited, and the optimal was found in seconds, showing this optimization method could be used in dynamic scenarios with several points to be visited.

**Keywords**—Planning, P-Space Optimization, Drone, Recursive Auction, Dynamic Environment

## I. INTRODUCTION

Drone, Quadricopter, Unmanned Aerial Vehicle (UAV), and Remotely Piloted Aircraft (RPAS) are some of the name variations for remotely controlled or autonomous unmanned aerial vehicles. The UAVs, to generalize the various denominations attributed to unmanned aerial vehicles, were originally used for war proposes. Still, with the evolution and accessibility of new technologies, today, its applications have been growing in the most several areas.

The importance of UAVs in precision agriculture is highlighted by authors such as [1] who register that UAVs provide the best platform to evaluate agricultural production, being fast and efficient compared to traditional terrestrial techniques, allowing reading data from all types of terrain. UAVs are versatile devices that can be used in a wide variety of agricultural applications, such as automatic monitoring of water in irrigation canals, detection and control of agricultural pests, and count of herd animals in the field.

From a technical point of view, currently, two types of UAVs are used in precision agriculture: the fixed-wing, which have the shape of a small airplane, known as Remotely Piloted Aircraft Systems (RPAS), and the rotative-wing, a vehicle like a helicopter with several propellers, most known as Drones.

Data from the systematic mapping developed by [2] indicate that 83% of the research involve fixed-wing UAV. This superiority is believed to be because they are the first unmanned aerial vehicles developed, as well as being applied to precision agriculture for many years, and also because of their autonomy, which is superior to the Drone.

However, this second has several other advantages, such as greater accuracy in reading the data, since they can fly more slowly without the risk of falling. They have greater ease of maneuver since unlike the fixed-wing, they can make sharp curves changing direction quickly, they also do not need the human interaction in the taking-off and landing, since the nature of these operations is vertical. When we use a Drone, it is possible to autonomous operations of landing, battery recharging, and taking-off [3].

A study presented in [4] made a comparison to identify the most favorable scenarios for the use of fixed-wing and Drones. It is concluded that for linear flights, for example, the monitoring of roads or borders, the superiority is held by fixed-wing vehicles. As for flights monitoring two-dimensional areas, where precision agriculture is commonly applied, the advantage is that of Drone, which can do the same work in a lesser time than fixed-wing due to its flight dynamic.

In this scenario, for the monitoring of vast terrain, which is common in precision farming, we can use a set of Drones, such as DJI Phantom, in place of a single fixed-wing. If they work in a coordinated and optimal manner, the area explored by these vehicles can be much larger than the area exploited by a single fixed-wing.

Still, we have enough technology that allows these Drones to fly autonomously, and human interaction with them is unnecessary, even in the most delicate operations such as landing, battery recharging, and taking-off. Also, while Drones are on the ground recharging its battery, it may be informed of its next mission, which can be centrally processed by a computer or IOT Device, that send the flight plans for them.

Thus, the biggest challenge is optimizing the flight plans of all Drones, as other tasks such as autonomous flight, communication between Drones, landing, recharging, and autonomous taking-off are operational activities that already present solutions in the literature. The optimization technique needs to consider that the flying occurs in a dynamic environment where weather can change quickly, like wind direction and speed, also could happen unforeseen operational events such as early battery termination or short battery life, or still, events due the dynamics of the environment, as a Drone or recharging bases could be included/removed at run-time.

From a computational point of view, a problem such this has a dynamic nature, which is formed by several waypoints, with many Drones and recharge bases is classified as a P-space complexity problem.

According to [5], P-Space is the set of all decision problems that can be solved by a Turing machine using a polynomial amount of memory space. While NP-type issues only need a polynomial amount of processing resources. So, in P-Space, we have to worry besides the amount of memory required to get solutions too.

We can make an analogy of this problem with the classic Traveling Salesmen Problem (TSP), but in the flight optimization with several Drones, we need to optimize the flight of all of them, and they need to pass by hundreds of way-points, also considering a dynamic environment, where the cost among the points can change given weather conditions, and in this scenario elements, like Drone and recharging station, can be included or removed dynamically in the environment. This feature classifying this problem as P-Space, that is harder to solve if compared with the classic TSP.

Given the complexity presented for the scenario, this paper proposes and evaluates an optimization algorithm based on recursive auctions, which supports the dynamics of problems such as UAV group flight optimization applied to precision agriculture. Being this exhaustively executed, the algorithm ensures that an optimal result is found. However, it requires a high computational effort, needing both processing and very high quantity of memory. Using heuristics in the proposed algorithm, it is possible to achieve feasible results, not necessarily the optimal, but good solutions with a relatively low computational cost, considering the dynamics of the environment.

To be able to test and evaluate this algorithm, the executions focused on the optimization itself, not on the dynamics of the environment, so some simplifications were applied to the scenario, being considered a single UAV, with a single recharging base, in a situation with static visiting points, as well as the cost to move among points is static, and considering that the UAV has sufficient autonomy to visit all the proposed points. Thus we can evaluate in this study, asymptotically, the behavior of this optimization algorithm. Issues such as how the algorithm works in dynamic environments will be addressed in future work.

## II. COMPUTATIONAL COMPLEXITY

Considering the computational problems that can be solved by algorithms, there are the so-called polynomial problems, or

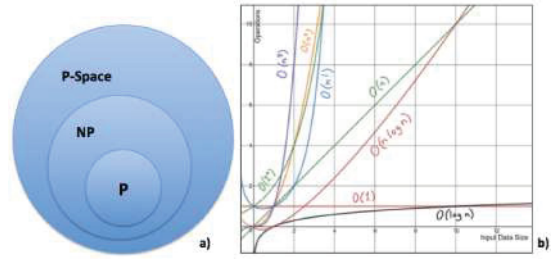


Fig. 1. a) Existing problem classes, where P problems are contained in NP, which in turn are contained in P-Spaces. b) Graph with the cost to solve several problems, wherein x-axis we have the number of input instances and in y-axis the complexity to solve the problem.

class P problems, which are problems that have an exact result and can be solved within an acceptable computational time. Examples of this problem are the location of data in a vector, or its ordering.

However, there is a much larger set of problems that are not contained in P, called Non-Polynomial Problems, or NP Problems, which do not have an exact solution and the only way to solve these problems is to explore all possible possibilities of solution, doing exhaustive processing, which is very costly from a processing standpoint. These are severe problems to solve, especially in instances with a lot of information. As an example of this, we have the traditional TSP [6].

But even more complicated than NP type problems are P-Space type problems, to which the present work is inserted. In this type of problem, in addition to the complexity of processing to find an optimal solution, we also have memory limitations to find this solution.

According to [7], P-type decision problems must be solved in polynomial time. In P-Space, decision problems must be solved in polynomial space of memory. As an example of P-Space, we have Facility Location Problem [8]. It studies the ideal allocation of warehouses to minimize product transportation costs, considering factors such as avoiding the placement of hazardous materials near homes or warehouses near product competitors.

In Figure 1.a) it is possible to check the classes of the existing problems, where the problems of type P are contained in the NP classes, as well as the NP are contained in P-Space, the latter being the one with the largest complexity shown in the figure.

To solve these problems, some algorithms are evaluated for their efficiency in resolution. Since it is difficult to make all algorithms run on the same computer for a fair comparison, their effectiveness is asymptotically analyzed, i.e., their efficiency in solving problems as the input instance is increased, which is the amount of information to be manipulated.

For example, a sequential search algorithm, which searches for an element within a vector, has a complexity  $O(n)$ , which is present in the graph in Figure 1.b). This complexity is linear, i.e., as elements are added to a vector (x-axis), the computational effort to perform search operations (y-axis) increases proportionally, regardless of the computer or instance in which this algorithm runs, its complexity will always be

the same. This type of analysis is called asymptotic analysis. We have other examples, such as binary search, wherein this the complexity is  $O(\log n)$ , which means that as the number of elements in the vector increases, the cost of operations does not increase in the same proportion. Vector ordering algorithms, for example, have greater complexity, like the bubble algorithm, with complexity  $O(n^2)$ . The sort algorithm called Merge Sort has complexity  $n \log n$ .

In the graph of Figure 1.b), we see that the most efficient algorithms tend to approach the x-axis, while the less efficient algorithms approach the y-axis.

All the algorithms presented in the previous paragraphs solve polynomial problems, that is, P-type problems, finding exact solutions in an acceptable computational time, which is good. However, NP class problems only have optimal results if the algorithms are executed exhaustively, and these algorithms are usually very close to the y-axis.

An example of an NP class problem is the Hamiltonian circuit problem [9]. This problem is characterized by having a graph, and we need to know if it is possible to visit all points of this graph, passing each point only once, and returning to the starting point. To make this possible, we must test all possible path combinations, which makes the complexity of this algorithm  $n$  factorial ( $n!$ ). With a small scenario with few points, it is possible to find optimal solutions in an acceptable computational time, but when the number of points increases, the computational complexity increases, so the computational cost to find solutions in this type of problem is very close to the y-axis, as seen in Figure 1.b).

However, this is not yet the worst of the computational problems. If we consider that to solve the TSP, it is necessary to find all the Hamiltonian circuits of the graph, to compute the cost of each path, to compare them so only to know the shortest path. We can say that the computational complexity of the TSP is greater than the computational complexity of solving the Hamiltonian path problem.

And finally, in the P-Space type problem, such as optimized UAV group flight planning, which is the approach used in this paper, the complexity is even greater because there are so many Drones, so many way-points, there are countless recharging bases, and there is also the dynamics of the environment, such as changing the cost to visit each point, which occurs when wind speed or direction changes, as well as adverse events such as the early termination of a Drone battery.

Thus, despite the complexity and characteristics of the present problem, the optimization algorithm used in this problem must be fast and dynamic, generating flight plans for UAVs quickly, even if they are not the best possible routes. Still, they are close to optimal solutions, since, for every change to the environment, new flight plans must be generated and sent to the UAVs.

Unlike the exhaustive-running algorithms used to solve the TSP, for example, which can run for many hours for not very large instances, here we need an algorithm that runs in seconds even on very large instances, something that is very common when you need optimization in scenarios involving IoT or precision agriculture, for example.

To obtain possible results, not necessarily the best of all, but acceptable results with little computational cost, it is common to use heuristics and meta-heuristics to solve NP or even P-SPACE problems.

An example of a heuristic used to solve the TSP, with a minimal computational cost compared to the traditional algorithms that find the optimal solution, is the Greedy algorithm [10].

This heuristic defines that, whenever the salesman needs to choose a city, the decision will be the city with the lowest cost, being executed until all cities are visited. This heuristic does not guarantee the optimal solution. Still, it does guarantee a single feasible solution in an acceptable computational time, but if the algorithm is executed countless times, all of them will return the same answer, since it is a deterministic algorithm.

To improve the results obtained by deterministic heuristics, such as the Greedy algorithm, meta-heuristics are used, which have random parameters, making that each execution, for example, has different results, and the best result is chosen at the end. In meta-heuristics, algorithms are executed until a stop condition is met, which may be several execution cycles or a timeout.

A known meta-heuristic for solving the TSP is the Greedy Randomized Adaptive Search Procedure (GRASP) [11]. This is based on the Greedy algorithm, however, for each city the algorithm chooses the  $n$  nearest cities, this being  $n$  defined by the programmer, raffling among them (this is the algorithm's random parameter) one of the nearest cities to visit, thus, if you run the algorithm several times, probably all the results will be different, and the best result presented will be chosen at the end.

Thus, the present study presents an algorithm to find quick solutions to the flight optimization problem for UAV groups. This is a P-space type problem. This algorithm has high computational complexity (near the y-axis), and being executed exhaustively ensures the optimal global meeting using a lot of resources such as processing and memory, which makes exhaustive execution for dynamic scenarios disproportionate. To this algorithm can be applied heuristics and meta-heuristics, which allow to find feasible solutions in acceptable processing time, considering that for the precision agriculture scenario, in which changes in the scenario happen very quickly, an algorithm is needed to adapt very quickly to this change, optimizing the flight in the same way, but not focusing on the optimal result, but on a feasible result given the time between one change in environment and another.

### III. RELATED WORKS

A Systematic Mapping Study was conducted to identify research that works with the optimization of flights of multiple aerial vehicles to highlight how the research is being done in this scenario [2].

One of the most significant researches was developed by [12]. This author worked with the cooperation of two fixed-wing UAVs applied in precision agriculture. The author divides the problem into two parts: first, he uses the principle of particle flow to make fixed-wing attracted to the way-points. It then uses an optimization algorithm developed in Mixed



Integer Linear Programming (MILP) to perform the flight optimization of the two UAVs since the way-points, the landing and, the taking-off locations are known. Among the limitations of the study is the fact that it is developed for only two fixed-wing, besides being a static model, so, before the flight is executed a MILP algorithm that generates the flight plans, which will later be performed by the UAVs.

According to the author, the number of two fixed-wing is ideal for cooperative flights in precision agriculture. Fixed-wing flights have a high cost to set up the aircraft, as well as the need for taking-off and landing intervention. Increasing the number of fixed-wing also increases the number of system operators.

Still according to the author, the justification for using fixed-wing on cooperative flights applied to precision agriculture is because most of the scientific work using this type of vehicle, and, also, fixed-wing have more autonomy than Drones.

Although the author highlights in his work the limitations of the fixed-wing use, such as restrictions related to the minimum speed of the aircraft, it can not be below a specific limit to avoid the risk of falling and the non-detailed reading of data by high flight speed.

Another work aimed at the cooperation of Drone is developed by [13]. It aims to make four Drones perform the same task: the transport of a sphere on a plate and the UAVs must identify the position of the sphere on a plate and perform movements to keep it balanced. Since all airborne devices have the same algorithm, when the ball position changes on the plate, all devices perform behavioral changes in a coordinated manner. For information processing, an artificial neural network algorithm was used. Initially, the UAVs were trained for all possible situations.

Optimization through the use of an artificial neural network brings high-speed processing since the longest phase of network training. However, this training allows great results for systems with few variables. For systems with many possibilities, optimization through a neural network may not bring good results.

Another work of significant contribution was published by [14]. It presented a fixed-wing multiple flight model, tested in a simulation environment. Although simulated, this work considers issues such as angles and minimum speed for the fixed-wing to perform maneuvers.

As a limitation presented is the system is not able to find the best possible solution for the scenario, identifying only a feasible solution given the computational complexity for it.

[15] presents in his work a model for optimizing fixed-wing flights, applied to data collection from wireless sensor networks. The application scenario is interesting because, in wireless networks, one of the highest costs is data transmission between network nodes. To minimize this cost, a fixed-wing flight optimization system is proposed using the wireless sensor network topology, and the sensors are distributed in *cluster*, with the nearby sensors sending all information to a node called *head* of the *cluster*. This node centralizes all information and forwards it to the fixed-wing as soon as it flies

over that point (cluster *head* node). Battery expenditures for information exchange within the network are compared to the traditional approach, called *Low Energy Adaptive Clustering Hierarchy* (LEACH), and the fixed-wing model showed overall system battery savings.

To optimize the fixed-wing flight that collects information, [15] used the *Particle Swarm Optimization* (PSO), applying a classic algorithm used to solve the *Traveling Salesman Problem*(TPS), which is not about dynamics, not even multiple fixed-wing. However, the author points out in his conclusion that one of the future works is to present techniques to treat data collection with many fixed-wing.

[16] presents a fixed-wing group flight optimization model from a PSO and AG hybrid algorithm. This paper considers the optimization of fixed-wing paths in a three-dimensional scenario. One of the contributions of this work is that processing is performed in parallel, optimizing the use of multi-core processors. Another contribution is the use of multi-purpose function, allowing us to optimize more than one system parameter. This work was also tested in the simulation environment.

In general, it is possible to observe that no much scientific papers deal with Drones flight optimization. Also, it is possible to verify that no article studied deals with the dynamics of the environment, a widespread problem faced in real flight environments. It is believed to be because most works deal with pre-optimization rather than optimization during system execution.

#### IV. PROPOSED SYSTEM

Although several preliminary studies have been developed and showed the viability of a system that deals with the execution of monitoring missions formed by several Drones, a research topic is still little explored: a faster flight optimization considering the opening of the system and dynamicity of the environment. In the studies showed in the last session, we have a couple of works that use classic algorithms of optimization, and, almost all, need of them needs a lot of time for training or processing, so it can not be used in a dynamic environment like flight optimization of Drones applied in the precision agriculture. As an example of these algorithms, we can highlight the PLIM, Neural Network, and PSO.

The opening is the term used to inform that the system can receive or lose components in the execution time; these components could be Drones or base. Dynamicity means that the cost between points can change over time, with the change of wind speed and direction

To be possible for such optimization in a dynamic environment in which Drone group flight is applied to precision agriculture, it is necessary to choose and test a fast, dynamic optimization algorithm that can be easily distributed on different processors. The proposed algorithm for this work is the recursive auction, presented in detail in this section, and evaluated in the following session.

This algorithm is relatively simple to apply in distributed computational systems from the FIPA Contract-Net protocol and allows the communication within the system regardless of the number of elements that may vary over time. According to

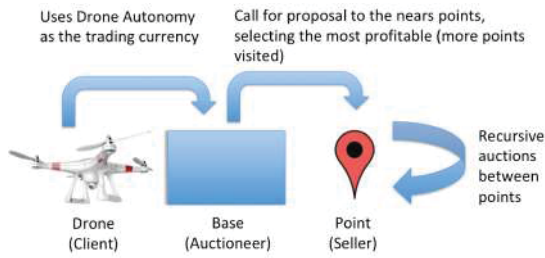


Fig. 2. Analogy of the system based on recursive auctions to a traditional auction.

[17], auction-based trading methods are often used to perform the distribution of tasks assigned to teams with dynamic elements.

