

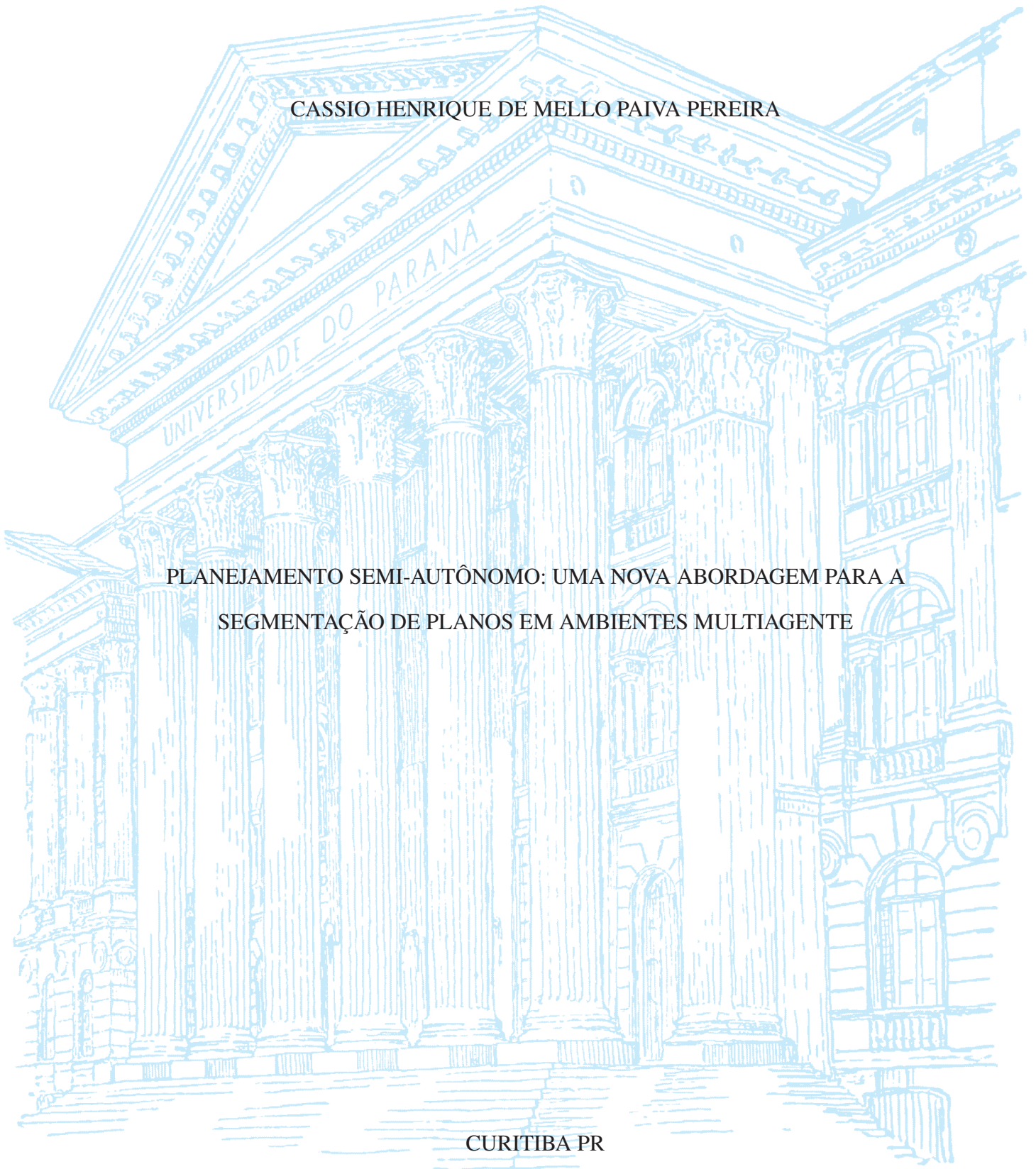
UNIVERSIDADE FEDERAL DO PARANÁ

CASSIO HENRIQUE DE MELLO PAIVA PEREIRA

PLANEJAMENTO SEMI-AUTÔNOMO: UMA NOVA ABORDAGEM PARA A
SEGMENTAÇÃO DE PLANOS EM AMBIENTES MULTIAGENTE

CURITIBA PR

2020



CASSIO HENRIQUE DE MELLO PAIVA PEREIRA

PLANEJAMENTO SEMI-AUTÔNOMO: UMA NOVA ABORDAGEM PARA A
SEGMENTAÇÃO DE PLANOS EM AMBIENTES MULTIAGENTE

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

Área de concentração: *Ciência da Computação*.

Orientador: Prof. Dr. Marcos Alexandre Castilho.

CURITIBA PR

2020

Catálogo na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

P436p Pereira, Cassio Henrique de Mello Paiva
 Planejamento semi-autônomo: uma nova abordagem para a segmentação de planos em ambientes multiagente [recurso eletrônico] / Cassio Henrique de Mello Paiva Pereira. – Curitiba, 2020.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2020.

Orientador: Marcos Alexandre Castilho.

1. Inteligência artificial. 2. Planejamento. I. Universidade Federal do Paraná. II. Castilho, Marcos Alexandre. III. Título.

CDD: 006.3

Bibliotecária: Vanusa Maciel CRB- 9/1928

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 Dissertação de Mestrado de **CASSIO HENRIQUE DE MELLO PAIVA PEREIRA** intitulada: **PLANEJAMENTO SEMI-AUTÔNOMO: UMA NOVA ABORDAGEM PARA A SEGMENTAÇÃO DE PLANOS EM AMBIENTES MULTIAGENTE**, sob orientação do Prof. Dr. MARCOS ALEXANDRE CASTILHO, 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 mestre 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 Março de 2020.



MARCOS ALEXANDRE CASTILHO

Presidente da Banca Examinadora (UNIVERSIDADE FEDERAL DO PARANÁ)



FABIANO SILVA

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)



BRUNO CESAR RIBAS

Avaliador Externo (UNIVERSIDADE DE BRASÍLIA)



AGRADECIMENTOS

Gostaria de agradecer a todos que viabilizaram esta conquista.

Ao Prof. Dr. Marcos Alexandre Castilho, pelo apoio e orientação desde que me interessei em Inteligência Artificial, ainda na graduação.

Prof. Dr. Bruno César Ribas e Prof. Dr. Fabiano Silva, pelo extenso corpo de sugestões feitas para a melhoria deste trabalho.

A todos os professores do curso de Mestrado, que compartilharam seu conhecimento dos mais diversos aspectos da ciência.

Aos colegas do Laboratório de Inteligência Artificial e Métodos Formais, que sempre estiveram dispostos a responder minhas dúvidas mais variadas e tangenciais.

Aos meus familiares e amigos, pelo contínuo suporte e compreensão.

RESUMO

Nesta dissertação é apresentada uma extensão ao planejador HEART publicado em (Gréa et al., 2018). O planejador HEART apresenta um mecanismo para a solução do problema de planejamento através do uso de rede de tarefas hierárquicas e ordenação parcial controlada por vínculos causais, permitindo a apresentação de planos intermediários em casos de problemas sem solução ou com restrição de tempo de planejamento. Esta extensão inova ao apresentar um mecanismo através do qual a aplicação do planejador passa a ser possível em ambientes multiagente distribuídos. Este novo mecanismo se aproveita da capacidade de geração de planos intermediários para a criação de subproblemas, permitindo a resolução de problemas mais complexos por agentes de menor capacidade sem comprometer a característica de execução em tempo real. É feita a revisão de conceitos fundamentais sobre planejamento em inteligência artificial, redes hierárquicas de tarefas e planejamento multiagente. Após a revisão é apresentada a base conceitual para esta extensão e é descrita a nova falha, denominada falha de atribuição externa. É então provado que esta falha e seu resolvidor mantém as características de corretude e completude do planejador, e são apresentados resultados da análise de sua inclusão, mostrando que a inclusão não compromete a performance e pode ser utilizada para o ambiente multiagente proposto.

Palavras-chave: Planejamento em IA. Rede de tarefas hierárquicas. Planejamento multiagente.

ABSTRACT

This dissertation presents an extension to the HEART planner, first published in (Gréa et al., 2018). The HEART planner has a mechanism to solve the planning problem by using hierarchical task networks and partial ordering controlled by causal links, allowing for intermediate plans in case of unsolvable problems or planning under time restrictions. This extension innovates by introducing a tool that allows for the usage of the planner in the distributed multiagent environment. This tool exploits the intermediate plan generation capability for creating subproblems, allowing less capable agents to solve higher complexity problems without sacrificing the real-time execution. A revision of fundamental concepts about artificial intelligence planning, hierarchical task networks and multiagent planning is made. After the revision, the conceptual basis for the extension is presented, followed by the inclusion of a new flaw, noted as external attribution flaw. It is then proved that this new flaw and its associated resolver maintain the planner properties of soundness and completeness, and results of its inclusion are analyzed, showing that it does not compromise performance and is thus applicable in the proposed multiagent environment.

Keywords: AI planning. Hierarchical task network. Multiagent planning.

LISTA DE TABELAS

5.1	Instanciação de fluentes	40
5.2	Refinamento (n=4)	40
5.3	Refinamento (n=3)	40
5.4	Refinamento (n=2)	41
5.5	Tempo de processamento após atribuição externa	41

LISTA DE ACRÔNIMOS

AIPS	AI Planning and Scheduling Conference
CASPER	Continuous Activity Scheduling, Planning, Execution, and Replanning
DINF	Departamento de Informática
HEART	HiErarchical Abstraction for Real-Time
HTN	Hierarchical Task Network
IA	Inteligência Artificial
ICAPS	International Conference on Automated Planning and Scheduling
IPC	International Planning Competition
IPEM	Integrated Planning, Execution and Monitoring
LRTA*	Learning Realtime A*
PDDL	Planning Domain Definition Language
POCL	Partial Order Causal Link
PPGINF	Programa de Pós-Graduação em Informática
PSP	Plan Space Planning
STRIPS	Stanford Research Institute Problem Solver
RAX	Remote Agent eXperiment
UFPR	Universidade Federal do Paraná
XAI	eXplainable Artificial Intelligence

LISTA DE SÍMBOLOS

\mathcal{D}, \mathcal{P}	Domínio e problema de planejamento
$pre(a), eff(a)$	Pré-condições e efeitos da ação a
$met(a)$	Métodos que decompõem a ação a
$age(a)$	Agentes capazes de realizar a ação a
$s \xrightarrow{P} w$	Vínculo da ação s como precedente de w pela causa P
$L^\pm(a)$	Vínculos causais de entrada (L^-) e de saída (L^+) do passo a . Não sinalizado indica todos os vínculos adjacentes
$\phi^\pm(l)$	Função de incidência para planos parciais, onde ϕ^- retorna passos de entrada e ϕ^+ retorna passos de saída para um vínculo l . Não sinalizado retorna o par correspondente ao vínculo l
$a_a > a_s$	Restrição de ordenação, onde o passo a_a antecede o passo a_s
$\downarrow_f a_n$	Objetivo parcial, fluente f não suportado pelo passo a_n
$a_b \otimes l_t$	Ameaça, onde ação a_b quebra o vínculo l_t
$a \triangleright_{\pi}^\pm a'$	Transposição dos vínculos de entrada (\triangleright^-) e de saída (\triangleright^+) da ação a' para a ação a
A_a^n	Conjunto de ações próprias da entidade a , até o nível de abstração n
$lv(a)$	Nível de abstração da entidade a
$a_n \oplus^m$	Decomposição da ação composta a_n utilizando o método m
$a_n \oplus^{m,e}$	Atribuição externa da ação composta a_n para resolução pelo agente e através do método m
$var : exp$	Separador a ser lido como var "tal que" exp
$\llbracket exp \rrbracket$	Colchetes de Iverson. 0 caso exp seja uma expressão falsa, 1 caso contrário

SUMÁRIO

1	INTRODUÇÃO	11
1.1	ORGANIZAÇÃO DO TRABALHO	11
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	PLANEJAMENTO EM INTELIGÊNCIA ARTIFICIAL	12
2.1.1	Planejamento Clássico	12
2.1.2	Controle de recursos	13
2.1.3	Controle temporal	14
2.1.4	Incerteza	15
2.2	EXTENSÕES CONCEITUAIS	16
2.2.1	Ordenação parcial por vínculos causais	16
2.2.2	Planejamento por rede hierárquica de tarefas	17
2.2.3	Planejamento multiagente	18
3	REVISÃO BIBLIOGRÁFICA	22
3.1	SISTEMA <i>CASPER</i>	22
3.1.1	Planejamento em tempo real	22
3.1.2	Planejamento em mundo aberto	22
3.1.3	Planejamento contínuo	23
3.1.4	Planejamento em múltiplos níveis	23
3.2	COMPETIÇÕES <i>ICAPS</i>	24
3.3	PLANEJADOR <i>HEART</i>	25
3.3.1	Planejamento explicável	25
3.3.2	Princípios de funcionamento	26
3.3.3	Definições adicionais ao <i>HEART</i>	30
4	PLANEJAMENTO SEMI-AUTÔNOMO	33
4.1	MOTIVAÇÃO PARA O NOVO CONCEITO	33
4.2	EMBASAMENTO TEÓRICO	33
4.3	ANÁLISE TEÓRICA	36
4.3.1	Aplicabilidade teórica e expansões possíveis	38
5	EXPERIMENTOS	39
5.1	ESTRUTURAÇÃO E METODOLOGIA	39
5.2	RESULTADOS	39
5.2.1	Instanciação de fluentes	39
5.2.2	Tempo de processamento do supervisor e paralelização	40
5.2.3	Tempo de processamento do executor	41

5.2.4	Análise	42
6	CONCLUSÃO	43
	REFERÊNCIAS	45

1 INTRODUÇÃO

A tomada de decisões é necessária em diversas situações, e ainda mais relevante quando existe um objetivo a ser atingido. Quando este objetivo é distante ou complexo se torna importante também a criação de passos a serem seguidos que permitam atingir este objetivo. A esta sequência de passos pode ser dado o nome de *plano*, e o processo que define este plano é o *planejamento*. Quando este planejamento ocorre através do uso de ferramentas automatizadas e técnicas de inteligência artificial então temos a área que será abordada neste trabalho, através do *Problema de Planejamento em Inteligência Artificial*.