Analyzing the problem in which the proposed system applies, it is possible to observe an analogy to a traditional auction - Figure 4, in which the Drone would be the consumers, and have as their trading currency their autonomy. They share this information with the bases, as known as base, which have the role of auctioneer that offers these resources to those interested by auction, which in this analogy are the points that wish to be visited. Each point receives the proposal for the Drone's visit, and the applicant responds to the cost to be visited, as well as how many unique points will be visited on the same flight. The best proposal is chosen based on the desired objective function. In the proposed system, the objective function is to maximize the number of visit points in a single flight.

If we run the auctions exhaustively, testing all possible possibilities, we can find the best result for the scenario, but in this situation, we have the same problem as the classic algorithms proposed to find solutions to the TPS problem, for example: time processing is disproportional, so in a scenario with a few dozen points, a normal computer cannot find results in a feasible time.

To decrease the computational complexity, heuristics can be used to reduce the processing time without significantly compromising the result found, as well as meta-heuristics, can also be applied to heuristics, allowing to considerably improve results without significantly compromising processing time.

Remember that N-Space problems are very complex to solve optimally with processing time and a proportional amount of memory. Therefore, for the scenario used, it is necessary to focus on the good results obtained quickly. It is more important than finding the optimal result. This is the scope of the proposed optimization algorithm based on recursive auctions.

In order to evaluate the processing time and the quality of the responses of the present algorithm, some questions regarding the environment openness and dynamics will be disregarded in these tests.

Thus, some simplifications were made in the scenario, being the tests using a single Drone, with a single recharging base, in a scenario with static visiting points, as well as the weight of static edges, and considering that the Drone has sufficient autonomy to visit all points proposed. As the algorithm

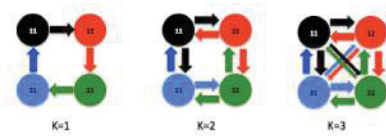


Fig. 3. 4 point scenario, with  $k$  varying from 1 to 3.

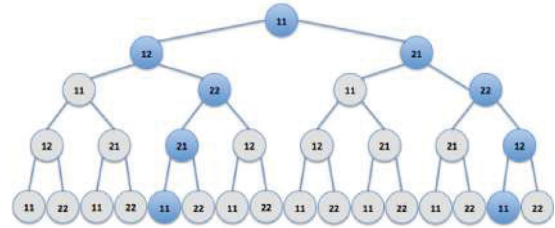


Fig. 4. Graph with the possibilities of displacement of a Drone considering the initial position the vertex 11,  $k = 2$  and Drone autonomy of 4

was already designed to be used in dynamic environments, in a second moment, it can be easily introduced to a multiagent system for system openness and dynamics tests, but at the moment the objective is to test the efficiency, the speed and the quality of the responses obtained.

A system constant that directly influences response quality, as well as processing time, is the  $k$  constant, which represents the number of neighboring points that a given point can visit and, consequently, the number of auctions that the point will make. If we use  $k$  with small values, results are found at a shorter processing time, but if we use  $k$  larger, better results tend to be found, but with longer processing time. Figure 3 presents examples using  $k = 1$ ,  $k = 2$  and  $k = 3$  for a 4-point scenario.

Considering this same scenario with four points to be visited, and with a value of  $k$  equal to two, and considering that the base is in point 11 of the Figure, we will have a graph representing the possible visits of the Drone according to Figure 4.

Note that the height of the tree is the autonomy of the Drone, that is, the number of points it can visit. For a scenario with 4 points to be visited: 11, 12, 21, and 22, it was required 30 auctions (graph edges). For this test, two optimal global solutions (darker points) were found, which ensure that the Drone passes through all points and return to the starting point.

For the recursive auction of the presented graph, it is considered that the base is at point 11, just like the Drone. Thus, it requests the two nearest neighboring points proposals for the Drone visit. These points respond with the Drone's travel cost to them, as well as the Drone's return cost to the base.

Drone being left autonomous, each of these neighboring points starts a new round of auctions, asking their  $k$  neighbors for new proposals for Drone's visit, with the most lucrative bid being chosen (lowest cost with the largest number of unique points visited) and returned to the point you requested. This sequence of auctions happens exhaustively as long as there is autonomy in the Drone, which ensures that the optimal solution is found.

This process is done by considering the starting point of each base point of the scenario, finding the best solution for each point, as well as the best position from point to base. Regarding the number of auctions executed, for the scenario presented (Figure 4), 30 auctions are held for each point chosen as the starting point of the Drone, as there are 4 points, in total, there will be 120 auctions for this scenario.

Thus, one of the heuristics adopted to decrease the number of auctions is to choose, by another algorithm or technique, a point to be the starting point of the Drone, and there is no need to test all possibilities, for this purpose. In the 4-point scenario, for example, there is a 75 % decrease in the number of auctions after applied this heuristic, and this decrease difference is even greater for larger scenarios.

Since the goal is to make an optimization in which the Drone must visit each point once, another time-optimizing heuristic can be adopted: to make each visited point not receive new requests for visits, that is, a great pruning using this one heuristic happens, as shown in the dark circle - Figure 4.

Initially, before the heuristic that prevents a point from being visited twice, there were 30 auctions, after the application of this heuristic, the number of auctions fell to 8, a reduction of 73.33 %, and for larger scenarios, the reduction in a number of auctions is even higher.

The next session will present the result of applying the auction-based optimization algorithm features for some pre-established scenarios, with performance testing.

## V. RESULTS

To test the algorithm, a Java SE application was developed to check message exchanges as well as processing time. The experiments were performed on a 2.9 GHz Core i5 processor laptop computer with 8GB of memory.

Three scenarios were used to verify the behavior of the algorithm: 4, 9, and 16 visitation points. Finally, the algorithm was tested in a real scenario that had 1,156 points.

For tests with 4, 9, and 16 points, a value of  $k = 2$  was used, which is the smallest possible value. Since the objective is to test the performance and the increase in the number of auctions, as well as the processing time, the value of  $k$  will not influence such analysis. For the last test, with a real scenario, a different  $k$  value was used to improve the results, since the number of points used was much higher.

The 4, 9, and 16-point performance tests were performed at three distinct times: first by running exhaustively the recursive auction-based optimization algorithm thoroughly, testing all possible possibilities for the recharge base, and flying overall points, including more than once the same point, having as its end condition the autonomy of the UAV, which is equal to the number of points of the scenario. With this execution, the overall optimum is found. However, the number of auctions and the processing time is longer. This run was named Original Model. The results are shown on Table I.

It can be seen in the "N. Auctions Original" column that numbers increase exponentially, given the complexity of this algorithm, which is equivalent to traditional algorithms for

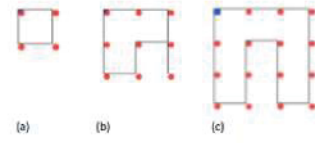


Fig. 5. First result obtained for the scenario with 4 points (a), 9 points (b) and 16 points (c), in all, the base is at Point 1 and the flight happens clockwise.

finding Hamiltonian Circuits (NP-type problems). This exponential increase in the number of messages is not good, which demonstrates the need to use heuristics to obtain equivalent results in shorter computational time.

For all scenarios, all possible positions for the base are tested to obtain the best recharge base placement. For the 4-point scenario, there are 8 possible optimum possibilities, it was required 120 auctions and using a processing time of 32 ms.

The same test was performed for 9 and 16 point scenarios. Thus, in the asymptotic analysis of the original model, to know the amount of auctions that will happen is used Equation (1), and the number of auctions depends on the number of neighbors that the points will have ( $k$ ), as well as the autonomy of the Drone ( $n$ ).

$$n \cdot \sum_{i=1}^n k^i \quad (1)$$

For each scenario, one of the optimal results was chosen to be represented in a graphical interface developed in Canvas with Java SE. The results are shown in Figure 5.

To improve processing time as well as decrease the number of auctions without significantly compromising the algorithm results, two heuristics were proposed.

The first heuristic, called h1, defines a starting point for the recharge base, not requiring all points to be tested as a possible starting point. Studies such as [18] present algorithms for defining a starting point for an area.

After applying the h1 heuristics, the number of auctions decreased considerably. In the Table I - Column "N. Auctions after h1", there is a significant decrease in the number of auctions held, which varies by a reduction of 75% for the 4-point scenario and reaches 93.75 % in a 16-point scenario, so the larger the scenario, the larger the optimization over the original model.

In asymptotic analysis, the complexity of the algorithm after using h1 is as presented in Equation (2).

$$\sum_{i=1}^n k^i \quad (2)$$

Using the first heuristic, the number of messages decreases considerably, but the increase is still exponential, which shows

TABLE I. RESULTS OF THE EXECUTION OF THE ALGORITHM OF OPTIMIZATION WITH RECURSIVE AUCTIONS FOR SCENARIOS WITH 4, 9 AND 16 POINTS.

Scenario	N. of Opt. Sol. Found	N. Auctions Original	N. Auctions after h1	N. Auctions after h2
4 points	8	120	30	8
9 points	234	9198	1022	18
16 points	2.432	2242784	140174	683

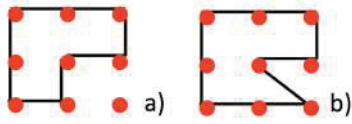


Fig. 6. Result for 9 point scenario. Left using  $k = 2$  and right using  $k = 4$ .

that the algorithm still does not perform well for large scenarios.

In order to improve the algorithm, a second heuristic is proposed, called h2, in which each point should be visited only once, and the results compared with the execution after h1, so the result is presented in Table I - Column "N. Auctions after h2".

For larger scenarios, such as 16 points, the reduction in the number of auctions has exceeded 99%, if it is compared with the original model. The formula representing the number of auctions after applying the heuristic h3 is presented in Equation (3). It is for the best situation, where we have a square area with  $k=2$ .

$$k.n.\log_k(n - k) \quad (3)$$

For the original model, 120 auctions were needed to achieve the global optimum in the 4-point scenario. After applying the heuristics, the optimum was achieved with only eight auctions.

Regarding the overall optimum, for the scenario with 4 and 16 points, this optimum was found, since they are scenarios that form a Hamiltonian circuit. For the 9 point scenario, the solution found allowed to pass only 8 points, since that on the heuristic h2 two points cannot be visited on the same flight and the value of  $k = 2$  prevents this optimum from being reached, thus the result obtained after applying h1 and h2 is shown in Figure 6.a), in this case it was necessary to change the value from  $k$  to 4 to go through all points as shown in Figure 6.b).

Thus, it is possible to verify that the value of  $k$  is an important factor for the quality of the scenario response, but it influences the number of messages considerably.

For nearest neighbors, as these are physical positions, close neighbors can be defined based on the actual distances between these points.

In relation to  $k$ , it is important to identify the shape of the area that will be mapped in points because depending on the geometric shape, different values of  $k$  can be used. Studies are being conducted to identify the best value of  $k$ , however, for the work in question,  $k = 2$  was used for the asymptotic analysis of the algorithm considering small scenarios, but in

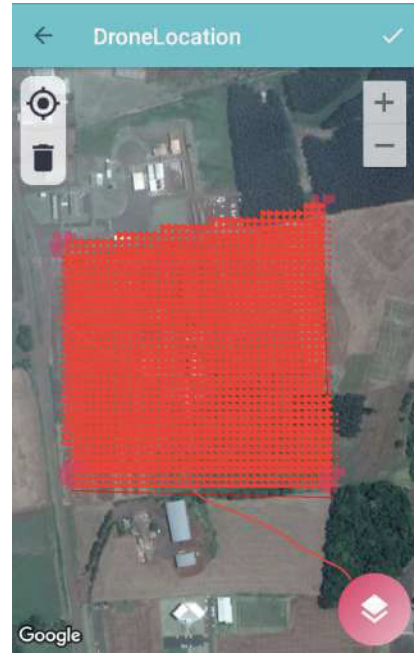


Fig. 7. The App Screen with all point.

real scenarios, the  $k = 4$  is the best to find good results in not so long processing time [19].

For the test in the real scenario, a simulation was used considering the survey of a Digital Elevation Model of the area [20]. Today, this process is done with satellite images, with a resolution of 30 meters between one point and another. For very rough terrain, this accuracy of 30 meters is not sufficient, and for higher accuracy with readings every 10 meters, for example, mathematical interpolation between points is required to infer the measurement every 10 meters, however, there is no guarantee of the quality of the data obtained.

For the test in the real scenario, a resolution of 10 meters was used between one point and another, providing an overflight for the realization of a digital elevation model. The area used for simulation is the experimental area of the university where this thesis research was performed, which has 16,000 square meters, or 16 hectares. The selected area has 340.0m x 340.0m, using 10 meters of precision, we need 34 x 34 points, totaling 1,156 points at all.

We used a program to delimit this area and put all the points with 10 meters of precision. The screen of this application is shown at the Figure 7. This application was developed using Android platform, that has limited processing resource as comparable with a regular computer, and the application can run without problem. The result of this algorithm when executed in an Android app is presented in [21].

Considering these 1,156 points and using 4 neighbors for each point, we used the optimization algorithm based in the recursive auction to return the best way to pass in all of this 1,156 points only one time. After 27 seconds of processing, the algorithm found the first optimal situation, showed at the Figure 8.

The be easy to see the way we put in a simulation

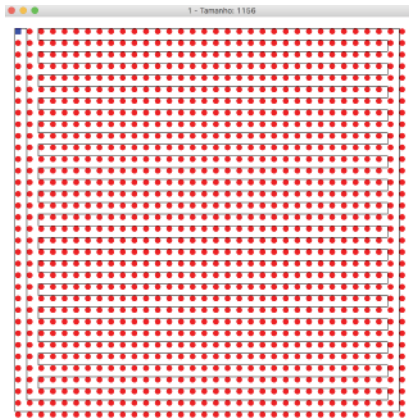


Fig. 8. Application Screen that selected, after 27 seconds of processing, a way to pass one time in each point in the selected area.

environment, without background map, all the points, and was traced the UAV path. How we have a lot of point in the map, probably we have a lot of option of way to the drone pass in all points, and probably the algorithm will spend a lot of time to returns all of the results, but, for our problem, we need quicker answers, because it, we use only the first best result.

But, probably, an area with this size could be overflowed for more than one Phantom, thus, the area necessary for each optimization could be smaller, and the processing time could be quicker.

## VI. CONCLUSION

The present study proposed and evaluated an algorithm for optimization Drones flights applied to precision agriculture.

Since there is a need for a dynamic optimization algorithm that allows for such optimization even with varying environmental data and returns results relatively quickly without training, as is common with other existing optimization techniques, an algorithm based on recursive auctions was proposed as well as evaluated in order to test its behavior asymptotically and in a real environment.

Throughout the article, it was presented that the problem in question is a complex P-Space problem, more complex than the traditional Traveling Salesman problem. However, the proposed algorithm presented relatively good results in short processing time.

The algorithm of optimization based on recursive auctions was tested in an application developed in Java SE, with a graphical interface in Canvas, in which it was possible to test the behavior of this algorithm in different scenarios. The algorithm was stable, obtaining optimum results for all the scenarios tested, regardless of the number of points to be visited and the Drone's autonomy.

As the number of auctions tends to increase exponentially as the number of points to be visited increases, two distinct but simple heuristics have been proposed. They have proved to be very efficient, as they have reduced the number of auctions carried out by up to 99% in scenarios with more points of passage. However, it was observed that in some scenarios, it

was not possible to go through all the proposed points, which are believed to be solved with the inclusion of new heuristics.

In real scenarios, as presented in the last case study, the algorithm was able to find a route considering many points, which is not commonly the case with a drone flight, as the flights appear to be shorter because of the Drone's limited autonomy, however, even over a larger land area, the time and route returned was acceptable for rapid drone flight planning as it is a dynamic environment.

As future work, the algorithm will be distributed to different software agents to test the algorithm scalability and dynamics in distributed systems, as well as the flight plans generated in a developed simulation environment based on QGroundControl.

In the future, the proposed model based on Multiagent System and recursive auctions will also be tested in other scenarios, such as the optimization of the use of urban electric vehicle batteries, in order to test its instanciability.

## REFERENCES

- [1] Eobin Alex George, Gaurav Tiwari, R N Yadav, Edward Peters, and Srishti Sadana. UAV systems for parameter identification in agriculture. In *2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS)*. Institute of Electrical & Electronics Engineers (IEEE), aug 2013.
- [2] Robison Cris Brito, Jose Felipe Loureiro, Eduardo Todt, and Roberto Pereira. A systematic mapping for the scenario of non-urban autonomous vehicle cooperation systems. In *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*. IEEE, nov 2017.
- [3] Min-Fan Ricky Lee, Shun-Feng Su, Jie-Wei Eric Yeah, Husan-Ming Huang, and Jonathan Chen. Autonomous landing system for aerial mobile robot cooperation. In *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*. IEEE, dec 2014.
- [4] Mainara Cristina Lorencena, Robison Cris Brito, Jose Felipe Loureiro, Fabio Favarim, and Eduardo Todt. A comparative approach on the use of unmanned aerial vehicles kind of fixed-wing and rotative wing applied to the precision agriculture scenario. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, July 2019.
- [5] Michel Sipser. *Introdução a Teoria da Computação*. Thomson, São Paulo, 2nd. edition, 2007.
- [6] Wikipédia. Travelling salesman problem, 2019. [Online; Acessado em 09 de abril de 2019].
- [7] Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson, mar 2005.
- [8] Wikipédia. Facility location problem, 2019. [Online; Acessado em 09 de abril de 2019].
- [9] Wikipédia. Hamiltonian path, 2019. [Online; Acessado em 09 de abril de 2019].
- [10] Thomas H. Cormen. *Introduction to Algorithms*. MIT Press, Massachusetts, 2009.
- [11] Thomas A. Feo and Mauricio G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, March 1995.
- [12] Gustavo Avellar, Guilherme Pereira, Luciano Pimenta, and Paulo Iscold. Multi-UAV routing for area coverage and remote sensing with minimum time. *Sensors*, 15(11):27783–27803, nov 2015.
- [13] Gerard Ely U. Faelden, Jose Martin Z. Maningo, Reiichiro Christian S. Nakano, Argel A. Bandala, and Elmer P. Dadios. A neural network approach to a cooperative balancing problem in quadrotor-unmanned aerial vehicles (QUAVs). In *2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. Institute of Electrical & Electronics Engineers (IEEE), dec 2015.