O problema de planejamento em inteligência artificial como será abordado tem sua origem em (Newell et al., 1959), que propôs um programa que separa o conhecimento do problema (hoje chamado de *domínio*) da estratégia para obter sua solução (cujo mecanismo usa hoje o nome *planejador*). Este tipo de planejamento é denominado *planejamento automatizado independente de domínio*, significando que o domínio pode ser modificado sem que haja necessidade de alterar o código do planejador. A área de estudo deste problema teve tempos de estagnação, mas desde a apresentação do planejador STRIPS (Fikes e Nilsson, 1971) apresenta uma ressurgência e se mantém ativa até o tempo atual. Estes avanços usam como base a variante para o problema definida como *planejamento clássico* (Ghallab et al., 2004), que mesmo simplificada é provadamente complexa (PSPACE-completa, conforme (Bylander, 1994)).

Este trabalho apresenta uma nova abordagem para a solução do problema de planejamento como utilizado numa variante chamada de *planejamento hierárquico* (Erol et al., 1994b) utilizando inspiração no problema de *planejamento multiagente* (Durfee, 2001), ambos conceitos explicados extensivamente a seguir. Esta nova abordagem permite a agentes de capacidade reduzida participação no processo de planejamento através da divisão do problema original em subproblemas tratáveis, utilizando a estrutura hierárquica como base para a divisão.

1.1 ORGANIZAÇÃO DO TRABALHO

O capítulo 2 apresenta uma revisão de teorias que compõe a área de planejamento em inteligência artificial, assim como extensões conceituais ao problema. O capítulo 3 revisa a bibliografia relacionada ao problema de planejamento sendo abordado, e explicita definições que embasam o conceito a ser apresentado. O capítulo 4 introduz o conceito de planejamento semi-autônomo, conceitua e contextualiza uma metodologia aplicável a este novo conceito e apresenta sua aplicabilidade teórica. O capítulo 5 apresenta a análise de viabilidade prática do mecanismo apresentado. No capítulo 6 são então apresentadas considerações finais sobre o trabalho e possibilidades de pesquisa futura.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentadas as técnicas que servem de fundamento teórico ao novo método sendo apresentado, junto a seus formalismos. A seção 2.1 apresenta o problema de planejamento clássico e suas extensões mais relevantes ao trabalho, enquanto a seção 2.2 apresenta extensões conceituais ao problema, que podem ser aplicadas como alternativas de resolução ou com o intuito de utilizar planejamento para resolver problemas mais complexos.

2.1 PLANEJAMENTO EM INTELIGÊNCIA ARTIFICIAL

O planejamento em Inteligência artificial é definido informalmente (Weld, 1999) como a tarefa de encontrar uma *solução* para um problema de planejamento, utilizando das seguintes estruturas:

- Uma descrição das informações relevantes ao estado inicial do mundo, utilizando alguma linguagem formal;
- Uma descrição do conjunto de informações desejadas ao estado objetivo;
- A descrição das ações disponíveis para a modificação do estado do mundo, com suas pré-condições e efeitos.

A partir desta entrada é gerado como resultado uma sequência de ações que, quando aplicadas a qualquer mundo que conforme ao estado inicial descrito o modificam para um estado que possui o conjunto do estado objetivo. Tal sequência é chamada de *plano*, e é a solução para este problema de planejamento.

2.1.1 Planejamento Clássico

O tipo de problema de planejamento mais simples que obedece esta estrutura é chamado de planejamento clássico, que pode ser definido da seguinte forma, adaptada de (Ghallab et al., 2004).

Definição 2.1. (Literais). Partindo de um conjunto de variáveis booleanas V , definimos o conjunto de literais L sobre V como $L = V \cup \{\neg v \mid v \in V\}$, onde o complemento \bar{l} de um literal l é definido por $\bar{\bar{v}} = \neg v$, e $\overline{\neg v} = v$, para $v \in V$.

Definição 2.2. (Ações). Ações são operações no problema de planejamento, representadas pela dupla $a = \langle P, E \rangle$, onde P é o conjunto de literais necessários para efetuar a ação (suas pré-condições), enquanto E é o conjunto de literais de seus efeitos.

Definição 2.3. (Problema de planejamento). O problema de planejamento é então definido pela quádrupla $\langle V, I, O, G \rangle$, onde:

V - Conjunto de variáveis proposicionais relevantes ao problema, que vão compor o estado do mundo.

I - Conjunto que representa o estado inicial do mundo, tal que $I : V \rightarrow \{0, 1\}$.

O - Conjunto de ações disponíveis para a alteração do estado do mundo.

G - Conjunto de literais que representam o objetivo a ser atingido. Para considerar o problema como satisfeito é necessário que o conjunto G seja um subconjunto do estado atual do mundo.

Adicionalmente, para um problema de planejamento em inteligência artificial ser definido como clássico, sua estrutura deve também obedecer às seguintes restrições:

- A0: Quantidade de estados do problema finita;
- A1: Observabilidade total;
- A2: Sistema determinístico;
- A3: Estático, apenas a tomada de ações afeta o estado atual;
- A4: Objetivo descrito apenas com literais positivos;
- A5: Plano finito e linearmente ordenado;
- A6: Passagem de tempo implícita;
- A7: Planejador desconsidera alterações posteriores ao problema.

Estas restrições tem por objetivo limitar o escopo de características representáveis, visando tornar estes problemas tratáveis por programas de análise automatizada e permitir sua aplicação como método de obtenção de soluções em tempo hábil. O amadurecimento da área de pesquisa tornou possível a reintrodução de algumas destas características ao problema, aumentando a gama de situações em que o uso de planejamento é aplicável.

Um problema de planejamento é classificado como não clássico quando suas características violam ao menos uma das suposições necessárias ao planejamento clássico. As seções a seguir darão foco a algumas variantes de planejamento não clássico relevantes a este trabalho, assim como exemplos dos avanços que tais extensões tornam possível.

2.1.2 Controle de recursos

Uma expansão ao planejamento clássico que permite resolver problemas em que existe a necessidade de controle numérico, assim como restringir numericamente mínimos e máximos para atingir o estado objetivo. A partir desta expansão, criada para a 3^a edição da competição *IPC* e incorporada na linguagem PDDL 2.1 (Fox e Long, 2003), foi possível o controle qualitativo de planos e sua otimização de acordo com as métricas de cada domínio.

Esta expansão consiste em complementar as variáveis lógicas com variáveis numéricas, que juntas passam a compor os estados do mundo. Estas novas variáveis são utilizadas de maneira análoga às variáveis lógicas, sendo utilizadas em pré-condições e manipuladas através das ações tomadas pelo agente, podendo ao final do plano estabelecer controle comparativo entre planos diversos.

Neste ponto tornou-se possível uma análise qualitativa mais complexa dos planos e dos planejadores¹, e também é gerado o conceito de plano ótimo, em que alguma variável numérica maximizada (ou minimizada) representa o melhor plano possível para cada cenário. A capacidade de representação adicional também permitiu a implementação de planejadores em ambientes mais realistas, utilizando técnicas de Programação Linear ou similares para resolver as questões de controle de valores numéricos.

¹Métricas anteriores consistiam usualmente em tempo real gasto na geração do plano ou em quantidade de ações tomadas até o objetivo.

Definição 2.4. (Planejamento com controle de recursos). Um problema de planejamento com controle de recursos pode ser definido como a quádrupla $\langle V, I, O, G \rangle$, onde:

V - Passa a corresponder aos conjuntos V_P e V_N , contendo respectivamente as variáveis proposicionais do planejamento clássico e as variáveis numéricas, usualmente pertencentes ao conjunto dos números racionais \mathbb{Q} . Uma ação deste problema possui pré-condições e efeitos em ambos conjuntos, com a porção numérica podendo ser expressa utilizando operadores numéricos, normalmente através de desigualdades. Os estados passam então a ser definidos como $S = \{(\alpha, \beta) \mid \alpha : V_P \rightarrow \{0, 1\}, \beta : V_N \rightarrow \mathbb{Q}\}$.

O - Para conjuntos adicionais de funções (de aridade maior ou igual a 1) sobre \mathbb{Q} , C e \mathcal{E} , é definida $f \in C$ de aridade n , n variáveis $v_1, \dots, v_n \in V_N$ e um operador relacional $r \in \{=, \neq, <, \leq, >, \geq\}$, definimos como condições numéricas sobre C a operação escrita $f(v_1, \dots, v_n)r0$, que será satisfeita se, e somente se, para o estado (α, β) , a afirmação $f(\beta(v_1), \dots, \beta(v_n))r0$ for verdadeira. Definimos como efeito numérico sobre \mathcal{E} a função escrita $v_1 \leftarrow f(v_1, \dots, v_n)$, que afeta a variável v_1 . Neste novo formalismo, uma ação $\langle P, E \rangle$ incorpora os conjuntos C e \mathcal{E} sobre P e E , respectivamente.

I, G - A composição dos estados inicial e final não sofrem alterações diretas pois o plano interage com as variáveis numéricas e recursos durante a aplicação de ações, e seus valores são armazenados em separado do conjunto de estados. Valores iniciais de recursos são definidos junto ao conjunto V_N , ou em uma ação de preparação que define o primeiro passo de um plano. Vale ressaltar que o estado final pode estabelecer requisitos mínimos e máximos para os valores de variáveis do conjunto V_N , além das condições proposicionais.

2.1.3 Controle temporal

Também introduzido na linguagem PDDL 2.1, o controle temporal corresponde a remoção da suposição de ações com efeitos instantâneos e à abstração do tempo gasto com a execução do plano.

No planejamento clássico ações são tomadas de maneira instantânea e seus efeitos são válidos por tempo indeterminado, até que outra ação os altere. A partir desta expansão estas características são substituídas por um controle explícito de tempo durante o planejamento, que permite às ações especificar se pré-condições devem ser atendidas apenas durante seu início ou por todo o período em que seus efeitos forem válidos. Os efeitos de ações também podem ser temporizados, podendo se tornar válidos após algum tempo da execução da ação e se manter válidos por tempo limitado. Esta característica também permite a execução em paralelo de ações, desde que todas as suas pré-condições estejam sendo atendidas.

Um problema de planejamento temporal usualmente tem como objetivo que dado conjunto de efeitos esteja válido em um intervalo de tempo, ou antes de um limite de tempo pré-definido. Devido à possibilidade de executar ações em paralelo pode-se gerar um novo problema, chamado de *problema de agendamento*, que consiste em otimizar a execução de um plano através do reposicionamento de ações no eixo temporal de acordo com suas limitações, cujo primeiro trabalho de maior significância na área de inteligência artificial pode ser visto em (Fox, 1994).

Uma representação para o controle temporal conhecida como visão orientada ao tempo (*time-oriented view*, em inglês) consiste em estender a representação de operadores lógicos com a inclusão de condições temporais, em que estados passam a ser computados a partir de funções parciais sobre as variáveis de estado em dado tempo². Neste modelo ações (também chamadas de *eventos*) ocorrem em momentos específicos e alteram valores de um conjunto de variáveis, que

²Nesta representação, uma função $pos(rI):t$ indica se um robô rI está em dada posição no tempo t .

mantém este valor pelo intervalo definido em sua condição de persistência (parte temporalmente definida das pós-condições deste evento).

Definição 2.5. (Planejamento temporal). O problema de planejamento temporal pode ser definido como uma versão especializada do problema de planejamento com recursos, da seguinte maneira:

V_n - Passa a ter um recurso correspondente ao tempo atual do plano, que funciona de maneira idêntica aos outros recursos numéricos.

O - As ações presentes em O podem receber intervalos de tempo em que podem ser executadas, tempo até que seus efeitos estejam ativos ou intervalos após os quais seus efeitos sofrem nova alteração.

I - Usualmente a este conjunto são inclusos os valores de controle temporais zerados, para representar uma situação de estabilidade ou espera, ou intervalos após os quais as condições iniciais não são mais garantidas e tem seus valores modificados.

G - Ao conjunto objetivo pode-se incluir restrições numéricas referente a limites de tempo, assim como requerer intervalos em que dada variável proposicional mantenha seu valor, representando situações de planejamento "contra o relógio", ou garantia de estabilidade de estado.

2.1.4 Incerteza

Extensão ao problema de planejamento clássico que visa permitir representação de sistemas não determinísticos, em que perde-se a certeza do efeito que a tomada de uma ação acarreta, que são então descritos através de probabilidades. Esta mesma classificação pode ser dada aos problemas em que o agente não possui todas as informações relevantes ao problema (Bonet e Geffner, 2000), ou nos quais existe a possibilidade de intervenção externa ao ambiente de planejamento, tal como em jogos de tabuleiro com mais de um jogador.

Devido à amplitude de variantes, problemas de planejamento com incerteza são subdivididos de acordo com a natureza da incerteza inserida no domínio. Por exemplo, um problema com observabilidade total em tempo de execução do plano e incerteza no estado atingido após a execução de ações pode ser chamado de *planejamento condicional* (Boutilier et al., 1999), onde o plano será estruturado de maneira similar a uma árvore, levando em consideração o estado observado após a execução de ações para a identificação do estado atual. Já em instâncias em que a conferência do estado atual via sensores em tempo de execução não é possível se faz necessária a geração de um plano que atinja o estado objetivo independente de variações nos resultados de ações tomadas durante a execução do plano.

Planos em ambientes com incerteza também podem ser classificados de acordo com suas garantias, sendo que um plano é denominado *fraco* caso exista ao menos uma instância de sua execução que atinja o estado objetivo, mas que ainda pode falhar em instâncias de acordo com os resultados das ações onde houver incerteza. Em contraponto, um plano é dito *forte* quando dá a garantia de que todas as ações não determinísticas estão sendo consideradas, e independente de seus resultados, o plano terminará sua execução no estado objetivo. Em ambientes com possibilidade de observação do estado do mundo existe também a possibilidade de gerar planos *fortes cíclicos*, que consistem em vincular a continuidade do plano com um efeito específico de suas ações, entrando em um ciclo de repetição caso contrário, até que o efeito desejado seja atingido.

Definição 2.6. (Planejamento com incerteza). No problema de planejamento com incerteza é utilizada uma definição diferente das abordadas até este ponto, onde os estados possíveis do mundo possuem representação explícita no problema³. Nesta representação o planejamento com incerteza é definido como a quintupla $\langle S, I, O, G, P \rangle$, onde:

³A esta representação é dado o nome de *modelo de estado-transição*, ou simplesmente *modelo de transição*

S - O conjunto finito S representa os estados possíveis do mundo, sendo $S = \{s_1, s_2, \dots, s_n\}$, com $s_i = \{l_1, l_2, \dots, l_m\}$ e para $l \in L$ gerado sobre o conjunto de variáveis V . Sua adoção sobre o conjunto V visa simplificar a representação de ações, facilitando as alterações necessárias aos outros itens da definição do problema.

I - Definido como $I \subseteq S$, com $|I| \geq 1$, o conjunto de possíveis estados iniciais define ao agente a incerteza sobre a situação inicial do problema, e possui análise funcionalmente idêntica ao estado inicial I do problema de planejamento clássico. Em situações em que este é o único ponto do problema em que existe incerteza o problema pode ser mapeado para planejamento clássico através da técnica apresentada em (Palacios e Geffner, 2009).

O - O conjunto de ações passa a ser representado como funções $o \subseteq S \times S$, e podem também incorporar aspectos não-determinísticos tais que sua execução passa o estado do mundo a um conjunto de estados, que devem ser distinguidos e considerados pelo agente após sua finalização através do uso de observações.

G - Definido como $G \subseteq S$, com $|G| \geq 1$, o conjunto de estados objetivo permite múltiplos objetivos (para um labirinto com mais de uma saída, por exemplo). Basta ao agente atingir ao menos um destes estados para que se possa considerar o problema como resolvido, permitindo ao agente a geração de planos mais variados.

P - O conjunto $P = \{C_1, C_2, \dots, C_n\}$ - é criado como uma partição do conjunto S , correspondendo a classes de estados indistinguíveis entre si⁴ tais que $\cup\{C_1, C_2, \dots, C_n\} = S$, $C_i \cap C_j = \emptyset$, para todo i, j tais que $1 \leq i \leq j \leq n$. A partir deste conjunto também podemos definir qual o nível de observabilidade do problema, sendo que se $P = \{\{s_1\}, \{s_2\}, \dots, \{s_n\}\}$ então afirmamos que P possui observabilidade total, se $P = \{s_1, s_2, \dots, s_n\}$ então afirmamos que P não possui observabilidade, e para os outros casos afirmamos que P possui observabilidade parcial do estado do mundo.

2.2 EXTENSÕES CONCEITUAIS

Nesta seção serão apresentadas extensões que não se referem apenas à estrutura do problema de planejamento, e sim às abordagens utilizadas para sua análise e resolução. As técnicas apresentadas a seguir não são representativas de todas as variantes existentes atualmente na literatura, e suas explicações serão focadas à apresentação de conceitos que possuam relação direta à técnica ou a possíveis extensões à versão tal como apresentada. A utilização destas técnicas permite aumentar a representatividade e capacidade de solução de problemas de planejamento sem que suas suposições sejam relaxadas, mas também podem ser aplicadas a versões de maior complexidade do problema.

2.2.1 Ordenação parcial por vínculos causais

Intuitivamente tem-se que um plano é definido como uma sequência linear de ações que ao serem executadas atingem um objetivo definido de antemão, mas na realidade esta linearidade é uma característica da execução do plano, não sendo inerente ao plano em si. Em (Sacerdoti, 1975) é levantada a possibilidade de *ordenação parcial de ações*, baseada na ideia de que apenas algumas ações requerem que outra ação tenha ocorrido anteriormente⁵. A esta abordagem foi dado o nome de *planejamento de menor comprometimento*⁶, abordada mais amplamente em (Weld, 1994).

⁴este conjunto também é chamado de *classes observacionais*

⁵para colocar um bloco a sobre um bloco b é necessário que não haja nenhum bloco sobre b , portanto é necessário remover qualquer bloco antes para esta ação estar disponível

⁶em tradução livre, do inglês *least commitment planning*

Neste novo método de montagem de planos é explicitado quando ocorre uma relação de ordenação obrigatória entre ações, para que o próprio planejador possa usar desta restrição, assim como redefinir quando esta é válida ou deve ser revista/removida. Em (McAllester e Rosenblitt, 1991) é definido o conceito de *vínculo causal*, que explicita esta relação entre ações do plano. Um vínculo causal é uma restrição de ordenação entre ações devido à necessidade de proteção de um conjunto de variáveis proposicionais. Ações que violem esta restrição recebem o nome de *ameaça*, e a restrição adicionada ao plano que resolve esta ameaça recebe o nome de *condição de segurança*.

O uso de ordenação parcial de planos controlada através de vínculos causais recebeu a sigla *POCL* (Barrett e Weld, 1994), e é uma técnica com uso altamente difundido em diversas subáreas do planejamento em inteligência artificial.

Definição 2.7. (Vínculo clausal). Um vínculo causal é definido como a tripla $\langle s, P, w \rangle$ onde:

P - O conjunto $P = \{p_1, p_2, \dots, p_n\}$ de variáveis proposicionais indica quais características do problema possuem a dependência, compartilhando de um mesmo vínculo causal.

w - Ação que possui todas as variáveis presentes em P em sua lista de pré-condições.

s - Ação que possui todas as variáveis do conjunto P em sua lista de efeitos adicionados.

O conjunto P está presente nas ações w e s , mas ambas podem possuir outras variáveis, que neste caso não possuem vínculo entre si.

A existência do vínculo causal $s \xrightarrow{P} w$ indica que a ação s precede a ação w , que nenhuma ação entre elas adiciona ou remove elementos do conjunto P , sendo utilizada para representar uma restrição ao plano sendo gerado.

Uma ameaça é definida como uma ação v , diferente de w e s , que inclui ou remove variáveis presentes em P , e a condição de segurança a este caso é a restrição de ordenação tal qual v preceda s ou ocorra apenas após w .

2.2.2 Planejamento por rede hierárquica de tarefas

O formalismo utilizado em HTN (cuja tradução mais difundida na literatura lusófona é *Rede Hierárquica de Tarefas*) foi criado para servir como ponte entre o planejamento em inteligência artificial e as técnicas desenvolvidas para pesquisa operacional, delineado originalmente em (Tate, 1977) e cuja fundamentação teórica pode ser vista de maneira mais aprofundada em (Erol et al., 1994a). Este formalismo tem o intuito de utilizar o conhecimento de um especialista do domínio sendo abordado para agilizar o processo de montagem de um plano viável através da identificação de etapas intermediárias, expandindo a capacidade de representação para múltiplos níveis de abstração.

Na representação modelo-transição usada pelo planejamento em busca no espaço de estados temos que cada estado do mundo é composto por um conjunto de literais positivos e a navegação nestes estados se dá através da tomada de ações que manipulam estes literais. De maneira similar, uma HTN utiliza estas estruturas como base, utilizando de *tarefas primitivas*⁷ em seu nível elementar para manipular o estado do mundo. O ponto em que uma HTN difere do formalismo STRIPS é pela presença de *redes de tarefas*, que são conjuntos de tarefas a serem executadas, assim como restrições e quais literais devem estar positivos após sua execução. Caso a rede de tarefas possua apenas tarefas primitivas, então recebe o nome de *rede de tarefas primitiva*, e sua solução é o resultado do agendamento de suas tarefas que satisfaça suas restrições.

O caso mais comum numa rede de tarefas é possuir tarefas não primitivas, que correspondem a referências a outras redes de tarefas, de maneira hierárquica até que seja

⁷uma tarefa primitiva corresponde funcionalmente a uma ação da representação STRIPS

decomposta em uma rede de tarefas primitivas⁸. Uma tarefa não primitiva não pode ser executada diretamente, mas permite a decomposição de tarefas altamente complexas em componentes mais simples, e também podem representar tarefas com mais de uma solução possível. A execução de uma tarefa não primitiva é feita através de *métodos*, que corresponde a vincular esta tarefa a uma nova rede de tarefas de um conjunto, e quando tiver todas suas tarefas executadas correspondem à execução da tarefa não primitiva original.

Uma vantagem desta abordagem é que o encadeamento de métodos a serem utilizados pode ser postergado, dando prioridade à geração de um plano válido em um maior nível de abstração (sem realizar a decomposição de tarefas não primitivas). Por exemplo, uma viagem pode ser considerada como uma tarefa não primitiva, cujos literais positivos esperados são a presença do agente na localidade destino, e cujo método utilizado para sua realização, caso não importem para o resto do problema, podem ser abordados em outro momento ou até mesmo por outro componente planejador.

Esta expansão garante a uma HTN uma maior riqueza estrutural se comparada à representação STRIPS. Enquanto qualquer conjunto de ações cujos efeitos culminem no estado objetivo é um plano válido para a representação STRIPS, uma HTN pode controlar ações disponíveis completamente, através de sua inclusão ou exclusão das possíveis expansões nas redes de tarefas não primitivas. Também é possível controlar estados intermediários e restrições no ordenamento sem a necessidade de incluir níveis adicionais de detalhamento ao domínio. Por exemplo, ao realizar a fabricação de peças em um ambiente de manufatura automatizada, existe em STRIPS a necessidade de incluir explicitamente condições de controle que garantam a ordenação das ações de manufatura enquanto no formalismo de uma HTN esta garantia está embutida nas redes de tarefas, permitindo uma descrição mais concisa e computacionalmente mais simples para o domínio.

2.2.3 Planejamento multiagente

Um sistema pode ser denominado *multiagente* caso exista mais de um componente inserido em um ambiente específico, com capacidades e limitações bem definidas e tentando atingir algum objetivo determinado através da manipulação do ambiente. O estudo de modelagem de problemas para agentes como apresentado foi adaptado da definição presente em (Jennings, 1999), vem sendo estudado fora do contexto de planejamento de maneira expansiva nas últimas décadas, e devido à sua extensão este trabalho se limitará à sua aplicabilidade para o problema de planejamento.

O que caracteriza um problema de planejamento como multiagente é a presença de agentes distintos entre si, em capacidades, restrições ou objetivos, que devem gerar um plano que atinja objetivos comuns ou privados, enquanto interagindo com os outros agentes de maneira cooperativa ou disruptiva. Devido a estas possibilidades o problema ganha maior capacidade de representação e abrangência, de acordo com os seguintes eixos de interação entre agentes:

- Grau de relação entre objetivos e recursos

Ausência de relação entre objetivos e recursos indica objetivos individuais díspares e não conflitantes, com recursos de uso dedicado a agentes. Em contraponto, uma relação forte indica dependência entre os agentes, podendo também indicar presença de ações necessariamente conjuntas. Quanto maior este grau mais complexo se torna o ambiente para planejamento, requerendo maior cooperação para atingir objetivos.

- Grau de cooperação entre agentes

⁸comportamento que dá a este tipo de planejamento outro de seus nomes, *planejamento decomposicional*

Agentes cooperativos podem focar nos objetivos globais e no sucesso coletivo, enquanto agentes egoístas ou adversariais podem reter informações e capacidades de maneira a inviabilizar objetivos comuns em favor a seus objetivos privados;

- Qualidade e frequência da comunicação

A passagem de informações frequente entre agentes cooperativos pode aumentar a eficiência do plano gerado, mas ambientes de planejamento hostis podem apresentar restrições de comunicação, por motivos de incapacidade técnica ou segurança. Agentes também podem restringir deliberadamente a passagem de informações, por motivos diversos.

Estas variações se tornam evidentes na etapa de *coordenação de planos*, onde cada agente apresenta informações sobre suas capacidades e objetivos (de acordo com seu grau de cooperação e privacidade) aos outros agentes, sendo o conjunto então analisado para evitar conflitos e preparar a separação de tarefas aos agentes mais capacitados a sua execução. Em planejamento multiagente o processo de geração de planos costuma ser abordado como uma coleção de subproblemas, que serão alocados aos agentes participantes e cuja execução atinge o maior número de objetivos (privados e comuns) possível. O processo de resolução pode ser dividido nas seguintes etapas, adaptado de (Durfee, 2001):

1. Alocação de objetivos a agentes;
2. Refinamento do objetivo global;
3. Coordenação pré-planejamento;
4. Montagem de planos individuais;
5. Coordenação pós-planejamento;
6. Execução do plano consolidado.

Devido à grande variedade de características conferida a problemas de planejamento multiagente cada etapa pode ser manipulada, estendida ou removida totalmente para melhor conformar ao ambiente sendo explorado. Nas seguintes subseções serão delineadas cada etapa e serão apresentadas algumas de suas variações possíveis.

2.2.3.1 Alocação de objetivos a agentes

A alocação de objetivos consiste em designar qual agente ficará responsável por atingir cada aspecto do objetivo comum. A aplicação de métodos centralizados pode tornar esta etapa trivial, pois um único agente distribui através de algum critério os objetivos entre os participantes, mas em um sistema menos controlado se faz necessário o uso de alguma técnica que permita maior autonomia e privacidade aos agentes.