- [14] Halit Ergezer and Kemal Leblebicioğlu. 3d path planning for multiple UAVs for maximum information collection. *Journal of Intelligent & Robotic Systems*, 73(1-4):737–762, oct 2013.
- [15] Dac-Tu Ho, Esten Ingar Grotli, P. B. Sujit, Tor Arne Johansen, and Joao Borges Sousa. Cluster-based communication topology selection and UAV path planning in wireless sensor networks. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, may 2013.
- [16] Vincent Roberge, Mohammed Tarbouchi, and François Allaire. Parallel Hybrid Metaheuristic on shared Memory System for real-time UAV Path Planning. *International Journal of Computational Intelligence and Applications*, 13(02):1450008, jun 2014.
- [17] Charles E. Pippin and Henrik Christensen. Cooperation based dynamic team formation in multi-agent auctions. In *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III*. SPIE, may 2012.
- [18] Robison Cris Brito, Nicollas Saque, Diego Addan Gonçalves, and Eduardo Todt. Optimized division of exploration areas in multi-robot systems considering static and dynamic charging stations. In *Submitted to 2018 Latin American Robotics Symposium (LARS) and 2018 Brazilian Symposium on Robotics (SBR)*. IEEE, nov 2018.
- [19] Robison Cris Brito, Jose Felipe Loureiro, and Eduardo Todt. Choosing the best k-neighbor parameters for optimizing a flight of a group of drone using recursive auction algorithm applied to the precision agriculture. In *B-Sides Tampa 2020*. B-Sides Tampa 2020, feb 2020.
- [20] Embrapa. Programa nacional de solos do brasil (pronasolos). *Embrapa Solos*, nov 2016.
- [21] Andre Rabello, Robison Cris Brito, Fabio Favarim, Alfredo Weitzenfeld, and Eduardo Todt. Mobile system for optimized planning to drone flight applied to the precision agriculture. In *Information and Computer Technologies (ICICT 2020)*. Information and Computer Technologies (ICICT 2020), mar 2020.

# Greedy Optimization applied on Group Drones Flights: A Event-based Approach

Robison Cris Brito Federal University of Technology UTFPR robison@utfpr.edu.br	André Guedes Federal University of Parana UFPR alpguedes@gmail.com	Fatemeh Afghah Northern Arizona University NAU fatemeh.afghah@nau.edu	Alfredo Weitzenfeld Univeristy of South Florida USF aweitzenfeld@usf.edu	Eduardo Todt Federal University of Paraná UFPR todt@inf.ufpr.br
--	--	---	--	---

**Abstract**—This study presents an algorithm to optimize flights of the Unmanned Aerial Vehicle like a Drone, applied to precision agriculture. The proposed optimization algorithm can be distributed and can be used to maximize the number of points visited by the Drone, considering the limitation of its autonomy. This work starts by presenting a dynamic environment. Next, it shows some related papers found in the literature as the advantages of the proposed solution. These studies allowed us to define the research problem and offer an algorithm to solve it. Finally, tests were developed to evaluate the proposed algorithm based on Greedy Optimization in a simulation environment. All of these results were obtained in less than 1 second of processing, that shows how quick is the Greedy Optimization for Dynamic environment. The optimization was tested in scenarios with 121 and 1,156 waypoints, with 1 and 4 Drones. All results are compared with the best flight plan found for each scenario. In the scenario with 121 points, the result obtained using Greedy Optimization is only 15% worst if compared with the best result. In the scenario with 1,156 points, the result was 7% worst.

**Keywords**—Planning, Optimization, Drone, Greedy, Dynamic Environment

## I. INTRODUCTION

According to [1], the concept of precision agriculture is usually associated with the use of high-tech equipment to assess or monitor conditions in a given area. With the data read from this kind of equipment, we can apply the various factors of production (seeds, fertilizers, water, etc.). Thus, based on the specific data from each area, the agricultural automation process is implemented, dosing inputs proportionally, different from the traditional agriculture, in which the same amount of inputs is used throughout all areas.

The importance of UAVs (Unmanned Aerial Vehicle) in precision agriculture is highlighted by authors such [2], who wrote that the UAVs provide the best platform for assessing agricultural production, being fast and efficient compared to traditional terrestrial techniques, and it allows to reading data on all types of terrain. [3] complements, saying that UAVs are versatile devices that can be used in the most varied agricultural applications, such as automatic monitoring of plantations and water in irrigation channels, pest detection and control, and in livestock, with the count of animals in the pasture.

Still, according to [3], it is in autonomous flights that UAVs highlighted in the agricultural environment since they

are frequently used in the monitoring and inspection of large plant areas. [4], usually to use autonomous UAVs to monitoring plant growth. It can read data and, through an algorithm, this data is classified, without human interaction. With autonomous flights and processing on the device itself, the amount of data processed, such as the area analyzed, can be larger.

From a technical point of view, currently, two types of UAVs are used in precision agriculture: the fixed-wing, which have the shape of a small airplane, known as Remotely Piloted Aircraft Systems (RPAS), and the rotative-wing, a vehicle like a helicopter with several propellers, most known as Drones.

A study presented in [5] made a comparison to identify the most favorable scenarios for the use of fixed-wing and Drones. It is concluded that for linear flights, for example, the monitoring of roads or borders, the superiority is held by fixed-wing vehicles. As for flights monitoring two-dimensional areas, where precision agriculture is commonly applied, the advantage is that of Drone, which can do the same work in a lesser time than fixed-wing due to its flight dynamic.

In this scenario, for the monitoring of vast terrain, which is common in precision farming, we can use a set of Drones, such as DJI Phantom, in place of a single fixed-wing. If they work in a coordinated and optimal manner, the area explored by these vehicles can be much larger than the area exploited by a single fixed-wing.

From a financial point of view, with the price of a fixed-wing, such as Wingo, from UAVision Aeronautic, which costs around \$ 150,000.00, it is possible to buy a massive swarm of Drones Phantom, from DJI, which costs about \$ 500.00.

Still, we have enough technology that allows these Drones to fly autonomously, and human interaction with them is unnecessary, even in the most delicate operations such as landing, battery recharging, and taking-off. Also, while Drones are on the ground recharging its battery, it may be informed of its next mission, which can be centrally processed by a computer or IoT Device that sends the flight plans for them.

Thus, the biggest challenge is optimizing the flight plans of all Drones, as other tasks such as autonomous flight, communication between Drones, landing, recharging, and autonomous taking-off are operational activities that already present solutions in the literature. The optimization technique needs to consider that the flying occurs in a dynamic environment where the weather can change quickly, like wind direction and speed,

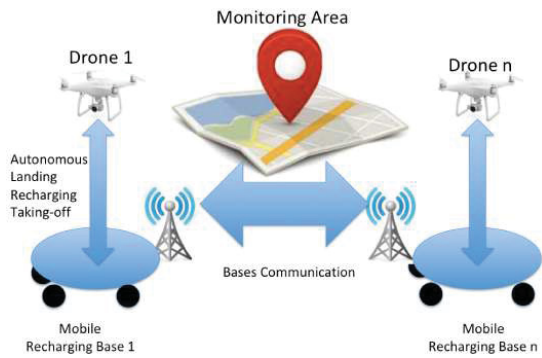


Fig. 1. Overview of a flight optimization system for multiple Drones.

also could happen unforeseen operational events such as early battery termination or short battery life, or still, events due to the dynamics of the environment, as a Drone or recharging bases could be included/removed at run-time.

This paper is part of a whole project, where propose an architecture, this presented in the Figure 1.

As shown in Figure, this architecture proposes the use of multiple Drones working cooperatively, applied to precision farming to perform missions that escape the Drone's capabilities. Precision farming was chosen because of its characteristics, as flights must be slow for a detailed reading of the terrain.

It is proposed as an event-based communication model. To optimize Drone battery usage, the biggest limitation of the system, communication only happens between the charging bases. The Drone battery is only used for the flight. When the Drone returns to the base, it feeds the system with the information retrieved during the flight, and the system defines the new flight plans based on this information for all other Drones. As the Drone recharges its battery, the land vehicle moves to a new strategic position for the Drone to begin the new mission.

For the optimization of the Drone flight plan, we are using an greed optimization algorithm, that was based on the algorithm proposed to [6], that originally was used for routing in Dynamic UAV Network. In this work, it was adapted for the scenario where a land need to be overflowed for a group of several Drones, this applied in precision farms. This algorithm is adapted and tested, to measure how quicker it is, and, how good is its results compared with the optimal result.

## II. RELATED WORKS

A Systematic Mapping Study was conducted to identify research that works with the optimization of flights of multiple aerial vehicles to highlight how the research is being done in this scenario [7].

One of the most significant researches was developed by [8]. This author worked with the cooperation of two fixed-wing UAVs applied in precision agriculture. The author divides the problem into two parts: first, he uses the principle of particle flow to make fixed-wing attracted to the way-points. It then uses an optimization algorithm developed in Mixed Integer

Linear Programming (MILP) to perform the flight optimization of the two UAVs since the way-points, the landing and, the taking-off locations are known. Among the limitations of the study is the fact that it is developed for only two fixed-wing, besides being a static model, so, before the flight is executed a MILP algorithm that generates the flight plans, which will later be performed by the UAVs.

According to the author, the number of two fixed-wing is ideal for cooperative flights in precision agriculture. Fixed-wing flights have a high cost to set up the aircraft, as well as the need for taking-off and landing intervention. Increasing the number of fixed-wing also increases the number of system operators.

Still according to the author, the justification for using fixed-wing on cooperative flights applied to precision agriculture is because most of the scientific work using this type of vehicle, and, also, fixed-wing have more autonomy than Drones.

Although the author highlights in his work the limitations of the fixed-wing use, such as restrictions related to the minimum speed of the aircraft, it can not be below a specific limit to avoid the risk of falling and the non-detailed reading of data by high flight speed.

Another work aimed at the cooperation of Drone is developed by [9]. It aims to make four Drones perform the same task: the transport of a sphere on a plate and the UAVs must identify the position of the sphere on a plate and perform movements to keep it balanced. Since all airborne devices have the same algorithm, when the ball position changes on the plate, all devices perform behavioral changes in a coordinated manner. For information processing, an artificial neural network algorithm was used. Initially, the UAVs were trained for all possible situations.

Optimization through the use of an artificial neural network brings high-speed processing since the longest phase of network training. However, this training allows great results for systems with few variables. For systems with many possibilities, optimization through a neural network may not bring good results.

Another work of significant contribution was published by [10]. It presented a fixed-wing multiple flight model, tested in a simulation environment. Although simulated, this work considers issues such as angles and minimum speed for the fixed-wing to perform maneuvers.

As a limitation presented is the system is not able to find the best possible solution for the scenario, identifying only a feasible solution given the computational complexity for it.

[11] presents in his work a model for optimizing fixed-wing flights, applied to data collection from wireless sensor networks. The application scenario is interesting because, in wireless networks, one of the highest costs is data transmission between network nodes. To minimize this cost, a fixed-wing flight optimization system is proposed using the wireless sensor network topology, and the sensors are distributed in *cluster*, with the nearby sensors sending all information to a node called *head* of the *cluster*. This node centralizes all information and forwards it to the fixed-wing as soon as it flies



over that point (cluster *head* node). Battery expenditures for information exchange within the network are compared to the traditional approach, called *Low Energy Adaptive Clustering Heirarchy* (LEACH), and the fixed-wing model showed overall system battery savings.

To optimize the fixed-wing flight that collects information, [11] used the *Particle Swarm Optimization* (PSO), applying a classic algorithm used to solve the *Traveling Salesman Problem*(TPS), which is not about dynamics, not even multiple fixed-wing. However, the author points out in his conclusion that one of the future works is to present techniques to treat data collection with many fixed-wing.

[12] presents a fixed-wing group flight optimization model from a PSO and AG hybrid algorithm. This paper considers the optimization of fixed-wing paths in a three-dimensional scenario. One of the contributions of this work is that processing is performed in parallel, optimizing the use of multi-core processors. Another contribution is the use of multi-purpose function, allowing us to optimize more than one system parameter. This work was also tested in the simulation environment.

In general, it is possible to observe that no much scientific papers deal with Drones flight optimization. Also, it is possible to verify that no article studied deals with the dynamics of the environment, a widespread problem faced in real flight environments. It is believed to be because most works deal with pre-optimization rather than optimization during system execution.

### III. PROPOSED SYSTEM

Although several preliminary studies have been developed and showed the viability of a system that deals with the execution of monitoring missions formed by several Drones, a research topic is still little explored: a faster flight optimization considering the opening of the system and its dynamicity. In the studies showed in the last session, we have some works that use classic algorithms of optimization, and, almost all, needs a lot of time for training or processing these algorithms, so it can not be used in a dynamic environment like flight optimization of a group of Drones applied in the precision agriculture. As an example of these algorithms, we can highlight the PLIM, Neural Network, and PSO.

Also, we have another characteristic in theses presented works: almost all use only one vehicle or use a known number of the vehicle. In a scenario like precision agriculture, where we have a large area that must be overflown, this flight can not be done using only one, for it, we need to use a swarm with several Drones, flying together, following an optimized flight plan.

In this scenario that we are proposing to optimize, we do not have a long time to run the sophisticated algorithm that allows us to find the best results. Most of all, the time that we have among one drone landing and another drone taking-off is short, and an optimization algorithm needs to find as the best flight plan in a short time.

Also, we have other characteristics of the system that need to be considered: Opening and Dynamicity. The opening is the term used to inform that the system can receive or lose

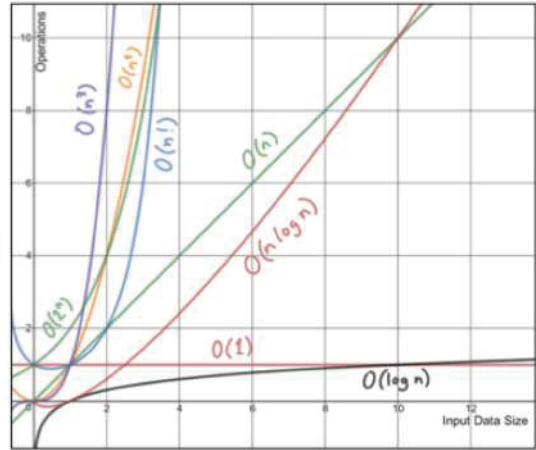


Fig. 2. Graph with the cost to solve several problems, wherein x-axis we have the number of input instances and in y-axis the complexity to solve the problem.

components in the execution time; these components could be Drones or base. Dynamicity means that the cost between points can change over time, with the change of wind speed and direction

To be possible for such optimization in a dynamic environment in which Drone group flight is applied to precision agriculture, it is necessary to choose and test a fast, dynamic optimization algorithm that can be easily distributed on different processors.

One of the most traditional heuristics that could be used in the optimization scenario is the Greedy Algorithm. According to [13], greedy is an algorithmic paradigm that builds up a solution piece by piece, always choosing the next piece that offers the most obvious and immediate benefit. So the problems where choosing locally optimal also leads to the global solution is the best fit for Greedy.

This kind of algorithm is relatively simple to apply in distributed computational systems. It allows communication within the system regardless of the number of elements that may vary over time. As it is a fast algorithm to be processed, it could be adapted to a dynamic environment.

These algorithms are known as a Linear Algorithm, which is good, because the complexity to find the result can increase proportionally with the number of the elements in the system, that is, its complexity is  $O(n)$  (the straight line in the Figure 2). Almost all optimization algorithms have an exponential complexity, where we can see in the graph as the line near the y-axis.

This way, the proposed greedy algorithm consists of making the flight plan for all Drone inside the system. The points are chosen using their distance to the recharging base. The closest point will be the first to be overflown. As a stop parameter, we have the Drone's autonomy, making it move away from the base until it is almost running out of battery, and it is time to return the base.

When the Drone returns to the base, it feeds the system with information about what points were visited, preventing

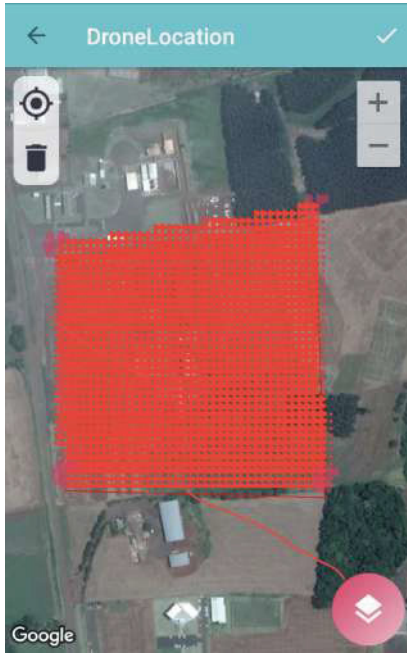


Fig. 3. Scenario 34 x 34 in a real area with a resolution of 10 meters.

avoiding that theirs can be visited twice, and also, supply the system with information regarding the environmental conditions, such as wind speed and its direction. Such information is used to calculate the cost for a point to be overflown. This cost is the amount of battery needed for this operation.

#### IV. METHODOLOGY

For the development of a model for this group flight optimization based on a greedy algorithm, the first step was to make a mapping, transforming the area that will be overflown into a Cartesian plane.

The area to be flown is a two-dimensional geometric figure, so the reading accuracy must be defined to make the markings of the points to be overflown, which is named waypoints.

For the digital elevation model [14], this being the focus of the application of this project, a precision of 10 meters was chosen, since, in Brazil, the accuracy of the digital elevation model is 90 meters. By using software interpolation, a non-real accuracy of 30 meters is possible, which is less precise than the 10 meters proposed by this work.

Thus, for the mapping of the experimental area of the university's Agronomy course, where the present study was carried out, with a size of 115,600 square meters, equivalent to 28 acres, 34 x 34 points were necessary, with a total of 1,156 points, separated for 10 meters, as shown in Figure 3.