Entre os métodos disponíveis estão a criação de coalizões entre agentes (Shehory e Kraus, 1998), o uso de leilões de objetivos (Wellman et al., 2001), ou simulações de mercado. Por exemplo, é apresentado em (Clearwater, 1996) como controles de custos e recursos podem ser transformados em ferramentas de controle e coordenação. Estas técnicas baseadas em teoria dos jogos se tornam especialmente importantes em ambientes orientados a valores, como minimização de custos ou controle estrito de recursos finitos. A alocação adequada possui especial relevância em robótica, e em (Gerkey e Matarić, 2004) foi realizada uma análise taxonômica para identificação formal das variantes possíveis aos agentes e tarefas.

2.2.3.2 Refinamento do objetivo global

Na segunda etapa será feita uma divisão dos objetivos e tarefas, de maneira que sejam definidos sub-problemas que possam ser distribuídos para e resolvidos por um agente. Os subproblemas resultantes desta divisão são então atribuídos aos agentes, que poderão utilizar quais técnicas lhes forem mais adequadas para obter uma solução para cada sub-problema.

2.2.3.3 Coordenação pré-planejamento

Após a definição inicial dos subproblemas, pode ser feita uma rodada de comunicação entre os agentes para coordenar a geração de planos e antecipar possíveis pontos de conflito e prioridades para uso de recursos limitados. Esta comunicação tem como intuito otimizar a geração individual de planos, minimizando replanejamento e reparo de planos.

2.2.3.4 Montagem de planos individuais dos agentes

Esta etapa consiste em cada agente resolver os problemas atribuídos a ele, idealmente acatando às regras sociais estabelecidas na etapa anterior. Neste ponto os agentes podem realizar troca de mensagens para agendamento de ações conjuntas, ou preparar seus planos totalmente isolados dos outros.

Nesta etapa cada agente pode utilizar-se de maneira independente aos outros das mais diversas técnicas de planejamento individual, desde planejamento não linear (Penberthy e Weld, 1992; Sacerdoti, 1975), passando por planejamento via HTN (Erol et al., 1994b) e planejadores temporais especializados em ambientes dinâmicos (Penberthy e Weld, 1994).

2.2.3.5 Coordenação pós-planejamento

Após ou durante a montagem de planos individuais, caso permitido aos agentes, existe a possibilidade de revisar planos para levar em consideração a possibilidade de cooperação e as restrições aplicáveis a recursos compartilhados ou de consumo finitos. Esta troca de informações deve levar em consideração as características dos planos e resultados esperados sendo compartilhados, pois mesmo que agentes homogêneos possam em teoria resolver o problema sozinhos, a paralelização de etapas pode ocasionar ganho de performance.

Uma das técnicas de coordenação pós-planejamento que pode ser executada de maneira centralizada é descrita em (Cox e Durfee, 2005), e consiste em analisar nos planos parciais de cada agente o fluxo causal, identificando pontos em que as restrições de ordenação entre ações desfazem uma condição necessária a ações futuras, que recebe o nome de *falha por ameaça causal*, e sua resultante, quando uma pré-condição não atingida é necessária para o prosseguimento do plano, denominada *falha de pré-condição aberta*. A técnica então realiza o reparo iterativo dos planos, tratando eles como planos parciais sendo executados em paralelo, buscando a montagem de um plano consistente e válido.

Em ambientes distribuídos, uma possibilidade é a comunicação entre agentes de planos/tarefas ou de resultados (Durfee, 2001). Quando comunicando tarefas, o agente pode ter identificado um objetivo que requer cooperação, e estar portanto requisitando a execução de ações conjuntas a outros agentes, ou ter determinado que seu plano não corresponde às restrições do problema, como limites de tempo e ações ou uso excessivo de recursos, e depender de separação e passagem de parcela de seus objetivos a outros agentes.

A comunicação de planos opera de maneira complementar, onde o agente identificou uma resposta válida a sua parcela do problema e oferece seu plano parcial para análise dos outros

agentes, com o intuito de permitir o agendamento de ações ou ofertar tempo ocioso para uso em cooperação em ações de outros agentes.

Já a comunicação de resultados entre agentes permite aos agentes que não queiram compartilhar suas capacidades e limitações a participação na coordenação de planos, partilhando apenas seu resultado⁹. Este tipo de comunicação costuma ser utilizado em ambientes de pouca relação entre objetivos e recursos, pois a recepção e análise dos resultados não é trivial, e quanto maior for a dependência entre ações ou recursos maior será a possibilidade de ocorrer interferência durante a execução do plano entre os agentes, o que pode comprometer o plano e costuma acarretar na necessidade de planos condicionais ou planos fortes, ou do monitoramento da execução de planos para reparo.

2.2.3.6 *Execução do plano consolidado*

A última etapa do problema é a execução do plano. Em ambientes estáticos ou determinísticos esta etapa serve para provar a validade do plano gerado, assim como adquirir dados de performance que corroborem o esperado. Caso o ambiente seja dinâmico existe a necessidade de monitorar e possivelmente reparar o plano sendo executado, de maneira centralizada ou distribuída.

O acompanhamento da execução, delineado em (Ambros-Ingerson e Steel, 1988) com o framework *IPEM*, visa identificar tipos de problemas com a execução do plano e aplicar a solução através de replanejamento e reagendamento de ações. Esta área de pesquisa possui muitas abordagens sendo consideradas, e recebe o nome de *Planejamento Contínuo* ou *Planejamento Contínuo Distribuído*.

⁹por exemplo, um agente pode apenas comunicar que suas tarefas serão concluídas no tempo t , atingindo o subconjunto N dos objetivos comuns

3 REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentadas as técnicas, aplicações e desenvolvimentos mais recentes de planejamento que servem como base à técnica apresentada, com foco em uma expansão conceitual do planejamento multiagente e em uma ferramenta que utiliza a estrutura de planejamento *HTN*.

3.1 SISTEMA *CASPER*

Nesta seção será apresentado o sistema *CASPER* (Knight et al., 2001), que serve como base conceitual para este trabalho. Para seu entendimento serão apresentados os conceitos em que o sistema se baseia, sendo eles a união de duas expansões ao problema de planejamento e uma abordagem alternativa à definição de objetivos, com foco nas características que o tornam relevante ao entendimento deste trabalho.

3.1.1 Planejamento em tempo real

Um subconjunto do problema de planejamento temporal, o planejamento em tempo real corresponde ao problema de definir um plano em ambientes em que a tomada de ações deve ser realizada com regularidade e em intervalos reduzidos, usualmente antes de o estado objetivo ter sido atingido. Este tipo de problema pode ser visto em jogos eletrônicos (Burns et al., 2013), veículos autônomos (Kim et al., 2013), em aplicações altamente especializadas em exploração controlada remotamente (Sherwood et al., 2014), entre outras.

Apresentado inicialmente em (Korf, 1990) com o algoritmo *LRTA**, que reconfigura o *A** clássico para que a tomada de ações seja antecipada. Em contraste ao *A**, em que o problema é resolvido integralmente antes que o agente comece a executar o plano, o algoritmo *LRTA** limita o tempo para a geração de planos, que passam a ser referentes a cada movimento¹. Esta limitação é feita de maneira independente ao grafo sendo gerado, e restringe a busca às informações mais pertinentes ao agente em sua posição atual e aos estados atingíveis de maneira mais próxima, assim evitando os saltos a nodos presentes no *A** e gerando planos aplicáveis a cada limite atingido.

A cada ação tomada o agente atualiza suas informações referentes às expectativas do problema, atualizando sua heurística e tornando a tomada de ações mais fluida em relação ao objetivo. Devido a esta limitação, pode-se observar situações em que as escolhas do agente não correspondem ao ótimo do problema, podendo inclusive ocasionar em repetição de estados previamente visitados em rápida sucessão (Hernández e Baier, 2012), fenômeno chamado de depressão heurística. Pesquisas atuais focaram na minimização deste problema, identificando depressões heurísticas e evitando ou mitigando o efeito que elas causam à geração de planos (Shimbo e Ishida, 2003; Sharon et al., 2013).

3.1.2 Planejamento em mundo aberto

Expansão do problema de planejamento com incerteza, o planejamento em mundo aberto remove a suposição de mundo fechado (em que todas as informações relevantes ao problema são sabidas de antemão), o que implica em uma quantidade possivelmente infinita de estados, e faz

¹este limite pode ser chamado de *horizonte* do planejamento

com que o agente precise tomar ações tanto com o intuito de atingir o estado objetivo quanto para adquirir conhecimento sobre o mundo. Esta alteração faz com que o estado atual possua valores e literais ainda não descobertos, que devem ser tratados de maneira diferenciada durante a geração de planos, tendo seus valores definidos através de ações sensoriais.

Como o estado do mundo conhecido é parcial o planejador pode utilizar a estrutura de *objetivos suaves*². No contexto de planejamento em mundo aberto, a inclusão deste tipo de objetivo pode ser realizada para incentivar o agente a realizar observações do mundo ou para consolidação de informações (tal como implementado no planejador disponível em (Giunchiglia e Maratea, 2007)).

Uma aplicação muito relevante para planejamento em mundo aberto é a robótica, em que o robô assume o papel de planejador e agente, e deve atingir algum objetivo a partir de um mínimo de informações sobre o estado do mundo a serem complementados durante o processo através de ações de observação do mundo. Desta área destaca-se pesquisas em contextos de robôs utilizados para busca e resgate³ (Talamadupula et al., 2010; Tomic et al., 2012; Kruijff et al., 2014), visando métodos que permitam atingir autonomia plena em ambientes diversos, mesmo com alto grau de interferência comprometendo comunicação estável entre o robô e humanos.

3.1.3 Planejamento contínuo

Cenário que surge da junção das duas expansões apresentadas, unindo a necessidade de aquisição e consolidação de conhecimento através de ações sensoriais com a execução de planos parciais em horizonte reduzido, para minimizar acúmulo de erro e permitir a retroalimentação de heurísticas e expectativas a medida que fatos são verificados.

O sistema CASPER incorpora todas as técnicas descritas até agora em conjunto, utilizando controle de recursos para monitorar limites computacionais (tal como uso de RAM), identificando e corrigindo sobrecarga através de reagendamento temporal de ações com utilização do recurso sobrecarregado, gerando planos parciais com um horizonte fixo e reduzido, e acompanhando via sensores o andamento da execução do plano atual para executar reparos e expandir seu escopo de maneira incremental, revisando objetivos e estados atingíveis a cada momento oportuno.

3.1.4 Planejamento em múltiplos níveis

O sistema CASPER foi desenvolvido para uso em exploradores planetários semi-autônomos. Robôs de exploração anteriores a este sistema dependiam de comunicação direta para recebimento de ordens, o que para distâncias maiores limitava as ações que poderiam ser executadas, por não possuírem autonomia suficiente para reagir a imprevistos e dependerem de intervenção direta através de replanejamento por controladores remotos. No sistema CASPER a responsabilidade do controlador se restringe à definição dos objetivos a serem buscados, passando ao agente a criação e acompanhamento do plano. Este aumento na expectativa e capacidade de raciocínio conferida ao agente permite a realização das tarefas de planejamento e reagendamento de ações localmente, poupando o tempo gasto com comunicação externa e tornando viáveis tarefas de maior nível de complexidade, e portanto maior capacidade de gerar e analisar dados relevantes às pesquisas sendo realizadas.

Este processo de reagendamento ocorre através da técnica de *conserto iterativo* de planos, descrita originalmente para o sistema ASPEN (Rabideau et al., 1999), que consiste

²um objetivo suave não compõe o estado objetivo do problema, mas é usado para melhorar o entendimento do estado do mundo

³esta área normalmente é referenciada pela sigla SAR, em inglês

em identificar anomalias em planos pré-formados e realizar o reagendamento dos aspectos diretamente relacionados a esta falha de maneira iterativa, até que seja criado um plano válido que atinja aos objetivos originais. Graças a este foco o tempo de processamento é reduzido consideravelmente, o que minimiza o tempo ocioso do robô e permite o uso de planejamento contínuo para identificar através de sensores o momento em que a retomada de ações pode ser realizada.

Esta maior autonomia possibilita ao sistema CASPER o recebimento de objetivos de maior nível de abstração, e a geração do plano inicial levando em consideração os objetivos passados e suas limitações operacionais. Este processo pode ser considerado como uma versão simplificada de uma HTN, pois apesar da divisão em níveis a estrutura do problema passada ao agente não possui tarefas abstratas, se limitando à definição de um objetivo em análise abstrata e fazendo todo o processo de planejamento remotamente sem utilização da estrutura hierárquica.

Desde sua apresentação em 2001, o sistema CASPER tem figurado como uma aplicação prática importante de um problema de mundo real mapeado ao planejamento em inteligência artificial, tem aparecido em comparativos de técnicas (Ingrand e Ghallab, 2017) e expandido para outras variantes de planejamento não clássico (Chanel et al., 2019). Para este trabalho serão utilizadas seu conceito de divisão de responsabilidade e sua aplicação de planejamento em horizonte limitado, adaptados ao formalismo HTN.

3.2 COMPETIÇÕES ICAPS

A área de pesquisa de planejamento em inteligência artificial teve forte ressurgência na conferência então chamada AIPS⁴ de 1998, através da introdução da linguagem PDDL (Ghallab et al., 1998) e da competição que a utilizaria, a primeira competição internacional de planejamento (McDermott, 2000). Desde então estas competições servem como palco para a apresentação de novos avanços em planejadores e técnicas, assim como de novas extensões e variantes ao problema de planejamento. Todas as variantes de planejamento clássico apresentadas no capítulo anterior deste trabalho já tiveram competições dedicadas, e diversas das extensões conceituais já foram utilizadas em resolvedores participantes.

O amadurecimento da área de planejamento permitiu que nos últimos anos as competições se tornassem mais complexas, abordando planejamento distribuído em cenários multiagente (Komenda et al., 2016) e mais recentemente analisando a performance dentro de um seletor de planejadores automatizado, no *Sparkle Planning Challenge 2019*. Este trabalho é em parte inspirado pela competição *Unsolvability IPC*, cujos resultados foram apresentados em 2016 e que analisou planejadores em quão eficientes eles eram em identificar domínios sem solução. Nesta competição o foco foi em separar domínios entre com e sem solução, não importando qual a solução existente ou o motivo que o torna insolúvel.