The next step was to convert the area into a Cartesian plane, divided into x (rows) and y (columns), thus, the first point in the upper left corner corresponds to index 1-1 in the Cartesian plane ( $x = 1$  and  $y = 1$ ). The last point, in the lower right corner, has an index of 34-34 ( $x = 34$  and  $y = 34$ ).

The Drone's initial position is also known. This point is called the Recharging point, so for each point in the Cartesian

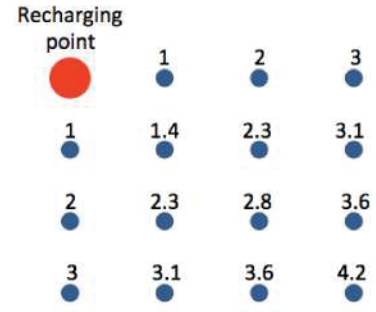


Fig. 4. Distances between the recharging point and all of the other points.

plan, was calculated the distance from the Recharging point, using for it the Euclidean Distance (DE) equation present in the Equation (1).

$$DE = \sqrt{(xa - xb)^2 + (ya - yb)^2} \quad (1)$$

This way, the Drone knows the distance value of each point to the Recharging point, and the flight plan is created classifying these points in increasing order of Euclidean distance. In the Figure4, we can see an example of a scenario with 16 points. Above each point, we have the cost to return to the researching point.

The Drone's autonomy is known, and it is known how many points it can fly over before returning to the base. Thus a flight plan is made based on this information.

As each point is added to the Drone's flight plan, it is marked as a possible visited point in a knowledge base shared by all Drones, after, when the Drone returns to the recharging base, each visited point are marked as visited, and all the other points are able to visit.

Upon returning to the base, the Drone feeds the system with the points he was able to visit, as the wind speed and wind direction. Using this information, the system will create a new flight plan for all Drones that are in the recharging base, allowing for a new optimization cycle.

As the communication only happens between the bases, it allows the Drone's battery is used only for its flight. Thus, the Drone's flight plan can only be changed while it is on the ground, recharging its battery. After it leaves for overflight, your flight plan can no longer be changed.

When the Drone is at the base, its flight plan can constantly change, since with each landing event, new information feeds the system, and a new flight plan is calculated for all Drones.

In this way, the proposed optimization system based on events, creating new optimized flight plans for Drones on the ground, which are recharging their batteries with each new landing. As the time between the landing of one Drone and the taking-off of another can be relatively short, so, the system must provide a flight plan optimization algorithm that finds feasible results in short processing time.

The greedy optimization, although not guaranteeing to find the best result for the scenario, guarantees that good results

are found in a linear processing time, which does not happen with other optimization algorithms, such as PSO or Genetic Algorithms.

## V. RESULTS

Initially, tests were performed for two scenarios: 11 x 11 and 34 x 34. For each of these scenarios, the route obtained using the Greedy algorithm was compared in relation to the optimal route.

The scenario 11 x 11 was chose because this area size can be overflowed through a DJI Phantom in a unique flight.

Considering a 10-meter precision, on the scenario with 11 x 11 points is equals than a 110 x 110 meter area, that is, 12,100 square meter, or tow acres.

One DJI Phantom has 20 m/s as maximum speed, although, for these tests, we considered 1 m/s speed, preventing with this, distortions in the data read by the sensors/cameras of the Drone.

The autonomy of DJI Phantom, according to its manual, is 25 minutes. However, a margin of 20 % error was considered. It is to prevent the Drone run out of battery before returning to the base, so its autonomy was considered in 20 minutes, that is, 1200 seconds.

If this drone were to fly over an area at a speed of 1 m/s, it would be able to cover 1,200 linear meters or 120 points. It is practically equivalent to an area of 11 x 11 points.

For the second test, the experimental area of the university where the study was carried out was used, which has a size of 340 x 340 meters, with an area of 115,600 square meters, which is equivalent to 28 acres.

How the optimization algorithm used is linear, the processing time increases proportionally with the number of points that have to be visited, which is good compared to other optimization algorithms, which have better results, but the processing time increase exponentially when have more point to be visited, which is shown in Figure ???. In this way, for 121 points (11 x 11 scenario) or 1,156 points (34 x 34 scenario), there was no considerable increase in processing time.

As a cost, for the performance analysis, it was considered the jumps from one point to another, that is, how many points were overflowed over so that the Drone could cover the entire area and return to the base. Figure 6. presents the path generated by the greedy optimization algorithm (left) and one of the optimal results for the scenario.

It is observed that the greedy algorithm, to fly over 121 points, its spend 145 points, since to return to the base, it had to fly over some points twice.

For an optimal flight plan to this scenario, it would be possible to fly over all the waypoints at the cost of 122 points. This scenario, it is 121 points, how we have an odd amount of points, one point needs to be overflowed twice, even in the best case.

A second simulation was carried out for the scenario with 1,156 points (34 x 34), a real area size. For this, a DJI Phantom 4 would not have enough autonomy to fly over this area,

TABLE I. DIFFERENCE BETWEEN GREEDY RESULT AND OPTIMAL GLOBAL.

Scenario	Points	Optimal Greedy	Optimal Global	Dif (%)
11x11	121	141	122	15,5%
34x34	1,156	1,238	1,156	7,1%

however, even so, tests were performed to compare the result using the Greedy Optimization of the best result. The result is shown in Figure 7.

To fly over these 1,156 points, with a greedy optimization, it was necessary to fly over through 1,238 points, that is, 82 points more than the optimal scenario, where no one point is overflowed twice.

Finally, a real test, using the area of 34 x 34 points, this overflowed by 4 DJI Phantom, which have, on average, an autonomy of 120 points, was carried out. The result is presented in Figure 8.

As can be seen, in this situation, 12 Drones overflights were required. The flyover time was not linear, since, at each moment, four drones were flying overhead, so the survey would take around 60 minutes of flights (20 minutes for each flight, multiplied by three which are the number of cycles). - Table I.

In the Gantt Chart of Figure 5, it is possible to see the flight of the 4 Drones over time, considering a setup time of 5 minutes for each Drone, as well as the flight time of 20 minutes and the recharging time of 30 minutes.

For all of these tests, the Greedy Optimization algorithm was running in a traditional computer. The processing time was less than 1 second, which shows that the Greedy Optimization algorithm can be applied to dynamic scenarios, where the response time is fast, since even in large scenarios, with more than 1,000 points, the processing time was very short.

## VI. CONCLUSION

The present study proposed and evaluated an algorithm for the optimization group of Drones flights applied to precision agriculture.

Since there is a need for a dynamic optimization algorithm that allows for such optimization even with varying environmental data and returns results relatively quickly without training.

Given the characteristics of the scenario in which the group flight of Drones is applied, an optimization algorithm that finds feasible results is necessary, however, in short processing time, this way, a Greedy optimization algorithm is proposed, it is applied in an event-based model, where, at each Drone landing the system is fed with information, which allows the creation of an updated flight plan for all Drones that are on the ground.

The results obtained were good. The processing time would be very short, since a linear optimization algorithm was used. And in relation to the optimal scenario, the results obtained were very close, and for the 11x11 scenario the result obtained by the Greedy Algorithm had an increase of 15% in relation to the number of points to be passed, and for the 34 to 34 scenario, the increase was 7%.

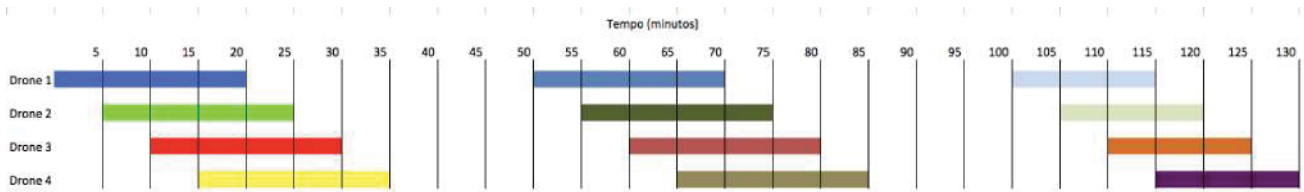


Fig. 5. Gantt chart for the Scenario 34 x 34 overflow for 4 Drones.

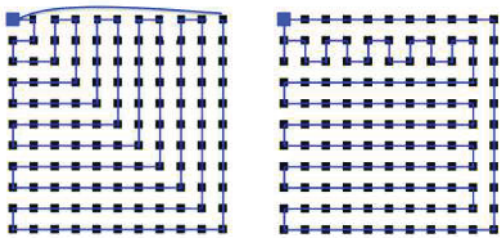


Fig. 6. Left. Flight plan generated for the Greedy Algorithm. Right. Best flight plan for this scenario. Scenario with 11 x 11 points.

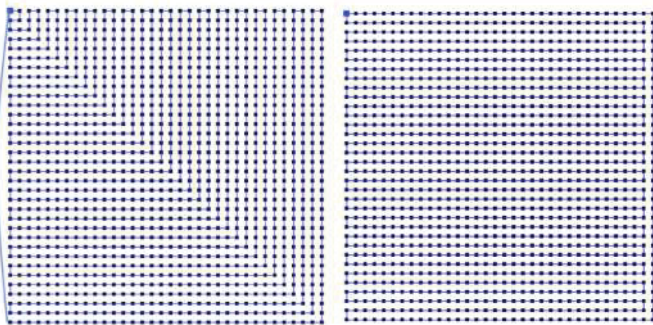


Fig. 7. Left. Flight plan generated for the Greedy Algorithm. Right. Flight plan optimal for this scenario. Scenario with 34 x 34 points.

However, the main purpose is to test the algorithm in the flight of multiple Drones, and quickly, the algorithm was able to make new flight plans, even in relatively large scenarios, such as 34 x 34, where 12 overflights were needed to cover this area. In every landing, the algorithm could find a new flight plan in less than 1 second.

For these flights, it took 130 minutes to cover all the area, which has 28 acres. At this time, it was considering the time on the ground for recharging the battery and time for Drone setup.

#### REFERENCES

- [1] Miguel Castro Neto, Pedro A. Pinto, and José Paulo P. Coelho. *Tecnologias de informação e comunicação e a agricultura*. Porto: Sociedade Portuguesa de Inovação, 2005.
- [2] Eobin Alex George, Gaurav Tiwari, R N Yadav, Edward Peters, and Srishti Sadana. UAV systems for parameter identification in agriculture. In *2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS)*. Institute of Electrical & Electronics Engineers (IEEE), aug 2013.
- [3] Alexandre S. Brandão, Felipe N. Martins, and Higor B. Sonoguetti. A vision-based line following strategy for an autonomous uav. In *2015*

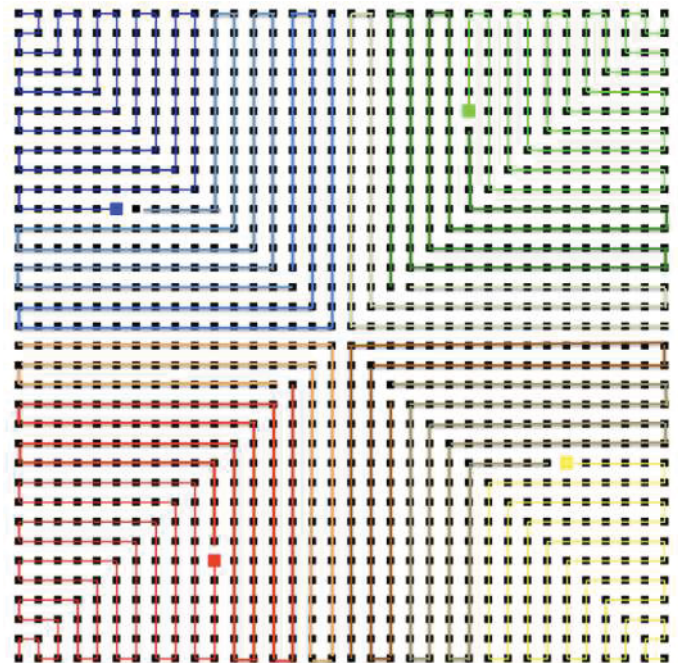


Fig. 8. Scenario 34 x 34 overflow for 4 Drones.

*IEEE 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. Institute of Electrical & Electronics Engineers (IEEE), jul 2015.

- [4] David Anthony, Sebastian Elbaum, Aaron Lorenz, and Carrick Dewtweiler. On crop height estimation with UAVs. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Institute of Electrical & Electronics Engineers (IEEE), sep 2014.
- [5] Mainara Cristina Lorencena, Robison Cris Brito, Jose Felipe Loureiro, Fabio Favarim, and Eduardo Todt. A comparative approach on the use of unmanned aerial vehicles kind of fixed-wing and rotative wing applied to the precision agriculture scenario. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, July 2019.
- [6] Mehrdad Khaledi, Arnau Rovira-Sugranes, Fatemeh Afghah, and Abolfazl Razi. On greedy routing in dynamic UAV networks. In *2018 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*. IEEE, June 2018.
- [7] Robison Cris Brito, Jose Felipe Loureiro, Eduardo Todt, and Roberto Pereira. A systematic mapping for the scenario of non-urban autonomous vehicle cooperation systems. In *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*. IEEE, nov 2017.
- [8] Gustavo Avellar, Guilherme Pereira, Luciano Pimenta, and Paulo Iscold. Multi-UAV routing for area coverage and remote sensing with minimum time. *Sensors*, 15(11):27783–27803, nov 2015.
- [9] Gerard Ely U. Faelden, Jose Martin Z. Maningo, Reiichiro Christian S. Nakano, Argel A. Bandalá, and Elmer P. Dadios. A neural network approach to a cooperative balancing problem in quadrotor-unmanned

- aerial vehicles (QUAVs). In *2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. Institute of Electrical & Electronics Engineers (IEEE), dec 2015.
- [10] Halit Ergezer and Kemal Leblebicioğlu. 3d path planning for multiple UAVs for maximum information collection. *Journal of Intelligent & Robotic Systems*, 73(1-4):737–762, oct 2013.
- [11] Dac-Tu Ho, Esten Ingar Grotli, P. B. Sujit, Tor Arne Johansen, and Joao Borges Sousa. Cluster-based communication topology selection and UAV path planning in wireless sensor networks. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, may 2013.
- [12] Vincent Roberge, Mohammed Tarbouchi, and François Allaire. Parallel Hybrid Metaheuristic on shared Memory System for real-time UAV Path Planning. *International Journal of Computational Intelligence and Applications*, 13(02):1450008, jun 2014.
- [13] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann. VNR algorithm: A greedy approach for virtual networks reconfigurations. In *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*. IEEE, December 2011.
- [14] Embrapa. Programa nacional de solos do brasil (pronasolos). *Embrapa Solos*, nov 2016.

# MultiDrone Simulator - An Open Source Multi-Plataform Tool to Use in Tests of Optimized Flight of Group of Drones

Robison Cris Brito Federal University of Technology UTFPR robison@utfpr.edu.br	André Guedes Federal University of Parana UFPR alpguedes@gmail.com	José Ricardo da Rocha Campos Federal University of Technology UTFPR jricardo28@gmail.com	Alfredo Weitzenfeld Univeristy of South Florida USF aweitzenfeld@usf.edu	Eduardo Todt Federal University of Paraná UFPR todt@inf.ufpr.br
--	--	--	--	---

**Abstract**—This article presents an open source multiplatform tool for simulating flight of Drones group. From this tool, the user can perform an offline execution, loading the flight plan from a JSON file, or even allows a dynamic online execution, since the simulator accepts the connection via socket from other client, that can be developed in any language, and it only need to allow a connection via socket with the simulator to be able to sending / receiving JSON messages. This simulator displays the result of the flight execution, as well as presenting the task allocation in a Gantt chart. During the simulation, this tool allows execution step by step, thus changes in the environment data, such as wind speed / direction, inclusion and exclusion of Drones, request for immediate Drone return of the base , among other resources. The tool was tested for different scenarios, using different optimization algorithms, and for all tests, the simulator behaved well, presenting the expected results.

**Keywords**—*Simulator, Optimization Flight, Drones Group, Dynamic Environment*

## I. INTRODUCTION

Unmanned Aerial Vehicles, also known as UAVs, were developed to facilitate monitoring, as well as information records, more quickly than terrestrial vehicles. With UAVs, data can be viewed from another perspective that until now, it was only possible by airplanes or helicopters, which significantly increased the cost of acquiring this data.

UAV flights can happen in two ways: remotely controlled or autonomous. In the second, there are the biggest advantages and, also, the biggest challenges. In autonomous flight, there is no limitation on the distance between the UAV and its controller. In controlled flight, this control often occurs using radio frequency, which is limited to the power of the transmitter, not exceeding a few hundred meters. Thus, in autonomous flights, it is possible to cover a larger area, or even a higher flight altitude, which is better to capture information through sensors or cameras inside the UAV.

In the context of autonomous flight, several works were published, such as [1], which proposes a solution for the identification of vehicle traffic on the road. In this, the UAV is equipped with a camera and, during the flight, it captures and processes the images for the identification of vehicles on the roads. The images are processed using an image processing algorithm that identifies three types of vehicles: cars, trucks,

and bicycles, computing how many of each is in the captured image. This information is subsequently transmitted to a device on land to feed a computer system.

Another autonomous UAVs example is presented in [2], which proposes the use of low-cost UAVs to assess and monitor mosquitoes that transmit diseases, such as malaria, dengue, chikungunya, and zika. The main idea is to use these autonomous vehicles to identify and spray in locations that have the focus of mosquitoes.

However, precision agriculture is where the use of autonomous UAVs stands out. According to [3], the concept of precision agriculture is usually associated with the use of high-tech equipment to assess, or monitor, conditions on a given space of land, then applying the various factors of production (seeds, fertilizers, water, etc.) accordingly with its necessity. Thus, based on specific data from geographically referenced areas, the agricultural automation process is implemented, dosing production inputs proportionally for each area, unlike traditional agriculture, in which the same amount of production inputs is used throughout all land.

Usually, in precision agriculture, UAVs called Fixed Wings, similar to small planes, are used, which the great advantage is their autonomy, that can exceed 60 minutes of flight, and since it can flight a high travel speed. It can easily fly a large amount of land in a short time.