A identificação de motivos possui área própria denominada *inteligência artificial explicável*, de sigla XAI (Gunning, 2019). O foco desta área é mitigar uma tendência de processos automatizados de aprender e utilizar de técnicas que os tornem caixas-pretas, onde as escolhas feitas podem não ser imediatamente entendíveis por um humano supervisionando o processo (ou observando seus efeitos). Esta área já possui pesquisas desde os anos 1970, através do sistema *MYCIN*, cujos detalhes podem ser vistos em (Shortliffe, 2012). Atualmente este conceito tem

⁴que após junção com outra conferência (*ECP, European Conference on Planning*) em 2003 recebeu o nome pelo qual é conhecida hoje, *ICAPS*

gerado maiores discussões no que se refere às técnicas naturalmente opacas de aprendizado de máquina, tal como *Deep Learning* e algoritmos genéticos⁵.

Mais detalhes sobre esta área podem ser vistos em (Montavon et al., 2019), sendo que neste trabalho será abordada a explicabilidade no que se refere ao planejamento (Fox et al., 2017). Entre as diversas maneiras de tornar planos explicáveis está o planejador HEART apresentado a seguir, e que será usado extensivamente neste trabalho.

3.3 PLANEJADOR HEART

Nesta seção será apresentado o planejador HEART, que é utilizado como embasamento técnico para a técnica apresentada neste trabalho. Inicialmente será apresentada a variante ao problema de planejamento que este planejador resolve, a notação adotada para as descrições de domínios e as definições que diferem do padrão atribuído a problemas de planejamento. Devido à extensão de sua utilização neste trabalho será dado maior foco e contextualização de suas características como ferramenta, apresentando também seu funcionamento em um domínio exemplo quando adequado ao entendimento das definições apresentadas.

3.3.1 Planejamento explicável

O maior foco das pesquisas de planejamento atuais é o aumento da robustez, expressividade e performance do planejador como ferramenta (Ghallab et al., 2014), gerando resolvedores altamente especializados a tipos limitados de domínio. Este aumento de especificidade de técnicas é fato notório da área e a busca por motivos e soluções vem sendo explorada amplamente (Ghallab et al., 2016; Winikoff et al., 2018; Sreedharan et al., 2019). Porém, esta especialização crescente tende a alienar pessoal não treinado, minimizando aplicabilidade como ferramenta e levando a maior nível de dependência de uso das soluções como caixa-preta, degradando a capacidade de explicar o que aconteceu quando o plano não conforma com as expectativas das pessoas que o observam. Neste contexto o planejador HEART foi proposto, com o intuito de tornar o conteúdo de um plano explicável, mesmo durante sua elaboração.

O conceito de planejamento explicável tem crescente número de publicações recentemente, desde a extração de explicação para problemas sem solução (Göbelbecker et al., 2010) até a criação de planos abstratos para análise humana (Seegebarth et al., 2012; Fox et al., 2017). O planejador HEART foi criado com estes mesmos intuítos, visando uma abordagem que permita o reconhecimento de intenção do planejador no domínio através da análise do plano gerado, estendendo ao ferramental os conceitos da teoria da mente da psicologia⁶. Esta nova capacidade de inferência visa aumentar a capacidade de o planejador interpretar dados aproximados através do comportamento de outros agentes ou através de aprendizado de fluentes durante o planejamento.

Sabendo que planejamento sob representações complexas de HTN possui complexidade EXPSPACE-difícil (Erol et al., 1994a) o planejador HEART tem como foco a geração de planos abstratos válidos, utilizando da representatividade de uma HTN para manter a capacidade de um analista humano entender o plano sem comprometer a corretude do plano gerado. Juntando à estas características a concisão do plano resultante seu uso passa a ser possível em problemas com dados aproximados ou como ferramenta de inferência em extração de planos através de ações em problemas de planejamento invertido (Ramirez e Geffner, 2009; Riabov e Udrea, 2016).

⁵Um algoritmo genético pode ser classificado como caixa-branca, mas mesmo assim o impacto de características em cada evolução nem sempre possuem motivação e podem apresentar tendências devido à implementação, ao invés de apenas sobre o conjunto de dados sendo analisado

⁶Em sumário consiste em inferir intenção de outrem através da aplicação de vínculo empático, exemplificada pela frase "O que eu faria se fosse esta pessoa?"

3.3.2 Princípios de funcionamento

O planejador HEART pode ser classificado como um planejador decomposicional em tempo real baseado em vínculos causais parcialmente ordenados. Ele opera integrando algoritmos de PSP (Bercher et al., 2014) com POCL através da inclusão de novos tipos de falhas relativas ao POCL ao processo de decomposição. Sua estrutura é uma versão aprimorada da utilizada no planejador HiPOP (Bechon et al., 2014), reduzindo o número de ajustes necessários para lidar com ações compostas.

Para melhor entendimento do funcionamento do planejador HEART serão apresentadas nesta seção as definições utilizadas por ele, como guia dos termos que serão usados para os experimentos, com exemplos simples onde adequado.

Definição 3.1. (Domínio). Um domínio é a tripla $\mathcal{D} = \langle E_{\mathcal{D}}, R, A_{\mathcal{D}} \rangle$, onde:

- $E_{\mathcal{D}}$ é o conjunto de entidades do domínio.
- R é o conjunto de relações sobre $E_{\mathcal{D}}^n$. Estas relações são similares a predicados de aridade n usados em lógica de primeira ordem.
- $A_{\mathcal{D}}$ é o conjunto de operadores composto de ações abstratas.

Definição 3.2. (Fluente). Um fluente f é a declaração parametrizada $r(arg_1, arg_2, \dots, arg_n)$ onde:

- $r \in R$ é a relação/função que representa uma característica do mundo.
- $arg_{i \in [1, n]} \in E_{\mathcal{D}}$ são os argumentos, possivelmente quantificados.
- $n = |r|$ é a aridade de r .

Fluentes aceitam sinalização, onde um fluente negativo é representado $\neg f$ e corresponde ao complemento lógico de f , sendo que quantificadores também estão sujeitos ao efeito desta sinalização. Não é utilizada a suposição de mundo fechado, portanto fluentes são satisfeitos apenas quando apresentados. Para indicar que um objeto não está sendo segurado é necessária sua descrição como $\neg segurado(objeto)$. Relação entre entidades também pode ser representada por fluentes de aridade superior, tal como $dentro(objeto, caixa)$ indica que a caixa contém o objeto.

Definição 3.3. (Plano parcial/método). Um plano parcial é o grafo acíclico direcionado $\pi = (S, L)$, com:

- S corresponde aos passos do plano representados por vértices, e sempre contém os passos inicial I_{π} e o passo objetivo G_{π} . Um passo é uma ação pertencente ao plano, abstrata ou primitiva.
- L é o conjunto de vínculos causais como arestas. Pela notação tem-se que $l = a_a \xrightarrow{c} a_s$ é o vínculo entre a ação antecessora e a ação alvo, causado pelo conjunto de literais c . Caso $c = \emptyset$ então o vínculo corresponde a uma restrição de ordenação de ações.

Na estrutura utilizada, uma restrição de ordenação é definida como a cobertura transitiva dos vínculos causais sobre o conjunto de passos. A notação para estas restrições é $a_a > a_s$, com o passo a_a sendo *anterior* a seu *sucessor* a_s . Restrições de ordenação não podem formar ciclos, portanto $a_a \neq a_s \wedge a_s \not> a_a$. Adicionalmente todo plano ocorre entre os passos inicial e final, portanto $I_{\pi} > G_{\pi} \wedge \nexists a_x \in S_{\pi} : a_x > I_{\pi} \vee G_{\pi} > a_x$. Caso exista a necessidade de forçar ordenação basta adicionar o vínculo correspondente sem especificar literais causadores.

Uma noção central a planejadores é o conceito de *operador*. Um operador instanciado é chamado de *ação*. Entretanto, na estrutura apresentada uma ação pode ser apenas parcialmente instanciada, e o termo é utilizado tanto para ações primitivas quanto abstratas.

Definição 3.4. (Ação). Uma ação é a tupla parametrizada $a(args) = \langle nome, pre, eff, met \rangle$ onde:

- *nome* é o nome da ação.
- *pre* e *eff* são os conjuntos de fluentes que representam as pré-condições e os efeitos da ação, respectivamente.
- *met* é o conjunto de métodos (planos parcialmente ordenados) que decompõem esta ação. Métodos deste conjunto e suas ações componentes não podem conter a ação *nome*.

Como exemplo, a ação *segurar(objeto)* pode ter como pré-condição apenas $\neg segurado(objeto)$ como garantia de que o objeto já não está sendo segurado, como efeito *segurado(objeto)* e nenhum método listado, caso corresponda a uma ação primitiva do agente.

Uma ação é dita *composta* ou *abstrata* caso possua métodos para ser decomposta, enquanto é dita *atômica* ou *primitiva* caso contrário. Esta representação diverge de uma HTN usual em que apenas ações primitivas possuem pré-condições e efeitos, pois sua inclusão nas ações abstratas viabiliza a geração de planos de maior nível de abstração.

Ações abstratas podem ter suas causas e efeitos informados junto à entrada do domínio. Caso omitidos é realizada a inferência através da unificação das pré-condições e efeitos utilizando o mesmo algoritmo de resolução de objetivos parciais do algoritmo POCL padrão. Como o objetivo é poder garantir validade de planos abstratos, se faz necessária a garantia de que a entrada corresponde a um domínio resolvível, caso esta validação falhe então a compilação do domínio é interrompida.

A inferência das pré-condições e efeitos de uma ação a é realizada a partir das seguintes fórmulas: $pre(a) = \bigcup_{a_S \in L^+(a)} causa(a_S)$ para pré-condições, e $eff(a) = \bigcup_{a_S \in L^-(a)} causa(a_S)$ para efeitos. A coerência destes valores é conferida através de uma instância $\mathcal{P}_a = \langle \mathcal{D}, C_{\mathcal{P}}, a \rangle$ do problema POCL para cada ação abstrata.

Definição 3.5. (Problema). Um problema de planejamento é então definido como a tupla $\mathcal{P} = \langle \mathcal{D}, C_{\mathcal{P}}, a_0 \rangle$, onde:

- \mathcal{D} é o domínio de planejamento.
- $C_{\mathcal{P}}$ e o conjunto de constantes do problema, disjunto às constantes do domínio (relações de aridade 0 do conjunto R).
- a_0 é o operador raiz do problema

A inicialização do operador raiz é feita através da ação de nome vazio $a_0 = \langle '', s_0, s^*, \{\pi_{lv(a_0)}\} \rangle$, com s_0 o estado inicial e s^* o objetivo especificado. O método $\pi_{lv(a_0)}$ é o plano parcial em que os estados inicial e final estão vinculados através de a_0 , com $\pi_{lv(a_0)} = (\{I, G\}, \{I \xrightarrow{s_0} a_0 \xrightarrow{s^*} G\})$, onde $I = \langle 'ini', \emptyset, s_0, \emptyset \rangle$ e $G = \langle 'obj', s^*, \emptyset, \emptyset \rangle$.

O método de solução utilizado pelo planejador HEART é o mesmo que um POCL clássico, em que planos parciais são refinados em uma solução através da remoção recursiva de todas as suas falhas.

Definição 3.6. (Falha). Uma falha é composta de um literal f e de um passo causador a_n , também chamado de *requerente*. Uma falha pertence a um de dois tipos:

- *Objetivos parciais*, condições em aberto que ainda não estão suportadas pelo passo provedor a_n . Recebe a notação $\downarrow_f a_n$.
- *Ameaças*, causadas por passos que quebrem um vínculo causal com seus efeitos. Chamados de *quebradores* do vínculo ameaçado. Um passo a_b ameaça o vínculo $l_t = a_p \xrightarrow{f} a_n$ se, e somente se, $\neg f \in \text{eff}(a_b) \wedge a_b \not\prec a_p \wedge a_n \not\prec a_b$. Por extenso, o quebrador pode cancelar um efeito provido pelo passo a_p antes que o requerente a_n possa utilizá-lo. Ameaças são notadas como $a_b \otimes l_t$.

Um plano é considerado válido quando todas as suas falhas forem corrigidas, e em POCL isto ocorre através da busca por seus *resolvedores*.

Definição 3.7. (Resolvedor). Um resolvedor clássico é um vínculo clausal que é adicionado para resolver uma falha, de acordo com seu tipo.

- Para objetivos parciais resolvedores são o conjunto de possíveis vínculos causais que possuem o fluente f entre suas causas, tendo o passo a_n como objetivo e o provedor a_s como sua origem.
- Para ameaças existem dois resolvedores: *rebaixamento* ($a_b \succ a_p$) e *promoção* ($a_n \succ a_b$) do quebrador relativo ao vínculo ameaçado. Vínculos adicionados sem originador podem ser referenciados como vínculos *protetores*.

Por exemplo, um objetivo de ter um objeto numa caixa ($\text{dentro}(\text{objeto}, \text{caixa})$) pode ser atingido com a inclusão de resolvedores que possuam uma ação $\text{colocar}(\text{objeto}, \text{caixa})$ como origem de um vínculo causal, carregando o fluente objeto como sua única causa.

A aplicação de resolvedores não necessariamente indica progresso. Tal inclusão pode causar consequências que vão requerer replanejamento de alguns aspectos para evitar violação do funcionamento do POCL.

Definição 3.8. (Efeitos colaterais). Uma falha causada pela aplicação de um resolvedor é dita uma *falha relacionada*. Tais falhas precisam ser consideradas para tratamento pelo algoritmo POCL, através de sua inclusão na *agenda*⁷.