However, a study developed by [4] presents all the advantages of using Rotating Wing UAV in relation to Fixed Wings UAV. The Rotary Wing UAV looks like a small helicopter, and it has pairs of propellers. It has limited autonomy, varying from 20 to 40 minutes. However, it has numerous other advantages in relation to the Fixed Wing, such as it can stop in the air to take a reading, it can fly at low speed. You can approach the ground for a detailed reading of the data, or going up in high speed for wider reading, but its main advantage is the cost. For example, with the price of a Fixed Wing Wingo, from UAVision Aeronautic (costs about \$ 120,000 each), it is possible to buy a large swarm of Drones Phantom, from DJI company (costs around \$ 400.00 each).

Given the limited autonomy of Rotative Wing UAV, from this moment just called as Drone, its application in precision agriculture only becomes viable if used in groups of UAVs

flying over an area autonomously, thus, a swarm with 10 Drones, for example, autonomously, could fly over a large area without human intervention. A model proposal for the use of Drone groups applied to precision agriculture is presented by [5].

In this scenario, a research theme is still little explored: the optimization of the flight plan of this group of Drones, because as such optimization happens in a dynamic scenario, where can change in weather conditions, the algorithm has to provide good results to optimize the time of flight, and at the same time, be fast enough to adapt to the dynamics of the environment.

In this area, a big challenge is to test these kinds of optimization algorithms, since most of the existing simulators provide ways to test only the flight of a single Drone, and not of a group, where Drones are dynamically inserted/deleted in the scenario.

In this context, the present work aims to propose and evaluate a simulator to test flight optimization algorithms for a group of Drones, these applied to precision agriculture, being necessary to add to the simulator the dynamics of the environment, to identify how the algorithms behave with the changing the speed/direction of the wind, as well as the early termination of the battery of any Drone, or allocation/exclusion of Drones from a mission while it is happening.

## II. RELATED WORKS

Regarding the tools and software commonly used for testing flight simulation of Drones, the tools that follow are the best known and have been tested by the author before the present work. In the following, there is a brief explanation of each tool, as well as its main characteristics:

- **Paparazzi UAV:** Available at <http://wiki.paparazziuav.org>, this is an open Software and Hardware project that covers autopilot systems and ground station software for UAVs of various types. This platform focuses on autonomous flights, although it can also be used on manual flights. One of its main features is the control of multiple UAVs. This system allows the definition of flight plans that happen as if they were missions, where way-points are treated dynamically;
- **JAUS - Joint Architecture for Unmanned Systems:** This architecture was designed by the United States Department of Defense in 1998. Available at <http://openjaus.com>, it has become an international standard for defining communication protocols for Unmanned Vehicle Systems. It employs an SOA-based approach (Service Oriented Architecture) to allow distributed command and control of these systems. This architecture is developed by the Society of Automotive Engineers under the Steering Committee of Unmanned Systems of Aerospace Standards. All documents that define JAUS can be purchased online directly from SAE. It is a heterogeneous architecture, which may involve land, water, and aerial vehicles; these can be autonomous or remotely controlled. JAUS is composed of the JAUS Service Interface Definition

Language (JSIDL), which is the communication protocol and the JAUS Core Service Set (JSS Core), which is the set of tools;

- **AETOURNOS : Airborne Embedded auTonomOUs Robust Network of Objects and Sensors:** This project is available at <http://aetournos.gforge.inria.fr>. It aims to control multiple UAVs, as well as work with their flight training. The platform offers a complete environment for testing and simulation, allowing simulations in environments with land, air, and water vehicles. This project makes use of Parrot Drones, AR model. Drone 2.0;
- **TAEMS - Task Analysis, Environment Modeling and Simulation:** This framework allows to model environments to be used in complex computational tasks, in which agent-based approaches can be used. With it, it is possible to simulate a group of UAVs working cooperatively. It allows analyzing and simulating the behavior of a Multi-Agent System quantitatively about the relevant characteristics of the computational environment in which it is inserted. TAEMS is thus not only a language that allows specifying the characteristics of a multi-agent system, but also a simulation system that allows demonstrating the tasks graphically, actions of agents and statistical data;
- **Mission Planner - ArduPilot Fligh Controller:** This environment is presented at <http://ardupilot.org/planner>, and has the function of being a control and simulation model for UAVs. It is integrated with Google Maps and allows you to create missions, assigning way-points to a UAV. So, this environment is part of the ArduPilot project, and must be used with the UAV microcontrollers provided by it;
- **Simulink para Matlab:** Another way to perform UAV flight simulations on a computer is to use Simulink, a simulator available by Matlab that allows you to model software agents featuring as UAVs, which can communicate and perform tasks in a simulated environment. Therefore, the application must be coded using the Matlab programming language;
- **UAVSim:** It is a UAV open-source simulator, available at <https://www.openhub.net/p/uavsim>, most used for simulations where the UAV must search for a target and/or attack it. In this environment, UAVs are modeled as software agents and communicate to achieve a common goal;
- **The Network Simulator - ns-2:** Open Source project available at <http://www.isi.edu/nsnam/ns>, it is a simple network simulator that allows simulating the communication between network elements. It is commonly used by projects that involve communication between UAVs for simulation and test environments.

With the study of the tools above, it was possible to identify the advantages and disadvantages of each environment, to propose a specific environment for the testing flight of Drone groups with the main objective of testing the optimization.

From the tools mentioned, some were proprietary without a free version for evaluation or providing a very limited version for testing. Others needed specific knowledge of some programming language to perform the tests and flight simulation, others were specific to a specific Drone model, or even, they had a focus more applied to the actual flight, not necessarily the simulation. Finally, many of them do not allow, at least without the use of a programming language, to treat the dynamics of the environment.

Thus, the present work proposes the MultiDrone Simulator, a software environment, developed in Java, that is, multiplatform, which allows the simulation of the flight of multiple Drones focusing on the dynamics of the environment, as well as on the ease of its use.

### III. PROPOSED SYSTEM

To simulate the environment, easier and quicker, without the need to program in a specific simulator, the Multidrone Simulator was developed. This is a free and open-source simulator. Developed on the Java platform, because of that, it can be run on multiple platforms, such as Windows, Linux, and Mac OS.

In the Multidrone Simulator, the user can upload a file containing the flight plan to be tested. This flight plan is a JSON file (JavaScript Object Notation), so this flight plan can be created in any language, making this simulator independent of the programming language.

Another feature supported by Multidrone Simulator is the possibility of dynamic testing. In static tests, the user loads the flight plan once, in JSON format, only at the beginning of the simulation, after the flight characteristics are placed, such as wind information, number of Drones, where they are found in the plane, and runs the simulation, from this moment on, it is not possible to interfere in the tests.

But the Multidrone Simulator has another mode of operation. It is named online mode. In this, the simulator opens a Socket server, allowing a client to connect and exchange information using its network. Thus, the client connects to the simulator's Socket server, sends the JSON of the initial flight plan, changes the characteristics of the environment directly in the simulator, and runs the tests. As the user changes some feature in the simulation environment, such as adding or removing Drone, or changing the information regarding wind speed or direction, the simulator, via Socket, sends such changes to the connected client, who has the opportunity to create new flight plans based on these changes and send it back to the simulator, which executes it.

Following are other features offered by Multidrone Simulator:

- **Definition of the size of the overflow area :** The simulator allows, graphically, to define the dimensions of the area to be overflowed. This size is defined in a number of way-points, which are the points over which the Drone must fly;
- **Definition of Drones:** The environment allows the simulation with one or more Drones, the user, when defining a Drone, must specify its name, which will

be used for its visual presentation in the simulator. Also, it is possible to define its autonomy, which is defined in way-points, its initial position in the area to be flown, and still, a standard range for all Drones, if simulating with only one Drone model;

- **Import of the initial flight plan:** The environment allows to import an initial flight plan generated by other programs, this file must be in JSON format and will be executed by the simulator.

The following are some data visualization features offered by Multidrone Simulator.

- **Visual map of the routes:** The simulator graphically displays the flights performed by each Drone. Each flight is represented by a line, which is drawn on the way-points overflowed by the flight. To differ the flight between different Drones, the lines have different colors;
- **Interactive Execution:** The user can opt for an interactive flight execution over time, in a step-by-step format, to check the behavior of each Drone, when they return to the charging base, how long they spend recharging the battery, among other information. The user, during this execution, can change characteristics in the environment, such as wind speed and direction, to see how his algorithm behaves, however, for that, he must use online execution, that is, his application must be connected to the simulator Socket server;
- **Complete Execution:** The user can click on the Play button on the simulator, to perform an execution without having to advance each step, however, at any time, the user can pause the complete execution and proceed with the execution step by step;
- **Gantt Chart:** Since the problem simulated by this environment is a resource allocation problem, where several Drones negotiate to fly over optimally several way-points, the environment also provides a Gantt Chart, in which the user can see the result of the execution from the allocation point of view, knowing at what time each Drone was overflight or was at the base reloading, as well as how many points it overflowed in each flight plan.

Finally, the environment also allows you to simulate the dynamics of an environment, which is the most important item in the Multidrone Simulator. In this simulator were implemented:

- **Adding/Removing Drones dynamically:** Allows you to simulate situations such as failure in Drones, or allocation of new Drones to the environment at the time of the mission's execution;
- **Early Drone Return to Base:** Even though the Drone's autonomy is known, at any time the user can request the Drone to return to base, simulating, for example, early termination of its battery and seeing how the system reacts to this adverse event;
- **Change in Weather data:** The cost of moving the Drone between one point and another change, basically, with the speed and direction of the wind. Thus,



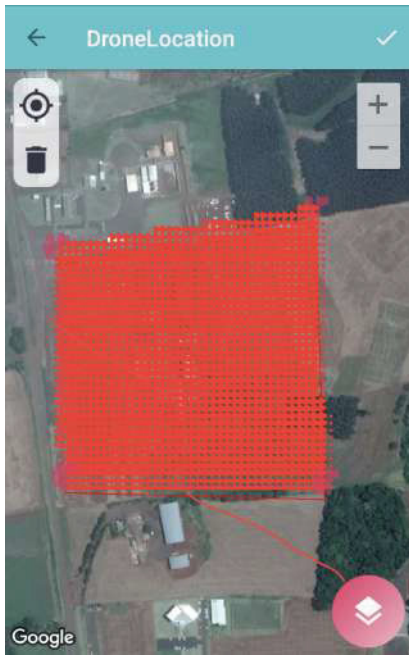


Fig. 1. Scenario 34 x 34 in a real area with resolution of 10 meters.

in the environment, the user can change this cost at any time in the four axes (North, South, East, and West), checking how the optimization algorithm behaves in these environmental changes.

#### IV. RESULTS

The results of the execution of two algorithms: Greedy flight allocation algorithm [6] and Algorithm based on recursive auctions [5], are tested in Multidrone Simulator.

Tests were carried out in various sizes of areas, aiming to know how the simulator behaved for different scenarios. The biggest scenario where the simulator was tested was the experimental area of the university where the study was carried out, which has a size of 340 x 340 meters, with an area of 115,600 square meters, which is equivalent to 28 acres.

For all the examples tested here, each way-point was distance 10 meters among them, so, to the real scenario was necessary 34 x 34 way-point, in total, 1,156 points - Figure 1.

For this scenario, the simulator could present the result in a graphic interface, but, due to the number of the point, the big images can not be showed with details in this paper, so, we reduce the area to 11 x 11 points, this way we can show all the information through the images.

A scenario of 11 x 11 crossing points was used, which gives a total of 121 crossing points. The computational complexity to find an optimal solution for this scenario is bigger, since this problem is similar to the Traveling Salesman Problem (TSP), with some more restrictions such as resource allocation. Still, it is as we had to go through 121 cities in the TSP.

As we can see, in Figure 2, we have the result of executing the Greedy Optimization Algorithm (left) and the one result based on Recursive Auctions (right), for the 11 x 11 point

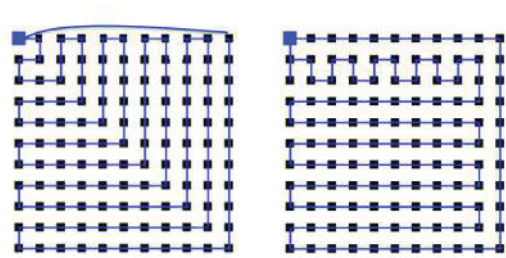


Fig. 2. Left. Flight plan generated for the Greedy Algorithm. Right. Optimal Flight plan for this scenario found using Recursive Auction Algorithm.

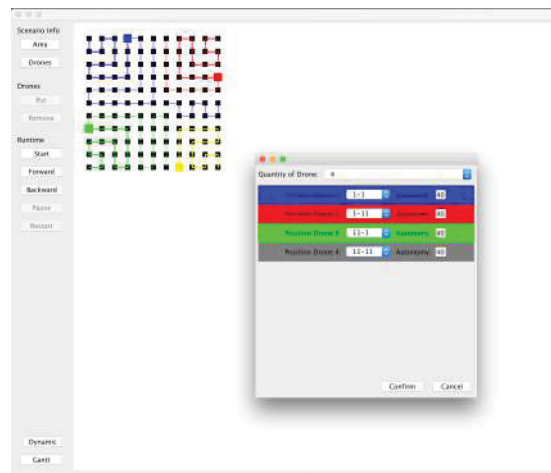


Fig. 3. Flight plan generated for the Greedy Algorithm using 4 Drones, with 40 points of autonomy each.

scenario. The right result is the optimum global result. For these executions, it was considered only one Drone, with enough autonomy to overflow all the points.

A second execution to test the simulator, where this same area (11 x 11) was overflight using 4 Drones, with 40 points of autonomy each. The result of the flight allocation is shown in Figure 3.

Another way to see the result of the flight and the allocation of resources is from a Gantt Chart, where on the y-axis we have the 4 Drones used in the simulation, and on the x-axis the time measured in time of flight between one point and another - Figure 4.

In this scenario, only Drone 1 had to make two flights. In the Graph, the waiting time at the base for recharging the battery is dark.

The dynamic screen of the environment is also presented, where resources can be allocated and removed at any time, as well as the direction and intensity of the wind, as shown in Figure 5.

Other tests were done, for example, running the simulation using step-by-step, modifying values of the environment, deleting Drones while one execution is running. In all of these cases, the simulator works well and shows the results to the user.

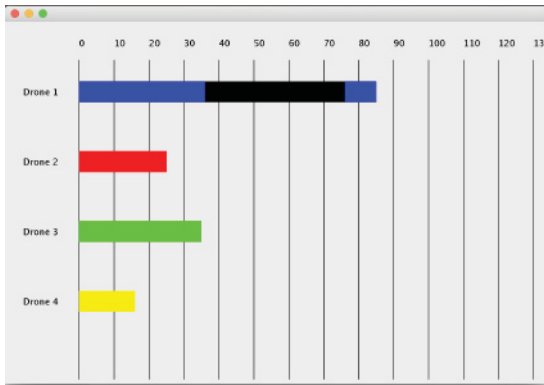


Fig. 4. Gantt Graph related to the allocation of four Drones in a Scenario with 121 points.

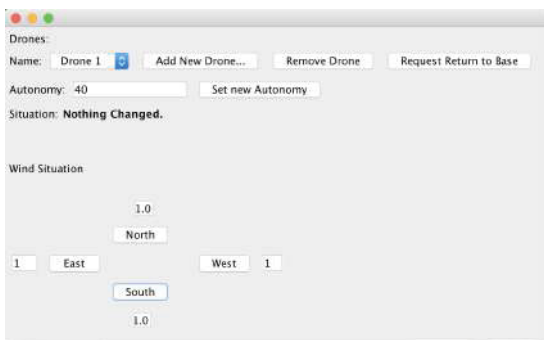


Fig. 5. Screen with infos that can be changed to the environment.

## V. CONCLUSION

The present study proposed and evaluated a simulator for optimization flights of a group of Drones, this applied to precision agriculture.

Before the development of this simulator, called Multidrone Simulator, eight simulators were studied: Paparazzi UAV, JAUS, AETORNOS, TAEMS, Mission Planner, Matlab Simulink, UAVSim and The Network Simulator. All have strengths, but also some weaknesses, which motivated the development of the simulator presented in this article.

Multidrone Simulator has characteristics such as the offline execution of one flight plan, as well as its online execution, in which it can exchange information with another application, through Socket, presenting the events and processing new flight plans.

The present simulator also has no limitations on the number of Drones and allows testing in dynamic environments, including simulating the inclusion/exclusion of Drones at runtime, changes in wind speed/direction, as well as the early termination of a Drone battery.

Step-by-step execution mechanisms for missions were also developed and tested, as well as visual presentation of data, using a Gantt Chart to verify the allocation of resources.

The Simulator worked well for different types of Drone group flight optimization algorithms, as well as in large scenarios, with more than a thousand crossing points, including overflight with several Drones.

As future work, now that the simulator has been tested and verified its functionality, it is to carry out its communication with the real Drones, making the simulator share the flight plan, which will be executed by the Drone itself, a resource that already exists in some tools studied before the development of this simulator.

## REFERENCES

- [1] Raphael Montanari, Daniel C. Tozadore, Eduardo S. Fraccaroli, and Roseli A.F. Romero. Ground vehicle detection and classification by an unmanned aerial vehicle. In *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*. Institute of Electrical & Electronics Engineers (IEEE), oct 2015.
- [2] John-Thones Amenyo, Daniel Phelps, Olajide Oladipo, Folly Sewovoe-Ekuoe, Sangeeta Jadoonanan, Sandeep Jadoonanan, Tahseen Tabassum, Salim Gnbode, Tanging D. Sherpa, Michael Falzone, Abrar Hossain, and Aerren Kublal. MedizDroids project: Ultra-low cost, low-altitude, affordable and sustainable UAV multicopter drones for mosquito vector control in malaria disease management. In *IEEE Global Humanitarian Technology Conference (GHTC 2014)*. Institute of Electrical & Electronics Engineers (IEEE), oct 2014.
- [3] Miguel Castro Neto, Pedro A. Pinto, and José Paulo P. Coelho. *Tecnologias de informação e comunicação e a agricultura*. Porto: Sociedade Portuguesa de Inovação, 2005.
- [4] Mainara Cristina Lorencena, Robison Cris Brito, Jose Felipe Loureiro, Fabio Favarim, and Eduardo Todt. A comparative approach on the use of unmanned aerial vehicles kind of fixed-wing and rotative wing applied to the precision agriculture scenario. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, July 2019.
- [5] Robison Cris Brito, Jose Felipe Loureiro, Andre Guedes, and Eduardo Todt. Optimization system for dynamic flight planning for groups of multiagent system and recursive auctions. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, July 2019.
- [6] Pham Dinh Thanh, Huynh Thi Thanh Binh, Do Dinh Dac, Nguyen Binh Long, and Le Minh Hai Phong. A heuristic based on randomized greedy algorithms for the clustered shortest-path tree problem. In *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, June 2019.

# Applying the Methodology of the Nearest Neighbor and Area Division for optimized group of Drones Flight through Recursive Auctions

Robison Cris Brito\*  
Federal University of  
Technology - Parana  
Pato Branco, Parana  
robison@utfpr.edu.br

Eduardo Todt†  
Federal University of  
Parana  
Curitiba, Parana  
todt@inf.ufpr.br

## ABSTRACT

This paper presents two methodologies that allow to reduce the processing time of the optimization algorithm based on recursive auctions, without significantly compromising the quality of the result. It is the closest neighbor methodology, which has reduced processing time by more than 60%, and the area division methodology, which has reduced processing time by more than 65%.

## CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

## KEYWORDS

Drone, Optimization, Recursive Auction, Precision Agriculture

### ACM Reference Format:

Robison Cris Brito and Eduardo Todt. 2020. Applying the Methodology of the Nearest Neighbor and Area Division for optimized group of Drones Flight through Recursive Auctions. In *Proceedings of*. ACM, New York, NY, USA, Article 4, 4 pages.

## 1 INTRODUCTION

Although several preliminary studies have been developed and showed the viability of a system that deals with the execution of monitoring missions formed by several Drones, a research topic is still little explored: a faster flight optimization considering the opening of the system and dynamicity of the environment.

Opening is the term used to inform that the system can receive or lose components in the execution time, these components could be UAV or Bases. Dynamicity means that the cost between points can change over time, with the change of wind speed and direction

To be possible for such optimization in a dynamic environment in which UAV group flight is applied to precision agriculture, it is necessary to choose and test a fast, dynamic optimization algorithm that can be easily distributed on different processors. The proposed

\*Professor of the Computer Department - UTFPR-Câmpus Pato Branco

†Professor of the Computer Pos-Graduated Program - UFPR-Curitiba

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© Copyright held by the owner/author(s).

algorithm for this work is the recursive auction based optimization algorithm.

This algorithm is relatively simple to apply in distributed computational systems from the FIPA Contract-Net protocol and allows the communicate within the system regardless of the amount of elements that may vary over time. According to [3], auction-based trading methods are often used to perform the distribution of tasks assigned to teams with dynamic elements.

Analyzing the problem in which the proposed system applies, it is possible to observe an analogy to a traditional auction, in which the UAV would be the consumers, and have as their trading currency their autonomy. They share this information with the bases of refills, which have the role of auctioneer that offer these resources to those interested by the auction, which in this analogy are the points that wish to be visited. Each point receives the proposal for the Drone's visit, and the applicant responds to the cost to be visited, as well as how many unique points will be visited on the same flight. The best proposal is chosen based on the desired objective function. In the proposed system, the objective function is to maximize the number of visit points in a single flight.

If we run the auctions exhaustively, testing all possible possibilities, we can find the best result for the scenario, but in this situation we have the same problem as the classic algorithms proposed to find solutions to the TSP (Traveling Salesman Problem), for example: time Processing is disproportional, so in a scenario with a few dozen points, a normal computer cannot find results in a feasible time.

The problem presented where a UAV should visit a known number of points only once and return to base is similar to the resolution of the Hamiltonian circuit, where we have a graph in which the vertices are the points to be visited. Edges are the paths between one point and another, and the UAV, in turn, has to make the decision as to which points it can visit, optimizing flight time, and consequently the use of its battery. The Drone has to go through all points only once and return to the recharge base.

This work presents two methodologies that allow to reduce the number of exchanging messages, and consequently, the processing time, in a model that optimized the flight of multiple Drones through recursive auctions. this model is presented with details in [1].

## 2 RESEARCH PROBLEM

For an overview about the optimization dynamics proposed by this work, we have Figure ???. In this, a Drone sends a request for



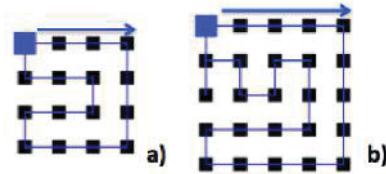


Figure 4: Comparison of routes obtained for Hamiltonian (a) and non-Hamiltonian (b) circuit.

Tabela 2: Number of auctions for Hamiltonian and non-Hamiltonian circuits

Scenario	Quantity of Points	Quantity of Auctions
4x4 points	16 points	43 Auctions
5x5 points	25 points	137 Auctions

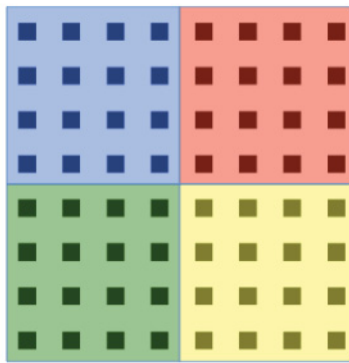


Figure 5: Graphical representation of the area division for flight in an 8x8 scenario with four Drones.

Regarding the number of auctions, for Hamiltonian circuits it can find results more quickly, given the pattern of flights coming-and-going, for non-Hamiltonian circuits, the number of messages is greater, as can be seen in Table 2.

### 3.2 Definition of Areas of Operation

Finally, for flights involving multiple Drones, it was observed that recursive auctions often extended to areas that would normally be overflowed by other Drones and, for this reason, these auctions were not chosen because they flew over a few points.

To avoid this type of auctions, a methodology was created to divide the Drone's area of operation because of its proximity to the base, thus, each Drone does the auctions with the points of its area of operation, only after this having been overflowed the Drone can fly over points in the area of operation of other Drones.

Figure 6 presents different areas of operation for an 8x8 point scenario, overflowed by four Drones. In the Figure, each color represents the area of one Drone operation.

Tabela 3: Number of auctions before and after applying the area division methodology by Drones

Scenario	Without area division	With area division	Number of Auction
8x8 points 4 drones	536 Auctions	187 Auctions	65,1%

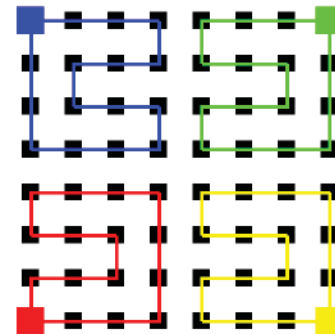


Figure 6: Flight division of four drones using the area division methodology.

To test this methodology, the flight plan was initially generated using the Identifying the Nearest Neighbor Methodology. Afterwards, a flight plan was generated using the methodology of the area division for Drones flight. The result is shown in Table 3.

As can be seen, after applying the methodology for dividing the Drones' area of operation, the first result was obtained with a 65.1 % decrease in the number of auctions. The result obtained using this methodology is presented in Figure 6.

## 4 CONCLUSION

The present work proposes two methodology to reduce the processing time in an algorithm for optimization Drones flights applied to precision agriculture. The scope of this system aims to solve existing limitations in proposals that use Fixed Wings to monitor large areas.

The first methodology, which makes the choice of the nearest neighbor, improved the processing time by 62.8 %, without significantly losing the quality of the response presented. It was also possible to observe that the improvement is more significant in scenarios that form a Hamiltonian circuit.

The second methodology, the definition of the area of operation, improved the processing time by 65.1 % in relation to the model without application of this methodology. All of these tests was performed in a simulation environment named MultiDrone Simulator.

## REFERÊNCIAS

- [1] Robison Cris Brito, Jose Felipe Loureiro, Andre Guedes, and Eduardo Todt. 2019. Optimization System for Dynamic Flight Planning for Groups of Drones using

- Cooperation with Mobile Recharge Bases by Means of Multiagent System and Recursive Auctions. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE. <https://doi.org/10.1109/compsac.2019.10262>
- [2] Robison Cris Brito, Jose Felipe Loureiro, and Eduardo Todt. 2020. Choosing the Best k-neighbor for Optimizing a Flight of a Group of Drone using Recursive Auction Algorithm applied to the Precision Agriculture. In *7th Annual BSides Tampa IT Conference*. BSides. <https://doi.org/10.1109/compsac.2019.10262>
- [3] Charles E. Pippin and Henrik Christensen. 2012. Cooperation based dynamic team formation in multi-agent auctions. In *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III*. SPIE. <https://doi.org/10.1117/12.919551>

## **A Model based on Multiagent System, Cognitive Agents, and Auctions applied to a Group of Drones Flight**

**Robison Cris Brito<sup>1</sup>, Maycon Meier dos Santos<sup>2</sup>, Eduardo Todt<sup>3</sup>**

<sup>1</sup> Federal University of Technology - Paraná, Pato Branco, Brazil

<sup>2</sup>Colorado University - Colorado Spring - USA

<sup>3</sup>Federal University of Parana - Curitiba - Brazil

(robison@utfpr.edu.br, mmeierdo@uccs.edu, todt@inf.ufpr.br)

**Abstract.** *This work presents an applied model of cooperation to optimize flights of unmanned aerial vehicles like quadcopters, also known as Drones, involved in precision agriculture. This model uses a Multiagent System to allow up the opening, which is the property of inserting and removing elements from the model at any time. To allow dynamism, which is the characteristic that the model has to recover from adverse events or failures, cognitive agents with BDI structure were used. To guarantee the exchange of messages in dynamic number of elements, the FIPA Contract-NET protocol were used. This model aims to optimize the flight time, which directly reflects the optimization of the Drone's battery use. This has to be the great limitation of this kind of aerial vehicle and which inhibits its use in precision agriculture. From the point of view of openness and dynamics, the model was tested in the MultiDrone Simulator, allowing to generate new flight plans, even with the simulated adverse events. The results of the simulation tests carried out maintain that the proposed model behaves as expected, showing itself as a promising research platform for the use of drones in precision agriculture scenarios, since this model allows the use of multiple Drones in environments dynamic and open, guaranteeing the flight optimization, which ensures battery saving for Drones.*

### **1. Introduction**

Unmanned Aerial Vehicles (UAV) are unmanned aerial vehicles controlled by distance and/or autonomous vehicles. UAVs were initially conceived for military proposes, but with the advance of new technologies and a consequent decrease in costs their use is being extended to a variety of area such as environmental management (pollution, forests, climate, or scientific applications), monitoring of wild fire, border control, combating drug trafficking, air surveillance, mapping, traffic monitoring, humanitarian aid, search and rescue of people and animals, and precision agriculture.

The importance of UAVs in precision agriculture is highlighted by authors such as [George et al. 2013] who have registered UAVs as the best devices for evaluating agricultural production, for being fast and efficient when compared to traditional ground techniques, and allowing data from most soil reliefs to be collected and analyzed. [Brandão et al. 2015] reinstate that and emphasizes that UAVs are versatile devices that can be used for a variety of agricultural applications, such as automatic monitoring of planting and of irrigation laterals, detecting and controlling of plagues, and in counting of grazing animals.

[Brandão et al. 2015] also states that autonomous UAVs have a relevant roll in the agricultural environment and are frequently used for monitoring and inspecting large cultivation areas. [Anthony et al. 2014] whose work proposes the use of UAV to monitoring plant growth, also reinstates the use of autonomous air aerial vehicles, since data can be collected and processed by specific algorithms without requiring a specialist user. Integrating automated flights and data processing in the same device

From a technical point of view, there are currently two types of UAVs: The fixed-wing UAVs, which have shape of an airplane, and the rotative-wings, that is similar an small helicopter, also known as Drones.

The fixed-wing UAVs are the most commonly used in precision agriculture, due to its autonomy. This type of UAV can flight up to 10 hours depending on the fixed-wing model. However, rotative-wings UAVs are more accurate in reading data for being able to fly with a much slower horizontal speed without the risk of falling. They also have higher maneuverability that allows it to make sharp curves and change their direction movement fast, as oppose to the low maneuverability of fixed-wing UAVs. Rotative-wings UAVs also don't require human interaction for take-off and landing, since these operations are vertically oriented. There are researches for automating operations of landing, battery recharging and take-off ([Maini and Sujit 2015]). Finally, Drones stand out for their low cost, being up to ten times cheaper than fixed-wings UAVs.

On the other hand, Drone usage for agricultural monitoring faces some limitations for large fields monitoring. This devices have a relatively low battery autonomy that precludes long flights, as well as the complete monitoring of large fields with a single Drone. To solve that problem a group of collaborative Drones can be used to optimize the monitoring of big areas, allowing efficiency and precision and creating means of contingency. This would allow other Drones to take over the tasks of a malfunctioning Drone.

Another advantage of using collaborative Drones is that the flight time of each Drone is drastically reduced and, therefore, the battery used is also reduced. This way, the same mission can be developed faster by Drones than by fixed-wing UAVs that are commonly used in precision agriculture. Also, the use of Drones that can land to recharge by itself can significantly reduce the need of human interaction in the process.

In order to develop a collaborative Drone system to be applied in precision agriculture, two main factors have to be considered:

- **Openness:** Property that allows the inclusion and exclusion of resources into the model, such as Drones and locals that are willing to be used as research sites during the execution of the project;
- **Dynamism:** Property that allows the execution of the model, independently of adverse events and failure;
- **Optimization:** The proposed model will use an algorithm of dynamic optimization that will support the generation of flight plans for multiple Drones.

Based on the properties that were previously state, this paper proposed a Multi-agent System structure aligned with cognitive agents coding, to provide the openness and dynamicity of the model and allowing the use of distributed optimizing algorithms. The model will take advantage of open source tools, such as MOISE+ [Hübner and Sichman 2003] to structure the multi-agent system, JASON



[Hübner et al. 2004] to develop the cognitive agents, FIPA Contract-NET [FIPA 2002] for communication between agents.

We are focused in presenting the structural model, which is composed by the Multiagent System, the cognitive agents and the communication technic. The optimizing algorithm tests are not going to be presented in this paper, but the literature provides algorithms such as the Greedy Algorithm proposed by [Khaledi et al. 2018] or the recursive auction algorithm proposed by [Brito et al. 2019].

## 2. Research Scenario

The architecture that the current project inserts is exhibited in the Figure 1. The picture shows a group of Drones, which were programmed to autonomously land and take off. Their recharge bases are automatic off-road vehicles, which can change its position on the terrain to be monitored while the Drones standby recharging. The system's communication, the focus of this paper, happens only between recharging bases because energy is not limited in those, as opposed to the drones. Thus, each base has a mechanism that allows them to communicate in order to ensure the well distributed optimization algorithm's execution proposed. For the mission's management by a user, there will be a web system of mission management, where the drones are assigned to the mission, as well as their recharging bases.

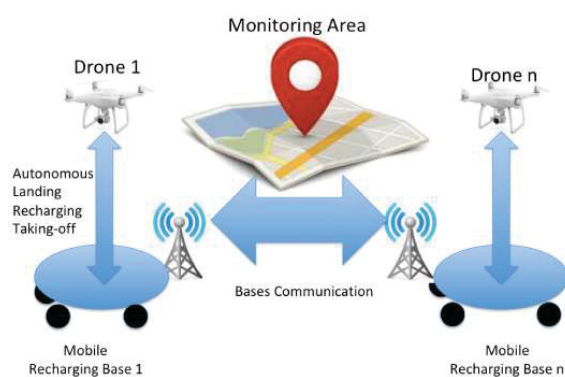


Figure 1. Model overview: Optimizing the flight of several drones

We are now going to present the challenges that will be faced when implementing this model to the proposed problem, as well as previous works developed or collaborated with by the author, aiming to the application of the model.

1. **Autonomous Landing/Recharging/Taking-off** : : This subject was already explored before by authors such as [Maini and Sujit 2015], and is feasible. Plenty of systems allow that the Drone returns to its dock using its own GPS resources, thus when it is flying over the dock it is possible for it to land using computer vision;
2. **Mobile Recharging Dock**: Although not mandatory, the recharging dock in rural environments may be mobile, such as a vehicle for example, and powered by solar panels. Thus, when the Drone lands and starts its charging process, which may take several minutes, the charging dock is free to move to a different strategical location, aiming to optimize the flight time regarding the vehicle movement. Some papers propose such vehicle, as presented in [Puhl et al. 2018]. Referring to the positioning of the autonomous vehicle in a new strategical position in the

area to be explored, there are specific techniques for the choice of the best positioning of the mobile recharging dock, one of these techniques is presented in [Saque et al. 2019];

3. **Area to be monitored:** for the execution, the area to be monitored must be known and registered in a mission management system. In this system, every point of view must be informed, as well as out-of-bounds points, starting dock position, number of drones needed for the mission, amount of available recharging docks, dock positioning, distance between the data points, among others. As the management system involves controlling multiple drones, it must be as simple as possible for the user, in a way that it provides only the necessary data for the mission to start, and after that the control must be carried on autonomously through an intelligent system;
4. **Communication between the charging docks:** the optimization of the drone's autonomy is a crucial factor for the proposed model, since this is the great limitation of its use in precision agriculture. For such optimization, an important setting is to limit the amount information exchanged between the drones during the mission execution, i.e., after they took off. This way, it is suggested that the communication occurs mainly between the docks, since as stated the docks have solar-based charging mechanisms, which has no battery limitations. Therefore, once the drones are back to the charging dock, it updates the whole system with the updated model, sharing the areas it mapped, updating other information such as the sensor data and camera, wind speed and weather conditions. The model processes such information and define a new flight plan for each drone. This communication process is named "half-line" and it is event-based.