- *Objetivos parciais relacionados* são as condições em aberto geradas pela inclusão de novos passos.
- *Ameaças relacionadas* são vínculos causais ameaçados pela inserção de um novo passo, ou remoção de um vínculo protetor.

A aplicação de resolvedores também pode tornar falhas irrelevantes. Este sempre é o caso em resolução de uma falha, mas também pode ocorrer com outras falhas. Estas *falhas invalidadas* são removidas da agenda na ocasião de sua detecção:

- *Objetivos parciais invalidados* são objetivos satisfeitos pelo novo vínculo incluso, ou cujo requerente tenha sido removido.

⁷este termo será mantido com o nome em inglês, pois sua tradução mais correta seria *plano*, o que causaria confusão.

- *Ameaças invalidadas* ocorrem quando o quebrador não mais ameaça um vínculo, através da inclusão de um protetor da ordem necessária ou caso algum dos passos do vínculo seja removido.

O seguinte algoritmo corresponde a uma versão de POCL, adaptada de (Ghallab et al., 2004), e será utilizado para explicar o processo de resolução através de refinamento.

Algoritmo 1 POCL:

Entrada a agenda, \mathcal{P} problema
Saída [Sucesso, Insucesso]

```

1: if Agenda  $a = \emptyset$  then
2:   return Sucesso // Agenda populada faz parte da entrada, caso contrário o plano está pronto
3: end if
4: Falha  $f \leftarrow escolhe(a)$  // Escolha heurística de falha
5: Resolvedor  $R \leftarrow resolve(f, \mathcal{P})$ 
6: for all  $r \in R$  do
7:   aplica( $r, \pi$ ) // Aplica o resolvedor ao plano parcial atual
8:   Agenda  $a' \leftarrow atualiza(a)$ 
9:   if POCL( $a', \mathcal{P}$ ) = Sucesso then
10:    return Sucesso
11:  end if
12:  reverte( $r, \pi$ ) // Resolução sem sucesso, reverte aplicação do resolvedor
13: end for
14:  $a \leftarrow a \cup \{f\}$  // Falha não resolvida
15: return Insucesso // Volta para a última escolha de  $r$ 

```

O processo de refinamento do plano ocorre da seguinte maneira. Ao identificar uma falha através da função *resolve* na linha 5, a busca por possíveis resolvedores é feita em duas etapas, a seleção de candidatos e a conferência de validade de cada um deles. O processo completo para um objetivo parcial ocorre na seguinte ordem:

1. Escolha de resolvedores candidatos:
É buscado entre as definições da ação abstrata referente ao objetivo parcial sendo refinado por resolvedores que possam atingir algum objetivo do problema.
2. Unificação de variáveis:
As variáveis das ações candidatas resolvedor são unificadas ao restante das já presentes no plano parcial. Este passo ocorre para garantir a compatibilidade dos resolvedor candidatos, e neste instante são instanciados todos os métodos, inclusive os compostos.
3. Seleção de resolvedor:
Após a separação das ações candidatas compatíveis é feita a escolha não determinística (que pode ser guiada por heurística) de um resolvedor, a ser aplicado no plano parcial atual.
4. Aplicação de resolvedor:
Na linha 7 é feita a aplicação do resolvedor ao plano, já com suas variáveis adequadas à falha sendo refinada.
5. Busca de efeitos colaterais:
Após a aplicação do resolvedor é necessário conferir efeitos colaterais, atualizando a agenda e chamando recursivamente o processo de refinamento.

Caso ocorra algum problema durante este processo então a linha 12 reverte as alterações e tenta outro resolvidor, e caso nenhum resolvidor consiga expandir o objetivo ou resolver a falha então ocorre o retrocesso do planejamento para escolhas anteriores de falhas/resolvedores.

3.3.3 Definições adicionais ao HEART

O grande diferencial entre o POCL padrão e a versão implementada pelo HEART é a integração de estrutura HTN a este processo, o que o torna capaz de gerar planos intermediários ainda com ações abstratas. As definições apresentadas até agora são características ao POCL, enquanto as que serão apresentadas a seguir são para viabilizar a implementação do planejador HEART como sendo apresentado.

Definição 3.9. (Transposição). Para poder realizar a decomposição de ações abstratas corretamente é necessário definir a relação dos vínculos causais entre ações e planos parciais. No contexto do planejamento usado pelo HEART entende-se como *transposição* a operação de redirecionar os vínculos causais de uma ação a' com as de um passo já definido a em um plano π , definida formalmente como:

$$a \triangleright_{\pi}^{-} a' = \left\{ \phi^{-}(l) \xrightarrow{\text{causa}(l)} a' : l \in L_{\pi}^{-}(a) \right\}$$

Esta mesma operação é aplicável com $a' \xrightarrow{\text{causa}(l)} \phi^{+}(l)$ e L^{+} para $a \triangleright^{+} a'$. Isto supõe que as respectivas pré-condições e efeitos de a e a' são equivalentes. Quando não sinalizada a transposição pode ser generalizada para $a \triangleright a' = a \triangleright^{-} a' \cup a \triangleright^{+} a'$.

Ao final de uma transposição não sinalizada todos os vínculos causais de a' passam a estar referenciados no passo a já presente no plano, incorporando a' ao plano e atualizando a agenda com os dados para o novo passo do planejamento.

Definição 3.10. (Ações próprias). Uma ação é dita *própria* quando está contida em ("é propriedade de") uma entidade. Para uma ação a isto é notado como $A_a = A_a^{lv(a)}$, e pode ser aplicado para diversos conceitos:

- Para um *domínio* ou *problema*, $A_{\mathcal{D}} = A_{\mathcal{D}}$.
- Para um *plano*, $A_{\pi}^0 = S_{\pi}$.
- Para uma *ação*, $A_a^0 = \bigcup_{m \in met(a)} S_m$. Recursivamente, $A_a^n = \bigcup_{b \in A_a^0} A_b^{n-1}$. Para ações atômicas, $A_a = \emptyset$.

Definição 3.11. (Nível de abstração). Esta é a medida do maior grau de abstração (separação de ações atômicas) que uma entidade pode expressar, definida recursivamente:

$$lv(x) = \left(\max_{a \in A_x} (lv(a)) + 1 \right) \llbracket A_x \neq \emptyset \rrbracket$$

Pela definição nota-se que o nível de abstração será sempre 0 para ações primitivas, e para todas as outras ações será 1 maior que o máximo presente em suas ações próprias. Esta medida é utilizada para a identificação de ações próprias e também para a criação de um novo tipo de falha a ser analisada pelo POCL, denominada *falha de decomposição*.

Em (Gerevini et al., 2008) é apresentada uma das maneiras mais simples de integrar abstração em um planejador regular, onde ações hierárquicas são tratadas em separado do processo de escolha de ações de um planejador regular. A versão utilizada pelo planejador HEART é

baseada no planejador HiPOP (Bechon et al., 2014), ajustada para manter a expressividade e utilizar o mecanismo de seleção de falhas do POCL. A versão utilizada no HiPOP faz o aterramento de ações hierárquicas (decomposição em ações atômicas) antes do início do planejamento, enquanto o HEART realiza esta decomposição durante a resolução do problema.

Definição 3.12. (Falha de decomposição). Uma falha de decomposição é caracterizada pela presença de um passo não atômico no plano parcial. Este passo é tido como o requerente a_n para a falha, e esta decomposição é notada com $a_n \oplus$.

- *Resolveror*: Uma falha de decomposição é resolvida por um *resolveror decompositor*. Este resolveror substitui o requerente por um de seus métodos instanciados em $m \in \text{met}(a_n)$ no plano π . Para tanto é utilizada uma transposição tal que: $a_n \oplus_{\pi}^m = \langle S_m \cup (S_{\pi} \setminus \{a\}), a_n \triangleright^{-} I_m \cup a_n \triangleright^{+} G_m \cup (L_{\pi} \setminus L_{pi}(a_n)) \rangle$.
- *Efeito colateral*: Uma falha de decomposição pode ser criada pela inclusão de uma ação composta por qualquer outro resolveror, e invalidada por sua remoção:

$$\bigcup_{a_m \in S_m}^{f \in \text{pre}(a_m)} \pi' \downarrow_f a_m \bigcup_{a_b \in S_{\pi'}}^{l \in L_{\pi'}} a_b \otimes l \bigcup_{a_c \in S_m}^{lv(a_c) \neq 0} a_c \oplus$$

Durante o processo de refinamento, ao escolher uma ação composta para resolver uma falha qualquer faz com que seja inclusa como efeito colateral a falha de decomposição da ação, que deverá ser resolvida através de um resolveror decompositor. Para decompor uma ação composta em um plano parcial é necessário que todos seus vínculos sejam transpostos para os passos inicial e final do método escolhido, e a ação composta e seus vínculos são então removidos do plano sendo criado.

A maior diferença entre a implementação do HiPOP e do HEART está nas funções de escolha de falha e no tratamento de seu resultado, sendo que o planejador HiPOP gera apenas um plano ao final do processo, enquanto o HEART gera um plano abstrato a cada ciclo de funcionamento. Para atingir este objetivo o planejador HiPOP foca em falhas de decomposição, o que apesar de apresentado como mais eficiente (Bechon et al., 2014) acaba por descartar possibilidades de geração de planos abstratos durante o processo.

Em contraponto, o foco do planejador HEART é a possibilidade de gerar planos completos que ainda possuam ações compostas, para melhor entendimento por operadores humanos ou simples expediência na apresentação de resultados válidos. Esta geração de planos abstratos é possível com a inclusão do conceito de *ciclos* ao processo de planejamento.

Definição 3.13. (Ciclo). Um ciclo é definido como a fase de planejamento da tripla $c = \langle lv(c), agenda, \pi_{lv(c)} \rangle$ onde:

- $lv(c)$ é o maior nível de abstração permitido para a seleção de falhas.
- $agenda$ é o conjunto de falhas pendentes no plano parcial sendo analisado.
- $\pi_{lv(c)}$ é o plano parcial onde a seleção de falhas ocorre.

Durante cada ciclo a escolha de falhas de decomposição são atrasadas até que não hajam outros tipos de falhas a resolver. Havendo apenas falhas de decomposição na agenda o plano corrente é então armazenado, sendo ele um possível plano abstrato $\pi_{lv(c)}$. Todas as falhas de

decomposição são então resolvidas para o início do novo ciclo, onde $lv(c') = lv(c) - 1$. Desta maneira, cada ciclo gera um plano abstrato com maior detalhamento que o anterior.

Planos abstratos armazenados permitem ao planejador realizar uma forma aproximada de *execução a qualquer momento* (Zilberstein e Russell, 1995), pois a qualquer momento o planejador é capaz de apresentar um plano com suporte total das informações correntes, apesar de não necessariamente com todos seus fluentes instanciados. Caso o primeiro ciclo não esteja ainda concluído será apresentado $\pi_{lv(a_0)}$. Estes planos intermediários não são solução do problema até que seja atingido o ciclo c_0 , com nível de abstração $lv(c_0) = 0$. Este trabalho visa adaptar esta aproximação para uma aplicação multiagente, a partir da técnica a ser apresentada no próximo capítulo.

4 PLANEJAMENTO SEMI-AUTÔNOMO

Neste capítulo é apresentado um novo conceito, denominada *planejamento semi-autônomo*. Este novo conceito apresenta uma abordagem alternativa para planejamento baseada na separação de responsabilidades utilizada pelo sistema CASPER e utilizando como base as características do planejamento em HTN e multiagente. Inicialmente serão apresentados o conceito e seu embasamento teórico. O capítulo se encerra com a descrição de seu funcionamento tal como implementado neste trabalho.

4.1 MOTIVAÇÃO PARA O NOVO CONCEITO

No contexto de planejamento multiagente podemos classificar sistemas de acordo com a responsabilidade de geração de planos por seus agentes de duas maneiras. Podemos definir um sistema como autônomo caso seus agentes sejam responsáveis pela elaboração e execução do plano, controlando cumprimento de objetivos e observando a manutenção das restrições do problema. Em contraponto, um sistema cujo planejamento é realizado de maneira centralizada pode ser definido como possuindo agentes não autônomos pois apesar de a execução das ações ser distribuída, a responsabilidade de geração, acompanhamento e resolução de conflitos são atribuídas a um agente centralizador.

Com base nesta classificação pode-se definir como planejamento semi-autônomo o meio termo entre os dois, onde parte da responsabilidade de geração e acompanhamento de subplanos mais restritos é repassada ao agente executor das ações, enquanto cabe a um agente *centralizador* ou *supervisor* a coordenação de planos em maiores níveis abstração para definir passos da resolução do problema. Este conceito não possui exploração ampla dentro da comunidade científica, sendo abordado apenas tangencialmente em análises de coordenação multiagente totalmente distribuída ou por inteligência de enxame de algoritmos evolutivos (Godoy et al., 2016; Contreras-Cruz et al., 2017).

Este trabalho tem por objetivo apresentar uma nova maneira de realizar esta divisão de responsabilidades através de uma rede hierárquica que permita a automatização da segmentação em tempo de resolução do problema, acomodando agentes de capacidades variáveis sem que seja necessária intervenção manual ao resolvidor.

4.2 EMBASAMENTO TEÓRICO

No sistema CASPER é utilizada uma divisão de responsabilidades na qual o agente remoto recebe de um componente externo quais os objetivos a serem atingidos pelo plano sendo iniciado. Mapeando este processo para as etapas delineadas ao planejamento multiagente temos os seguintes passos:

1. A alocação de objetivos é realizada pelo centro de controle, atribuindo todos os objetivos definidos ao resolvidor remoto (por haver apenas um agente disponível);
2. De maneira similar não há necessidade da divisão em subproblemas, pois todos os objetivos e tarefas serão distribuídos ao único agente;
3. Neste momento o objetivo definido pelo centro de controle é passado ao agente, que pode então começar a planejar;

4. A etapa de montagem de planos individuais é o processo padrão de planejamento do sistema CASPER, começando pela definição de horizonte e objetivo parcial;
5. Por não haver outros agentes para auxiliar na definição do plano a coordenação pode ser considerada como concluída assim que o plano parcial inicial for concluído;
6. Por utilizar de planejamento contínuo, a execução de ações ocorre com o plano parcial já definido e o processo volta para a etapa de planejamento individual.