According to these steps, the proposed model will be run in one of the recharging bases, which will be the main base, that will evaluate new flight plans every time a Drone lands. Due to the environment dynamism, the model will be based on a Multiagents System and Cognitive Agents, which allows the processing of data to take place in either a single device or to be distributed to multiple devices of the network while they communicate. An example of a Multiagent System using Moise+ and Jason distributed over a Drone network is presented by [Menegol et al. 2018].

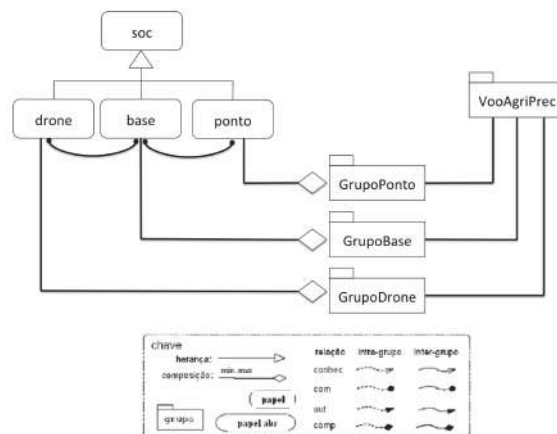
### 3. Methodology

Aiming to assure the environment openness, the proposed model uses a Multiagent System that is based on organization. Therefore, for every change in the system, such as including or excluding a Drone, new flight plans are generated for every check point.

In order to assure dynamicity during the change of nature factors such as wind speed and direction, or operational changes such as early battery discharge of a Drone, the agents are developed based on mental state of believes, desires and intentions is proposed.

The communication will be developed based on an auction protocol, assuring that it can work with any number of elements.

To optimize the system, the model has this variable as a degree of freedom that allows it to be set the way it best fits the system. In this paper, only the organization, the openness and the dynamicity will be tested.



**Figura 2. Definition of the groups, the roll of the agents and the communication restrictions**

### 3.1. Model's Organization of the Agents

This paper approach is based on a Multiagent System that's is mainly focused on organization and is provided by the free source tool MOISE+. The reason for this decision is due the openness model as well as assuring its dynamicity. The agent groups are well defined and their rolls are known by all agents. This allows good execution of the process even with the high dynamicity of the environment where the flights are taking place, and being able to recover from adverse external events such as drone's battery discharge, increase in covered area during the mission, change in wind speed and direction, including or excluding drones, as well as other unexpected events.

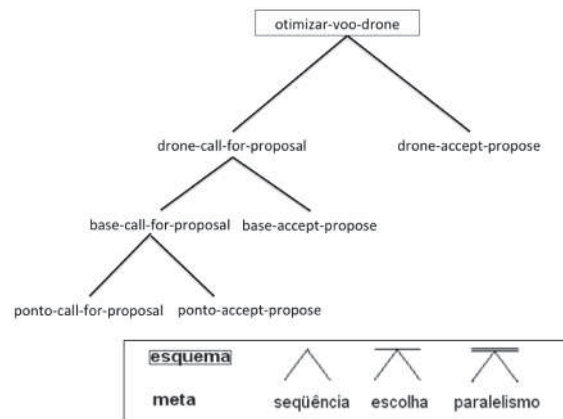
We started settling the Multiagent System by creating the Organizational Specs that was done using a XML file in the MOISE+ platform. This specs include three well defined groups of agents:

- grupo-drone: Where Drone agents are stated;
- grupo-base: Where recharging bases are stated;
- grupo-ponto: Where the points that need to be covered by the mission are stated.

The Multiagent System Structural Specs also define that the agents can take three different rolls: drone, base or check point. The system will receive an agent for every element that exist in the real testing scenario.

The amount of agents in each group is free to vary, but there is a minimum of one agent per group and no limits for a maximum amount of agent in any groups. There is also a restriction for the communication between the groups, where the Drone-group can only communicate with the Bases-Group and the Bases-Group can only communicate with the CheckPoint-Group. The agents in the CheckPoint-Group can only communicate with agents that belong to the same group. Figure 2 illustrates an overall view of the agents and their rolls in the system.

The communication inside the model follows the logic of a traditional auction where Drones work as auction clients that are willing to acquire a product a whose currency is their battery autonomy. The products of the auction are the check points that need to be covered during the mission, and our goal here is to optimize the number of



**Figura 3. Functional Specs of the Multi-agent System**

points that are taken. The Drone will be negotiating with the recharging bases, and the recharging bases negotiate with the check points in order to create the flight plan. Based on this analogy, the auction protocol from FIPA Contract-NET was chosen for the communication.

To set the negotiation process in the system, a Functional Specification was generated in MOISE+. This function sets the Drones to be the agents who start the negotiation. This process is run over all Drones to start the simulation and also every time the Drone returns to the recharging base.

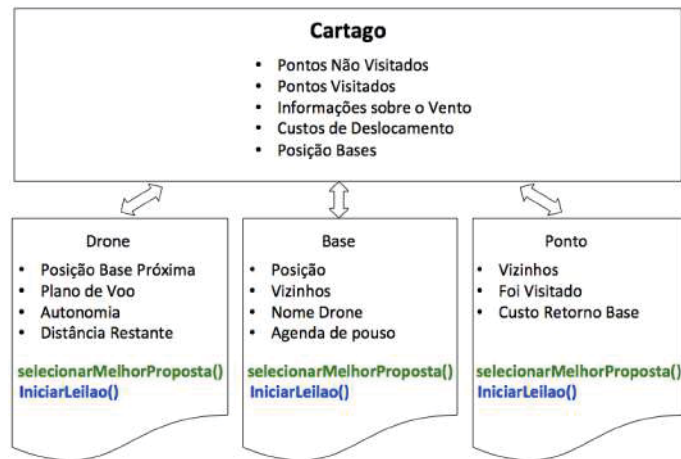
The negotiation model is called half-line, since it is oriented to the event of the drone returning to the base. When it returns to the base, it provides the system with the date the was collected during its mission and the model will restart the optimizing process and generating a new flight plan for each drone.

The Functional Specification is divided in Global Objects (network) and Local Objects (goal), as shown in Figure 3, where the Local Objective is set to an agent and the Global Objects is reached when all check points are taken by the agents. It is required that a sequence followed to maintain the efficiency, and that was approached by generating a hierarchical specification. That way, the optimization of the drone flight is only reached when the goals drone-call-for-proposal, base-call-for-proposal, checkpoint-call-for-proposal, checkpoint-accept-proposal, base-accept-proposal, drone accept-proposal are reached. When all goals are set it becomes possible to identify which agent is responsible for taking each check point.

The goal call-for-proposal are responsible for requesting a proposal from every agent. After receiving the proposal and choosing the best option, the agent executes the goal accept-proposal, which will update all agents in the chosen proposal.

Finally, we use the MOISE+ environment to link the Structural Specifications (Organize the Multiagent System) with the Functional Specification (activities of the Multiagent System). This is done through the Deontic Specification.

The Deontic Specification is responsible for selecting the drones to start the auction and pointing the communication from drones to bases to check points. It also sets these steps to take place in a logical order.



**Figura 4. Agents' Structure programmed in BDI, with believes in black, Desires in green and intentions in blue**

### 3.2. Cognitive Agents Proposal

The Multiagent System are cognitive agents, which allows them to know the organization they are inserted in and negotiate for themselves to achieve the global goals.

To program this agents we used the JASON platform, that is based in the Agent Speak(L) language and used BDI principles, allowing the agents to have well defined believes, desires and intentions.

The believes are information that all agents share from a date base of believes. This data base is updated every time a drone lands to recharge. There are a few data bases that can be used for this current project, and that is because the information is shared to all agents. We hereby highlight the Cartago repository that was used for this paper.

Based on the believes, Drones can track which check points have been mapped and which ones are still to be mapped, as well as wind data and the recharging bases position. This data is important for the drone agents to be able to calculate displacement battery costs from a check point to another.

Other information about the Multiagent System, such as how many Drones and bases are available don't need to be allocated as believes since the MOISE+ model provides the system with that data.

The agents' only goal is to take all check points. When the system reaches this condition the global goal is achieved. However, every agent has its on desires on how to achieve this goal. For example, the Drone has the wish of starting the auction with the base and this local goal is achieved when it receives all the proposals, although that does not mean the wish was achieved. If it is necessary, new desires of starting an auction with the base can be dynamic generated. The same happens with that base's desires towards the check points.

Figure 4 shows the programming structure of the cognitive agents and show the data base with the global believes that are shared between all agents (Cartago) and every agent with its on believes are shown in black, desires in green and intentions in blue.

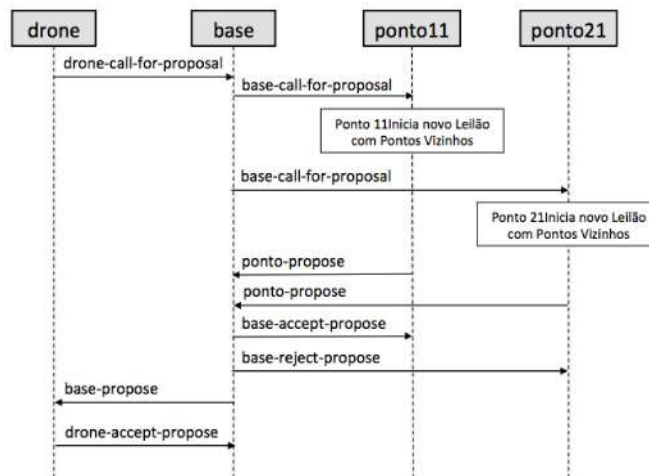


Figura 5. Sequence Diagram of FIPA Contract-Net auctions

### 3.3. FIPA Contract-NET applied to the Optimizing Drone Flight Model

The FIPA Contract-NET protocol for auctions base on agents programmed in JASON assures that the agents will be able to negotiate among themselves even if they are in different platforms.

Basically, the auction will start when a drone-agent is added to the drone-group or when this agent returns to the base. At the moment, the agent takes the goal of starting a call-for-proposal with a base-agent in which it has been inserted. The base-agent will then take the goal of starting a call-for-proposal sending a request for a proposal to the check point agents. These agents then start a recursive auction aiming to find the best flight plan for the drone.

When the best proposal is identified, the agent starts an accept-proposal goal, that updated the whole system with the new proposal, and gives the drone-agent the data with the check points to be covered.

Similarly, after the drone-agent finishes the auction it shares its flight plan with the other drone-agents that are in the base so that they can update their believes and the new flight plans get optimized.

Figure 5 shows a sequence diagram of the FIPA Contract-Net protocol use with a recursive optimization algorithm.

When a Drone-agent is included to the environment or it returns to the base, an auction is started with the base-agent, and that agent starts an auction with the all nearby checkpoint-agents. Every checkpoint-agent receives a proposal request starts an auction with its nearby checkpoint-agents. After receiving the proposal from its neighbors, it chooses the best according to the optimization algorithm, based on the cost of visiting that checkpoint, and returns its proposal to the base. The base will then receive the proposal from all nearby checkpoints, chooses the best one and send it to the drone-agent.

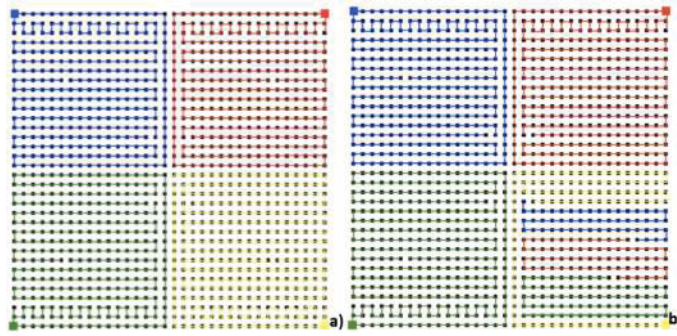


Figura 6. a) Original flight plan for four drone-agents. B) Flight plan after removing drone-agent 4 (in yellow), that happened after the first landing

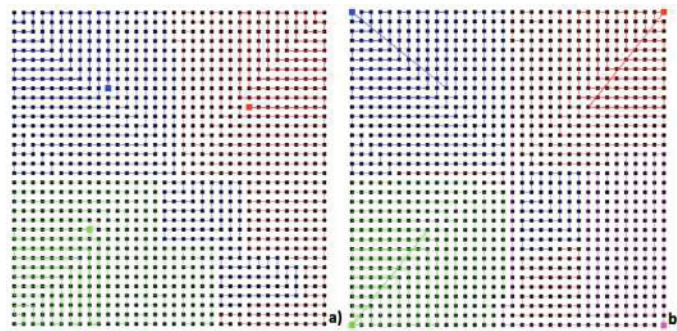


Figura 7. a) Original flight plan with 3 drones. b) Updated flight plan after including a fourth drone.

## 4. Results

### 4.1. Opening Tests of the Model

The Multiagent model focus on organization, with a structural, a functional and a Deontic Specification, that allows the model to work independently from its agents. That is important since in this structure, every new element already knows the models due to the data and can instantly take its roll. In the event of an element being removed, the resources that were previously allocated to that elements are redistributed to the remaining elements.

From an openness perspective, the model behaved as expected. The first test started with a flight plan for four drone-agents (Figure 6.a). After the first flight was run, the drone-agent 4, in yellow, was taken out of the system. The route that was left by that drone-agent was then taken over by the remaining drone-agents, as shown in Figure 6.b. In that figure, we can see that the yellow drone stayed partially in the system. That happens because that agent had already done its first flight, and therefore, had covered part of its flight plan.

The flight plan that was selected in this case used the optimizing algorithm proposed by [Brito et al. 2019].

The second test started with three drones. A fourth drone (in pink) was inserted in the system after the first flight. The checkpoints were then redistributed between all drone-agents. Figure 7.a shows the original flight plan for four drones and Figure 7.b shows the updated flight plan after including the fourth drone.

In both tests, with the inclusion or removal a drone-agent, the model was able to recover from this adverse event and a new distribution was provided to the system after the first landing. That reinstates that the proposed model is an open model. The average processing time for generating new flight plans was of 8 seconds.

Although in this paper only inclusion and removal of drone-agents were presented, the model also allows the inclusion and removal of check points. It has similar processing to those presented by drone-agents since after adding new check points, the drone-agents will identify the changes as soon as it returns to the base and it will start the auction for the new check points. The difference here are that check points can only be removed if they were not yet covered.

## 4.2. Dynamicity Model Tests

The seconds set of tests approached the dynamicity of the environment. The main point for this characteristic is the intelligence of the agents inserted.

All agents, from any group of agents, were programmed according to the BDI characteristic in the AgentSpeak(L) language using the framework JASON. These agents have believes, and some of these believes are shared through the Cartago tool. Among these believes are included the check points that have already being taken, the wind speed and direction, among others.

Therefore, for every change in the environment, the drone-agent has access to them when it returns to the base, starting a new process of recursive auction, considering all the new data.

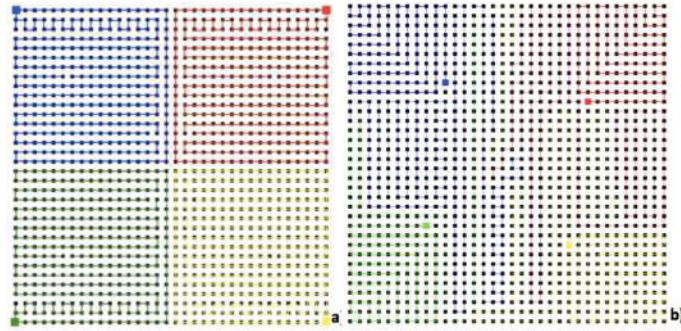
The first dynamicity test was performed by changing the wind speed and direction. Initially, a wind free plan was generated according to what is presented in Figure 8.a. After the first flight, a change in the wind was added to the system. The new settings for the wind were applied as its shown in Figure 8.b with wind in the northeast direction and a 50% increase to its intensity from the first flight.

Along with that change, the MultiDrone Simulator platform also allows other dynamicity tests, such as the early discharge of a drone battery. This scenario was also considered, and new flight plans were generated for all these changes, that are considered adverse.

This is possible due to two important characteristics of the system.

1. **Multiagent System:** The organization of the model is in an organizing base, the model is open for the inclusion and removal of agents at any moment. The only condition to stop the problem would be the reduction of the number of the agents in one of the groups to zero. Since that can not happen, the openness of the model works for any number of agents.;
2. **Cognitive Agents:** The dynamicity of the environment is approached by using cognitive agents that have a set of believes (Perception of the environment they are inserted), wishes (flight plan optimization), and intention (flight plans). That assures that the model can recover from adverse events and generate new flight plans every time a landing takes place..





**Figure 8. a) Original flight plan with four Drones. b) Adapted flight plan with a new wind direction/speed**

## 5. Conclusion

This paper presented a cooperative model that optimizes Drone Flights used in precision agriculture. The drones cooperate among each other, with the recharging bases and with the check points. This model proposal is to solve the limitation in using rotative wing UAVs in large fields.

An organizational structure of the elements was defined for the model based on MultiAgents systems. Since the agents are cognitive, their programming is based on mental states, with allows openness and dynamicity to the model.

For the proposed multi-agent system, the Framework MOISE+ was chosen in order to provide the model with openness, and a Framework JASON was used to configure the cognitive agents, enabling the model to work in dynamic environments. The communication between agents was developed using the FIPA Contract-NET protocol, assuring the model work for any number of agents that is different then zero.

The Multiagent System proposed was build and tested based on this paper's objectives. However, this structure can be used in other optimization problems that share a planning characteristic, such as optimizing the use of battery based electric vehicles in urban environment. The tests that were run using a MultiDrone Simulator, with the inclusion and the removal of elements from the system during the processing.