O conceito de planejamento semi-autônomo como apresentado vai utilizar as primeiras etapas de maneira similar ao conceito do CASPER, com a diferenciação de que existe uma primeira passagem pelas etapas que ocorre apenas no supervisor, enquanto as etapas posteriores ocorrem em cada agente executor de ações. Para a correta definição dos pontos de parada entre estas duas passagens é necessário que o supervisor/centralizador conheça as capacidades e restrições de todos os agentes que vão participar do planejamento.

Neste novo modelo as etapas de coordenação são responsabilidade do centralizador, que gera planos de níveis de abstração variáveis, de acordo com a capacidade de cada agente executor. As etapas ficam então definidas da seguinte maneira:

1. O centralizador escolhe (internamente) como será feita a divisão de tarefas, levando em consideração as capacidades reportadas por cada agente;
2. No passo inicial não há necessidade de coordenação, pois o processo ainda está ocorrendo apenas no centralizador;
3. O centralizador faz o primeiro plano, utilizando uma versão adaptada do planejador HEART que usa um novo tipo de resolvidor para remover da agenda falhas de decomposição em ações abstratas do nível que cada agente afirmou ser capaz de resolver, até que não existam mais falhas a serem resolvidas ou todos os agentes tenham uma ação atribuída. Esta etapa gera subproblemas para cada agente disponível, que serão abordados posteriormente;
4. Após esta atribuição o planejador complementa os subproblemas com as ações atômicas antecessoras e sucessoras da ação abstrata que corresponde a cada subproblema, respeitando limitações dos agentes, para que todos os passos do plano sendo montado sejam executados;
5. Agora a responsabilidade de planejamento é passada aos agentes executores através da comunicação dos subproblemas atribuídos a cada um deles, e o planejamento passa para sua segunda parte, remota;
6. Neste ponto cada agente executor passa a montar seu plano individual, como se as etapas anteriores fossem as três primeiras etapas de um problema de planejamento multiagente regular;
7. Havendo falhas, cada agente pode tentar realizar o reparo iterativo de seu subproblema, e caso o subproblema não possua solução então o supervisor é avisado, e replaneja o problema tornando aquela ação abstrata indisponível àquele agente e atribuindo os novos passos a algum agente, ou a mesma ação abstrata a outro agente;
8. Ao resolver um subproblema o agente comunica ao supervisor quais as ações atômicas a serem tomadas, que são então incluídas no plano sendo gerado;

9. O problema é tido como resolvido quando o plano armazenado no centralizador atinge $lv(\pi) = \emptyset$, a agenda está vazia e todos os agentes executores já apresentaram resposta válida de seus subproblemas (pois ações atômicas não são resolvidas pelo centralizador, e portanto também precisam ser executadas por um agente). Caso ainda existam falhas na agenda após a resposta de todos os agentes (positiva ou negativa) então o processo volta à definição de subproblemas do passo 3, até que seja atingido o estado objetivo ou que não haja possibilidade de atribuir uma ação a agentes.

Estas etapas requerem novas estruturas ao processo padrão do HEART para acompanhamento de agentes e conferência de progresso, que serão apresentadas aqui.

Para adequação ao ambiente multiagente é necessário rever a definição do problema, conforme apresentado a seguir:

Definição 4.1. (Problema). O problema de planejamento passa a ser a tupla $\mathcal{P} = \langle \mathcal{D}, C_{\mathcal{P}}, a_0, \mathcal{E} \rangle$, onde \mathcal{E} é o conjunto de agentes executores do plano¹. O supervisor não precisa estar listado, e aos agentes pode ser passado o problema com a definição original caso não seja desejada interação entre eles.

Para a escolha do agente executor responsável por ações abstratas é necessário que o centralizador saiba todas as ações que cada agente executor é capaz de resolver.

Definição 4.2. (Ação). Uma ação passa a ser definida pela tupla parametrizada $a(args) = \langle nome, pre, eff, met, age \rangle$, onde $age \subseteq \mathcal{E}$ é o conjunto de agentes que são capazes de executar a ação. Para ações abstratas este conjunto implica que o agente também é capaz de executar todos os métodos presentes em met desta ação recursivamente.

Esta nova informação é utilizada na etapa inicial do novo processo, e permite ao centralizador saber durante a decomposição quais são os agentes que estão aptos a receber dada ação como subproblema, e também permite ao processo identificar quando uma ação está indisponível (nenhum agente é capaz de resolvê-la).

A atribuição de tarefas para agentes executores é feita a partir de um novo resolvidor decompositor, denominado *resolvidor de atribuição externa*.

Definição 4.3. (Atribuição externa). O resolvidor de atribuição externa funciona de maneira análoga ao resolvidor de decomposição, porém indica ao centralizador que a ação abstrata será passada para resolução por um agente executor. Sua notação reflete esta similaridade, sendo notado onde necessário como $a \bigoplus_{\pi}^{m,e}$, indicando que a ação a do plano parcial π está sendo atribuída para resolução pelo agente e através do método m .

Este resolvidor é aplicado durante a etapa de passagem de subproblemas, e caso haja resposta positiva do agente executor então o requerente é substituído pelas ações atômicas que o agente reportar como resposta de seu subproblema. Em caso de resposta negativa a falha é reinserida na agenda para resolução posterior, e o agente é removido do conjunto age da ação. Efeitos colaterais deste resolvidor são tratados exatamente igual aos resultantes da aplicação de um resolvidor decompositor.

Para que o centralizador possa conferir o andamento da atribuição de tarefas é necessário que todas as ações dos planos parciais possuam também um identificador de qual agente foi responsável pela sua solução.

¹Para evitar confusão com os passos apresentados nas definições originais do HEART não foi escolhida a letra \mathcal{A}

Definição 4.4. (Plano parcial). Ao plano parcial armazenado pelo planejador é inclusa a função $atr(a)$, que para cada ação $a \in \pi$ atribui um agente executor $e \in \mathcal{E}$ indicando que a ação a é de responsabilidade do agente e . Ações sem agente vinculado são atribuídas a algum agente durante a passagem de subproblemas (mesmo que atômicas), e o planejamento não é finalizado até que todas as ações do plano $\pi_{lv(c_0)}$ tenham sido atribuídas a um agente com sucesso.

Estas adequações são aplicadas ao algoritmo do POCL da seguinte maneira:

- A função *resolve* passa a considerar o conjunto \mathcal{E} durante a escolha de resolvedores e pode escolher o novo tipo de resolvidor onde adequado, sendo que sua prioridade é estritamente superior à resolução de falhas de decomposição padrão, para agilizar a passagem de subproblemas a agentes executores;
- Para resolvedores de atribuição externa as variáveis do conjunto R passam a conter o identificador do agente escolhido para a atribuição, e a cardinalidade do conjunto aumenta em até $|\mathcal{E}|$ devido à possibilidade de diversos agentes serem capazes de resolver dada ação;
- Quando r for atribuição externa a função *aplica* passa a ser *atribuir*, situação na qual a execução dos passos posteriores do laço passa a ser de responsabilidade do agente executor. O plano π' passado ao agente externo vai ser composto da ação requerente do resolvidor, assim como todas as ações atômicas antecessoras e sucessoras à ação abstrata sendo decomposta que o agente possa executar e ainda não tenham sido atribuídas, garantindo a completude da atribuição de ações;
- As funções *atualiza* e *reverte* passam a ter o agente escolhido como parâmetro, onde *atualiza* atribui às ações envolvidas nesta etapa o agente que as resolveu, enquanto *reverte* remove o agente do conjunto *age* das ações que lhe foram atribuídas e não puderam ser resolvidas.
- Ao final de cada ciclo o planejador sincroniza os agentes executores, esperando até que todos tenham atualizado a agenda, e caso necessário as ações. Neste momento as ações do plano parcial são analisadas para definir qual o nível de abstração do próximo ciclo e o processo se reinicia.

4.3 ANÁLISE TEÓRICA

Nesta seção é apresentado que esta variante do planejador HEART mantém as características de seus métodos e planos resultantes: O planejador é completo, correto e seus planos abstratos sempre podem ser decompostos em soluções válidas.

A completude e corretude de POCL já foi provada em (Penberthy e Weld, 1992), e as mesmas características já foram provadas para o planejador HEART em (Grea et al., 2018). Aqui será provado que as alterações feitas ao planejador não comprometem estas características. Uma propriedade relevante dos algoritmos POCL é que a estratégia de escolha de falhas não compromete sua corretude/completude. Ressaltando que para estas provas o caractere $:$ quando apresentado na expressão $var : exp$ deve ser lido como $var \text{ tal que } exp$.

Teorema 1. (Atribuição externa preserva aciclicidade). *A atribuição de uma ação composta com métodos válidos em um plano acíclico a um agente executor capaz de resolvê-la resulta em um plano acíclico.* Formalmente: $\forall a_s \in S_\pi : a_s \not\prec_\pi a_s \implies \forall a'_s \in S_a \oplus_\pi^{m,e} : a'_s \not\prec_a \oplus_\pi^{m,e} a'_s$.

Demonstração. A atribuição externa funciona de maneira idêntica à decomposição normal para efeito de controle do plano, sendo sua única diferença o fato de sua aplicação acontecer remotamente, portanto: Ao realizar a atribuição externa da ação composta a com um método válido m para um agente executor e capacitado a resolvê-la dentro de um plano existente π , todos os passos S_m são adicionados ao plano refinado. Tanto π quanto m são garantidos por definição como acíclicos.

Podemos notar que $\forall a_s \in S_m : \left(\nexists a_t \in S_m : a_s > a_t \wedge \neg f \in \text{eff}(a_t) \right) \implies f \in \text{eff}(a)$.

Por extenso, se uma ação a_s contribui um fluente f para o objetivo do método m então f está necessariamente presente nos efeitos de a . Como a escolha de ações dá preferência às de maior nível de abstração durante a escolha de resolvidores então nenhuma ação de seus métodos vai estar presente no plano parcial atual quando a atribuição externa ocorrer. Isto pode ser notado como $\exists a \in \pi \implies S_m \sqcup S_\pi$, significando que o grafo formado pelos planos parciais m e π não podem conter as mesmas arestas e portanto sua aciclicidade é preservada em sua inserção. \square

Teorema 2. (Atribuições externas resolvidas não reocorrem). *A aplicação de um resolvidor de atribuição externa sobre um plano π garante que $a \notin S'_\pi$ para todo plano parcial refinado a partir de π em que não houver reversão de a .*

Demonstração. Como apresentado na definição de métodos (Definição 3.3), temos que $a \notin A_a$. Isto significa que a resolução por atribuição externa não pode introduzir a através de sua decomposição ou da decomposição de suas ações próprias, mesmo que resolvidas por um agente executor. De fato, após sua expansão e a finalização do ciclo atual, pela diminuição do nível do ciclo para $c_{lv(a)-1}$ sua escolha passa a ser impedida pelo seletor de resolvidores. Tal ação não pode estar contida em métodos de ações selecionadas posteriormente pois caso contrário, pela definição de níveis de abstração (Definição 3.12), seu nível de abstração seria ao menos $lv(a) + 1$. \square

Teorema 3. (Atingir o nível de abstração 0 com todas as ações atômicas atribuídas a um agente garante solubilidade). *Encontrar um plano que contenha apenas falhas de atribuição externa ou decomposição com ações de nível de abstração 1 e com agentes executores capazes de resolvê-las garante uma solução ao problema.*

Demonstração. A introdução da necessidade de atribuir ações (mesmo que atômicas) a agentes torna este ponto mais complexo em relação à versão original do planejador HEART.

Qualquer método m disponível ao agente e de uma ação composta $a : lv(a) = 1$ é por definição uma solução do problema atribuído ao agente $\mathcal{P}_a = \langle \mathcal{D}, \mathcal{C}_\mathcal{P}, a \rangle$. Como por definição $a \notin A_a$ e $a \notin A_a \bigoplus_\pi^{m,e}$ (significando que a não reocorre após decomposta pelo agente executor). Também por definição temos que a instanciação da ação e seus métodos mantém coerência das restrições (tudo está instanciado antes da escolha pelo resolvidor). Como o plano π possui apenas falhas de decomposição e de atribuição externa (que por definição não diferem em referência ao controle de geração do plano) e todas as falhas dentro de m possuem garantia de solubilidade, e ambas possuem garantia de aciclicidade através da decomposição/atribuição pela aplicação de qualquer $a \bigoplus_\pi^{m,e}$, então o plano possui solução. \square

Teorema 4. (Planos abstratos garantem solubilidade). *Encontrar um plano parcial π que possua apenas falhas de decomposição/atribuição garante a existência de solução do problema.*

Demonstração. Recursivamente, ao aplicar a demonstração anterior ao plano de nível de abstração superior notamos que a decomposição ao nível 2 garante solução, pois os métodos das ações compostas possuem garantia de solubilidade.

□

A ressalva necessária à demonstração do teorema 4 é de que a informação das capacidades dos agentes executores precisa estar correta para preservar a prova, o que pode não ser verdade em ambientes multiagente em que a privacidade escolhida pelos agentes impeça ou limite a passagem das limitações de capacidade ao supervisor. Para os experimentos a serem apresentados neste trabalho sempre haverá suposição desta corretude.

4.3.1 Aplicabilidade teórica e expansões possíveis

Com base nos resultados teóricos apresentados pode-se afirmar que a utilização desta variante do planejamento proposto pelo HEART pode ser utilizada em ambientes multiagente cooperativos, se aproveitando do mecanismo de divisão de responsabilidade para permitir a agentes de capacidade reduzida a resolução de problemas impossíveis sem auxílio. A maior restrição para a utilização da ferramenta como implementada é a mesma que o planejamento HTN já possui, na necessidade de mapear os problemas para que possua ações abstratas corretamente tipadas e agrupadas de acordo com o resultado esperado pelo domínio, exacerbado pelo uso de notação diferente da mais usual a HTN que o planejador requer e pelo novo parâmetro de identificação de agentes capazes de executar cada ação abstrata.