The cognitive agents were developed with JASON, using AgentSpeak(L) language. Although this is very different from common imperative programming languages, it was possible to program software agents, changing their mental states dynamically and allowing the agents to communicate. The Cartago Tool allowed the model to share believes between different agents and the dynamicity tests were developed in the MultiDrone Simulator platform using environment adverse changes, such as wind speed and direction and inclusion of drone-agents. In all tested cases, the model was able to reorganize the flight plans.

The communication between agents was build with a FIPA Contract-Net protocol for the development of auctions. This technique is of easy implementation in computational environments and since the model behaves as an auction, such protocol could be easily implemented and worked for a variety of number of elements.

## Referências

- Anthony, D., Elbaum, S., Lorenz, A., and Detweiler, C. (2014). On crop height estimation with UAVs. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Institute of Electrical & Electronics Engineers (IEEE).
- Brandão, A. S., Martins, F. N., and Soneguetti, H. B. (2015). A vision-based line following strategy for an autonomous uav. In *2015 IEEE 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. Institute of Electrical & Electronics Engineers (IEEE).
- Brito, R. C., Loureiro, J. F., Guedes, A., and Todt, E. (2019). Optimization system for dynamic flight planning for groups of drones using cooperation with mobile recharge bases by means of multiagent system and recursive auctions. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE.
- FIPA (2002). Fipa contract net interaction protocol specification. In [www.fipa.org/specs/fipa00029/SC00029H.pdf](http://www.fipa.org/specs/fipa00029/SC00029H.pdf). Acessado em 10 de julho de 2018.
- George, E. A., Tiwari, G., Yadav, R. N., Peters, E., and Sadana, S. (2013). UAV systems for parameter identification in agriculture. In *2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS)*. Institute of Electrical & Electronics Engineers (IEEE).
- Hübner, J. F., Bordini, R. H., and Vieira, R. (2004). Introdução ao desenvolvimento de sistemas multiagentes com jason. In *XII Escola de Informática da SBC, volume 2*. SBC.
- Hübner, J. F. and Sichman, J. S. (2003). Organização de sistemas multiagentes. In *Jornada de Atualização em Inteligência Artificial*. JAIA.
- Khaledi, M., Rovira-Sugranes, A., Afghah, F., and Razi, A. (2018). On greedy routing in dynamic UAV networks. In *2018 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*. IEEE.
- Maini, P. and Sujit, P. B. (2015). On cooperation between a fuel constrained UAV and a refueling UGV for large scale mapping applications. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE.
- Menegol, M. S., Hübner, J. F., and Becker, L. B. (2018). Evaluation of multi-agent coordination on embedded systems. In *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*, pages 212–223. Springer International Publishing.
- Puhl, L., Favarim, F., Pegorini, V., and Torrico, C. (2018). Desenvolvimento de robô trekking autônomo integrando sensoriamento inercial, geoespacial e visual. In *2018 Computer on the Beach*. Meditec.
- Saque, N., Brito, R. C., Goncalves, D. A., Favarim, F., and Todt, E. (2019). Optimized division of exploration areas in multi-robot systems considering static and dynamic charging stations. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE.

WESAAC<sub>2</sub>0<sub>5</sub>MA<sub>5</sub>BC.pdf

## **APÊNDICE B – ANEXOS**

## Regras da ANAC para uso de drones entram em vigor

### Norma cria condições para operações mais seguras

**Brasília, 2 de maio de 2017** – A Diretoria Colegiada da ANAC aprovou, nesta terça-feira (02/05), o regulamento especial para utilização de aeronaves não tripuladas, popularmente chamadas de drones. A norma (Regulamento Brasileiro de Aviação Civil Especial – RBAC –E nº 94) estará publicada no Diário Oficial da União desta quarta-feira (03/05).

O objetivo é tornar viáveis as operações desses equipamentos, preservando-se a segurança das pessoas. A instituição das regras também contribuirá para promover o desenvolvimento sustentável e seguro para o setor.

O normativo foi elaborado levando-se em conta o nível de complexidade e de risco envolvido nas operações e nos tipos de equipamentos. Alguns limites estabelecidos no novo regulamento seguem definições de outras autoridades de aviação civil como [Federal Aviation Administration \(FAA\)](#), [Civil Aviation Safety Authority \(CASA\)](#) e [European Aviation Safety Agency \(EASA\)](#), reguladores dos Estados Unidos, Austrália e da União Europeia, respectivamente.

A partir de agora, as operações de aeronaves não tripuladas (de uso recreativo, corporativo, comercial ou experimental) devem seguir as novas regras da ANAC, que são complementares aos normativos de outros órgãos públicos como o [Departamento de Controle do Espaço Aéreo \(DECEA\)](#) e da [Agência Nacional de Telecomunicações \(ANATEL\)](#).

O regulamento sobre aeronaves não tripuladas foi amplamente discutido com a sociedade, associações e empresas interessadas, bem como com outros órgãos públicos. Foram realizados dois workshops e técnicos da Agência participaram de diversos eventos. A proposta ficou em audiência pública (AP nº 13/2015) por 60 dias, com sessão presencial. Foram recebidas 277 contribuições.

### Assessoria de Comunicação Social da ANAC

Gerência Técnica de Relações com a Imprensa

Telefones: (61) 3314-4491 / 4493 / 4494 / 4496 / 4498 / 4642

Plantão de Imprensa: (61) 99112-8099

E-mail: [jornalismo@anac.gov.br](mailto:jornalismo@anac.gov.br)

[www.anac.gov.br](http://www.anac.gov.br)

O novo regulamento da ANAC dividiu as aeronaves não tripuladas em aeromodelos, drones usados para fins recreativos, e aeronaves remotamente pilotadas (RPA), drones utilizados para operações comerciais, corporativas ou experimentais.

## DRONES

O termo "drone" é utilizado popularmente para descrever qualquer aeronave (ou mesmo outro tipo de veículo) que possua alto grau de automatismo. No entanto, como não há uma definição formal para o termo, a regulamentação da Agência não utiliza essa nomenclatura, mas sim "aeromodelos" e "aeronaves remotamente pilotadas" (RPA). O que diferencia essas duas categorias de drones é a sua finalidade:



### Aeromodelo

É toda aeronave não tripulada com finalidade de recreação.

### Aeronave Remotamente Pilotada (RPA)

É uma aeronave não tripulada pilotada a partir de uma estação de pilotagem remota que tenha qualquer outra finalidade que não seja recreativa, tais como comercial, corporativa e experimental.



Pela regra geral, os drones com mais de 250g só poderão voar em áreas distantes de terceiros (no mínimo 30 metros horizontais), sob total responsabilidade do piloto operador e conforme regras de utilização do espaço aéreo do DECEA. Caso exista uma barreira de proteção entre o equipamento e as pessoas a distância especificada não precisa ser observada.

Para voar com drones com mais de 250g perto de pessoas é necessário que elas concordem previamente com a operação, ou seja, a pessoa precisa saber e concordar com o voo daquele equipamento nas proximidades onde se encontra.



## IMPORTANTE!

As operações totalmente autônomas desses equipamentos, ou seja, naquelas onde o piloto remoto não é capaz de intervir, continuam proibidas no país. Essas operações diferem-se das automatizadas, nas quais o piloto remoto pode interferir em qualquer ponto.

## Classificação de drones

Os drones de uso comercial, corporativo ou experimental (RPA) foram categorizadas em três classes, de acordo com o peso máximo de decolagem do equipamento.

Classe	Peso Máximo de Decolagem	Exigências de Aeronavegabilidade
<b>Classe 1</b>	Acima de 150kg	A regulamentação prevê que equipamentos desse porte sejam submetidos a processo de certificação similar ao existente para as aeronaves tripuladas, promovendo ajustes dos requisitos de certificação ao caso concreto. Esses drones devem ser registrados no Registro Aeronáutico Brasileiro e identificados com suas marcas de nacionalidade e matrícula.
<b>Classe 2</b>	Acima de 25 kg e abaixo ou igual a 150 kg	O regulamento estabelece os requisitos técnicos que devem ser observados pelos fabricantes e determina que a aprovação de projeto ocorrerá apenas uma vez. Além disso, esses drones também devem ser registrados no Registro Aeronáutico Brasileiro e identificados com suas marcas de nacionalidade e matrícula.
<b>Classe 3</b>	Abaixo ou igual a 25 kg	<p>A norma determina que as RPA Classe 3 que operem além da linha de visada visual (BVLOS) ou acima de 400 pés (120m) deverão ser de um projeto autorizado pela ANAC e precisam ser registradas e identificadas com suas marcas de nacionalidade e matrícula.</p> <p>Drones dessa classe que operarem em até 400 pés (120m) acima da linha do solo e em linha de visada visual (operação VLOS) não precisarão ser de projeto autorizado, mas deverão ser cadastradas na ANAC por meio do sistema SISANT, apresentando informações sobre o operador e sobre o equipamento.</p> <p>Os drones com até 250g não precisam ser cadastrados ou registrados, independentemente de sua finalidade (uso recreativo ou não).</p>

## Idade mínima para pilotagem

Para pilotar aeronaves não tripuladas RPA, os pilotos remotos e observadores (que auxiliam o piloto remoto sem operar o equipamento) devem ter no mínimo 18 anos. Para pilotar aeromodelos não há limite mínimo de idade.

## Cadastro

O cadastro dos drones (aeromodelos ou RPA Classe 3) com peso máximo de decolagem superior a 250g é obrigatório e deve ser feito pelo Sistema de Aeronaves Não Tripuladas (SISANT) da ANAC pelo endereço [sistemas.anac.gov.br/sisant](http://sistemas.anac.gov.br/sisant). O número de identificação gerado na certidão de cadastro deve estar acessível na aeronave ou em local que possa ser facilmente acessado, de forma legível e produzido em material não inflamável.



## Registro de voos

Os voos com aeromodelo e RPA Classe 3 não precisam ser registrados. O voos com as demais aeronaves não tripuladas devem ser registrados.

## Licença, Habilitação e Certificado Médico Aeronáutico

Operadores de aeromodelos e de aeronaves RPA de até 250g são considerados licenciados, sem necessidade de possuir documento emitido pela ANAC desde que não pretendam usar equipamento para voos acima de 400 pés.

Serão obrigatórias licença e habilitação emitidas pela ANAC apenas para pilotos de operações com aeronaves não tripuladas RPA das classes 1 (peso máximo de decolagem de mais de 150 kg) ou 2 (mais de 25 kg e até 150 kg) ou da classe 3 (até 25 Kg) que pretendam voar acima de 400 pés.

Pilotos remotos de aeronaves não tripuladas RPA das classes 1 (mais de 150 kg) e 2 (mais de 25 kg e até 150 kg) deverão possuir ainda o Certificado Médico Aeronáutico (CMA) emitido pela ANAC ou o CMA de terceira classe do DECEA.

## Saiba mais

**Operação BVLOS** – Operação na qual o piloto não consegue manter o drone dentro de seu alcance visual, mesmo com a ajuda de um observador.

**Operação VLOS** – Operação na qual o piloto mantém o contato visual direto com o drone (sem auxílio de lentes ou outros equipamentos).

**Operação EVLOS** – Operação na qual o piloto remoto só é capaz de manter contato visual direto com o drone com auxílio de lentes ou de outros equipamentos e precisa do auxílio de observadores de drone.







## Documentos obrigatórios durante as operações

Nas operações realizadas com aeronaves não tripuladas (aeromodelos e RPA) com peso máximo de decolagem superior a 250g, os operadores deverão portar documentos obrigatórios. Dentre eles estão o manual de voo, documento de avaliação de risco e apólice de seguro. Leia mais sobre os documentos exigidos pela ANAC para cada uma das classes no guia de Perguntas frequentes.

## IMPORTANTE!

Mais documentos poderão ser necessários de acordo com outros órgãos competentes. Consulte as normas do DECEA e da ANATEL sobre o assunto.

## Seguro

É obrigatório possuir seguro com cobertura contra danos a terceiros nas operações de aeronaves não tripuladas de uso não recreativo acima de 250g (exceto as operações de aeronaves pertencentes a entidades controladas pelo Estado).



## Transporte de cargas

Não podem ser transportados pessoas, animais, artigos perigosos ([RBAC nº 175/2009](#)) e outras cargas proibidas por autoridades competentes. Artigos perigosos poderão ser transportados quando destinados a lançamentos relacionados a atividades de agricultura, horticultura, florestais ou outras definidas pelo novo regulamento.

Poderão ser transportados equipamentos eletrônicos que contenham baterias de lítio necessárias para seu funcionamento, desde que sejam destinadas para uso durante o voo, tais como câmeras fotográficas, filmadoras, computadores etc. Artigos perigosos requeridos para operação do equipamento também poderão ser transportados. As regras referentes aos artigos perigosos não se aplicam aos drones controlados pelo Estado (sob total responsabilidade das entidades e em cumprimento ao [RBAC nº 175/2009](#)).



## Uso de drones por órgãos de segurança pública

As operações de drone por órgãos de segurança pública, de polícia, de fiscalização tributária e aduaneira, de combate a vetores de transmissão de doenças de defesa civil e do corpo de bombeiros, ou de operador a serviço de um desses, são permitidas pela ANAC sem observar os critérios de distanciamento das áreas distantes de terceiros. Essas operações devem ocorrer sob total responsabilidade do órgão ou operador e possuir avaliação de risco operacional. Devem também obedecer as regras de utilização do espaço aéreo estabelecidas pelo DECEA.



## Locais de pousos e decolagens de drones

Pousos e decolagens também podem ser feitos em áreas distantes de terceiros e desde que não haja proibição de operação no local escolhido. A operação de aeronaves não tripuladas em aeródromos só pode ocorrer se for expressamente autorizada pelo operador aeroportuário, podendo a ANAC estabelecer condições específicas.



### Fiscalização

Os órgãos de segurança pública farão a fiscalização de drones no dia-a-dia. Casos de infrações configuradas como contravenção penal ou crime serão tratados por esses órgãos. Por parte da ANAC, a fiscalização será incluída no programa de vigilância contínua e as denúncias recebidas serão apuradas administrativamente de acordo com as sanções previstas no Código Brasileiro de Aeronáutica (Lei nº 7.565/86).

Outros órgãos farão a fiscalização de acordo com os aspectos relacionados às suas competências, como utilização do espaço aéreo (DECEA) e de radiofrequência (ANATEL).

## Penalidades previstas pela ANAC

Irregularidades em relação ao cumprimento da norma são passíveis de sanções previstas no Código Brasileiro de Aeronáutica (Lei nº 7.565/86). A descrição das infrações e das penalidades pode ser consultada na Resolução nº 25/2008. Cautelarmente, a ANAC poderá suspender temporariamente as operações nos casos de suspeita ou evidência de descumprimento do regulamento que impactem o nível de risco da operação.

## Penalidades previstas por outros órgãos

Outras sanções também estão previstas nas legislações referentes às responsabilizações nas esferas civil, administrativa e penal, com destaque à inviolabilidade da intimidade, da vida privada, da honra e da imagem das pessoas.

O Código Penal prevê, em seu Art. 261, prevê pena de reclusão de dois a cinco anos para quem expuser a perigo embarcação ou aeronave, própria ou alheia, ou praticar qualquer ato tendente a impedir ou dificultar navegação marítima, fluvial ou aérea.

O Código Penal também tipifica a exposição de pessoas a risco, em seu Art. 132, que prevê pena de detenção de três meses a um ano (ou mais se o crime for considerado mais grave) nos casos em que se coloquem em perigo direto ou iminente a vida ou à saúde terceiros.

Pela Lei das Contravenções Penais, dirigir aeronave sem estar devidamente licenciado pode gerar pena de prisão simples (quinze dias a três meses) e pagamento de multa. Pelo Art. 35 da mesma lei, praticar acrobacias ou fazer voos baixos, fora da zona permitida em lei, bem como fazer descer a aeronave fora de lugares destinados a essa finalidade, também pode gerar prisão simples (15 dias a três meses) e multa.

Outras penalidades poderão ser aplicadas conforme regras de outros órgãos públicos como a ANATEL, o DECEA e o Ministério da Defesa.

### Resumo da regulamentação da ANAC

	RPAS Classe 1	RPAS Classe 2	RPAS Classe 3	Aeromodelos
Registro da aeronave?	Sim	Sim	BVLOS: Sim VLOS: Sim <sup>1</sup>	Sim <sup>1</sup>
Aprovação ou autorização do projeto?	Sim	Sim <sup>2</sup>	Apenas BVLOS ou acima de 400 pés <sup>2</sup>	Não
Limite de idade para operação?	Sim	Sim	Sim	Não
Certificado médico?	Sim	Sim	Não	Não
Licença e habilitação?	Sim	Sim	Apenas para operações acima de 400 pés	Apenas para operações acima de 400 pés
Local de operação	A distância da aeronave não tripulada NÃO poderá ser inferior a 30 metros horizontais de pessoas não envolvidas e não anuentes com a operação. O limite de 30 metros não precisa ser observado caso haja uma barreira mecânica suficientemente forte para isolar e proteger as pessoas não envolvidas e não anuentes. Esse limite não é aplicável para operações por órgão de segurança pública, de polícia, de fiscalização tributária e aduaneira, de combate a vetores de transmissão de doenças, de defesa civil e/ou do corpo de bombeiros, ou operador a serviço de um destes.			

<sup>1</sup> Todos os aeromodelos acima de 250 gramas e RPA entre 250 gramas e 25 kg que se destinem a operações na linha de visada visual (VLOS) até 400 pés acima do nível do solo, devem ser cadastrados por meio de ferramenta online disponível no endereço <https://sistemas.anac.gov.br/sisant>.  
<sup>2</sup> Para todos os RPAS Classe 2 e os RPAS Classe 3 que se destinam a operações além da linha de visada visual (BVLOS) ou acima de 400 pés, o fabricante pode optar pelo processo de certificação de tipo estabelecido no RBAC nº 21 ou pela autorização de projeto na Subparte E do RBAC-E nº 94.

## Mais informações

[Perguntas Frequentes](#)

[www.anac.gov.br/drones](http://www.anac.gov.br/drones)

[Orientações para Usuários de Drones](#)

[Página sobre drones do Ministério de Transportes, Portos e Aviação Civil](#)