No tocante a expansões, existe pesquisa para a utilização de Ontologias com o exposto intuito de auxiliar a criação de domínios para planejamento (Behnke et al., 2015) que poderia melhorar a aplicabilidade do planejador, assim como o planejador poderia ser alterado para utilizar identificação de subproblemas independentes em domínios grandes (Lotem e Nau, 2000), permitindo melhora na escolha de ações e agentes durante a atribuição externa. O processo de reparo de planos também poderia ser alterado para o proposto em (Höller et al., 2018), que afirma ser mais enxuto e compatível à estrutura HTN. Referente à teoria da técnica, este mesmo conceito poderia também ser aplicado à variantes de HTN mais voltadas à acompanhamento de agentes, tal como redes GTN (Alford et al., 2016) ou TMK (Hoang et al., 2005).

No próximo capítulo será apresentada a análise prática do planejamento semi-autônomo tal como apresentado.

5 EXPERIMENTOS

Neste capítulo serão apresentadas as validações e análises da técnica apresentada. Serão apresentadas as características dos experimentos práticos, seus resultados e uma breve análise de como isso afeta a viabilidade de uso da versão implementada.

5.1 ESTRUTURAÇÃO E METODOLOGIA

O planejador HEART original está implementado na linguagem Java, configurado para utilizar uma única thread de processamento e sem limitações de uso de memória ou tempo de execução¹. A variante utilizada mantém estas características para permitir a validação através de comparação também com o planejador original, caso desejado.

Todos os testes utilizaram um computador com processador Intel i7-4510U com clock de 2 GHz e 16 GB de memória RAM (para garantir que utilização de RAM não seja um gargalo) e cada experimento foi repetido não menos que 100 vezes para minimizar o impacto de variações de velocidade nos resultados obtidos. Os critérios analisados são a qualidade do plano abstrato referente à instanciação de fluentes durante cada ciclo de refinamento, complexidade referente ao tempo de processamento para domínios com características diferentes, e o potencial de paralelização da atribuição externa, que consiste em verificar a quantidade de subproblemas que poderiam ser atribuídos a agentes executores para planejamento em paralelo a cada ciclo do refinamento.

Para melhor controle os testes de prova conceitual foram realizados utilizando instâncias aleatórias dentro de domínios artificiais, para tanto poupar variações decorrentes da tradução de domínios para a notação esperada pelo planejador HEART quanto permitir uma análise comparativa mais enxuta, estimando melhor a performance ideal que a técnica possui em domínios genéricos e evitando a tomada de resultados tendenciosos a algum tipo de domínio, conforme exposto na seção sobre metodologia de (Barrett e Weld, 1994). Esta regularidade faz parte da análise, sendo aplicadas variações de nível máximo de abstração e aridade dos nodos correspondentes a cada ação abstrata.

5.2 RESULTADOS

Nesta seção serão motivados e explicados cada um dos gráficos gerados, e será exposto o que cada um indica para a viabilidade da técnica em aplicação direta.

5.2.1 Instanciação de fluentes

A tabela 5.1 apresenta os resultados do teste de descoberta a instanciação de fluentes através do processo de refinamento. Conforme apresentado na definição 3.3 o planejador HEART não requer a instanciação de todos seus fluentes antes do início do planejamento, portanto é necessário que esta tipagem ocorra durante o refinamento. Esta análise mostra qual o impacto desta característica relativo ao nível de abstração do problema sendo abordado, sendo a contagem dos fluentes presentes em $\bigcup_{a \in S_\pi} eff(a)$, e esta métrica pode ser utilizada para aproximar a probabilidade de existência de objetivo com base em observação de domínio (utilizada em reconhecimento de planos, tal como em (Riabov e Udrea, 2016)).

¹conforme o artigo original, seu código fonte está disponível em <http://genn.io/heart>

Tabela 5.1: Instanciação de fluentes

Nível de Abstração	Fluentes instanciados (%)	Tempo de execução (ms)
HEART (4)	44	0
HEART (3)	60	1,3 ±0,51
HEART (2)	74	2,7 ±0,67
HEART (1)	88	4,2 ±0,85
HEART (0)	100	6,5 ±1,04
POP (0)	100	6,0 ±0,86
HiPOP (0)	100	11,3 ±2,41

Tabela 5.2: Refinamento (n=4)

Nível de Abstração	Tempo médio (ms)	1º Quartil	3º Quartil	Ações abstratas
5	2,743642	1,680150	2,667050	4
4	7,136958	4,808800	7,258875	16
3	41,737807	29,666325	44,142125	64
2	81,070437	51,924100	98,065700	256
1	81,292585	52,193125	98,145900	0

A comparação com POP e HiPOP serve para mostrar que em algumas instâncias pode ser mais eficiente utilizar a instanciação parcial do HEART (mesmo na versão variante apresentada) do que esperar o aterramento completo de fluentes utilizado pelos planejadores nos quais o HEART se baseia. A saber, a diferença entre os planejadores POP e HiPOP é a utilização da estrutura hierárquica pelo HiPOP, que é suprimida em execuções do POP. Estes valores são adquiridos apenas através da transposição das ações próprias de ações abstratas, portanto antes do processo de planejamento propriamente dito, e por permitir três quartos do total de fluentes em menos da metade do tempo podem ser úteis como meio-termo entre qualidade e tempo gasto.

5.2.2 Tempo de processamento do supervisor e paralelização

As tabelas 5.2, 5.3 e 5.4 apresentam o comportamento do planejador durante os ciclos de refinamento. Cada tabela assume o nível de abstração máximo do domínio como 5 (significando que $lv(\pi_0) = 5$), e apresenta o tempo gasto pelo planejador para atingir o fim de cada ciclo a partir do início. Neste teste foram utilizados domínios artificiais compostos de uma única ação abstrata de nível 5 composta de uma quantidade n de ações de nível imediatamente inferior ² de maneira recursiva, formando uma estrutura similar a uma árvore de aridade n . Todas as ações

² n pode ser chamada de a *largura* do domínio

Tabela 5.3: Refinamento (n=3)

Nível de Abstração	Tempo médio (ms)	1º Quartil	3º Quartil	Ações abstratas
5	2,512098	1,371025	2,469800	3
4	6,784925	4,472925	7,004725	9
3	40,539433	27,104500	47,440475	27
2	79,531228	46,428900	92,974700	81
1	79,744913	46,629075	93,212325	0

Tabela 5.4: Refinamento (n=2)

Nível de Abstração	Tempo médio (ms)	1º Quartil	3º Quartil	Ações abstratas
5	2,488476	1,360575	2,633075	2
4	6,896970	4,661725	6,972200	4
3	39,936906	28,243975	43,744250	8
2	79,198726	52,468025	91,577600	16
1	79,434450	52,839500	91,811450	0

Tabela 5.5: Tempo de processamento após atribuição externa

Abstração inicial	Tempo total (ms)	Memória utilizada (Kb)
Nível 4	359,4252	77291
Nível 3	63,6504	14336
Nível 2	26,0766	8033
Nível 1	16,4875	7019

atômicas destes domínios possuem efeito sobre um único fluente, o objetivo é o efeito da ação abstrata de nível 5 e o estado inicial está vazio.

Para cada largura n foi calculado o tempo médio da execução do refinamento por ciclo, seus limites inferior e superior da metade central dos valores adquiridos e a quantidade de ações abstratas que serão tratadas no próximo ciclo. O comportamento do nível de abstração 1 é atípico porém consistente, e parece ser consequência do mecanismo de transposição das ações atômicas durante a montagem do problema gerar maior variação, estando presente na análise do planejador HEART original e ocorrendo sempre apenas no ciclo de refinamento do último nível ainda com ações abstratas. Como os domínios utilizados são artificiais então a quantidade de ações abstratas corresponde a $n^{lv(\pi)-lv(\pi_{c+1})}$ para cada ciclo c , até que para $lv(c) = 1$ existem 0 ações abstratas e $n^{lv(\pi)}$ ações atômicas.

O comportamento computacional do planejador segue a regra exponencial característica da complexidade P-SPACE do problema de planejamento (Erol et al., 1995). mas cuja função de tendência possui comportamento próximo de linear nos primeiros ciclos, o que viabiliza o uso deste planejador como supervisor para a atribuição externa nos níveis superiores, evitando a explosão exponencial.

5.2.3 Tempo de processamento do executor

A tabela 5.5 possui o tempo total de execução do processo de planejamento após a passagem para o executor externo. Este tempo está listado de acordo com o nível de abstração do subproblema passado, sendo que o "planejamento" para ações atômicas é trivial³ pois a ordenação recebida pelo agente executor é mantida e retornada. Novamente é clara a complexidade com explosão exponencial, mas pode-se ver que a paralelização dos níveis de abstração mais baixos passa a ser vantajosa. Supondo um problema de largura 4, a aplicação do refinamento até o nível de abstração 4 (em média 7,14 milissegundos) pode então ser atribuída a 16 agentes executores que finalizarão seus planos em aproximadamente 63 segundos cada, finalizando todo o processo em 70 segundos.

³exceto caso o problema seja relaxado para permitir planejamento temporal, quando então o agendamento de ações passa a ser necessário mesmo para ações atômicas

5.2.4 Análise

Com base nos valores apurados na seção anterior pode-se afirmar que a técnica possui aplicabilidade prática, mas seu comportamento carece de uma análise mais profunda. A anomalia presente na última etapa de refinamento parece ser consequência de um processo interno da implementação do planejador, então para melhor clareza dos resultados seria proveitosa uma análise complementar desta técnica aplicada a outro planejador hierárquico, mas não foi encontrado em publicações de vulto nenhum outro planejador que apresente a capacidade de gerar planos abstratos através do uso de HTN.

Outro aspecto que pode ser melhorado é o tempo necessário ao agente executor para realizar o recebimento de cada subproblema, pois na implementação atual é necessário que o agente executor refaça o instanciamento do domínio (e valide todas as ações) para então iniciar o processo de refinamento de seu subproblema. Esta necessidade poderia ser evitada com uma estrutura validada de antemão pelo agente, ou até mesmo pela passagem do domínio já limitado pelas ações abstratas (e todas suas ações próprias) e ações atômicas sendo passadas.

Em suma pode-se dizer que a técnica de planejamento semi-autônomo como proposta representa uma alternativa viável e válida para o processamento segmentado de planos multiagente utilizando a estrutura HTN, e seu grande potencial de paralelização pode se mostrar valioso em cenários tais como planejamento distribuído de enxame (destacando um agente como supervisor dos outros), desde que respeitadas ou contornadas suas necessidades atuais de cooperação completa e ausência de privacidade dos agentes em relação ao supervisor.

Como a estrutura HTN é aplicável para outras aplicações de planejamento não clássico e o planejador é completo e correto, então sua extensão pode ser útil em outros cenários fora do escopo deste trabalho, tais como planejamento com restrição de recursos. Também devido à estrutura HTN este planejador pode ser utilizado em sistemas de pesquisa de ponta sobre planejamento supervisionado, tal como planejamento de iniciativa mista (Ramoul, 2018; IJtsma et al., 2019).

6 CONCLUSÃO

Neste trabalho foi apresentado o conceito de *Planejamento semi-autônomo*, que consiste em dividir a responsabilidade de planejamento entre um agente dedicado ao acompanhamento do problema e outros agentes, responsáveis pelo planejamento em menor escala e execução do plano final. Esta nova abordagem pode permitir a utilização de agentes de capacidade reduzida em cenários mais complexos, bastando que haja um supervisor/centralizador para a coordenação e atribuição do problema em pedaços reduzidos aos agentes.

Inicialmente o foco do trabalho foi em apresentar a abordagem alternativa sem se ater à cenários de aplicação em mundo real, analisando sua viabilidade como técnica para análise acadêmica de problemas multiagente cooperativos. Através da prova da manutenção das características do planejador HEART foi provado que a técnica possui completude e corretude mesmo antes da conclusão do planejamento, o que permite sua aplicação em análises voltadas à aproximação de resultados em cenários sem solução ou com limites de tempo que impeçam a finalização do plano. Ao final desta análise foram apresentadas expansões que permitiriam maior adequação do planejador às técnicas mais recentes para planejamento HTN, assim como sugestões de variações adicionais que talvez apresentem melhor compatibilidade quando comparada à estrutura padrão de uma HTN.

Em análise de performance prática o planejador modificado apresentou comportamento muito similar à versão original, o que não é surpresa tendo em vista o foco em alterações de pouco impacto na complexidade e ausência de otimizações ao processo. A inclusão do novo resolvidor não aumentou significativamente o tempo gasto com a expansão dos planos abstratos iniciais, então devido às características do planejamento multiagente dar preferência pela sua utilização depende das restrições sobre o tempo gasto com a comunicação entre os agentes. Porém, na análise do potencial de paralelização ficou aparente o que pode se tornar a maior contribuição desta nova técnica na sua capacidade de gerar uma grande quantidade de planos parciais em um intervalo relativamente curto de tempo. Esta capacidade pode se mostrar útil no controle de exames com grau mínimo de interferência humana durante o processo, embutindo ao controle de fluxo os objetivos de cada grupo de agentes e atribuindo este controle a um dos participantes do exame.

Levando em consideração a implementação como utilizada neste trabalho as maiores limitações são de características já conhecidas da estrutura HTN, tal como a necessidade de criação de domínios por um especialista e o fato de tanto o planejador HEART quanto a variante proposta utilizarem de uma linguagem de representação dos domínios e problemas própria, necessitando tradução mesmo quando o domínio já se encontra escrito em HTN. Talvez esta limitação de tradução pode ser mitigada através da aplicação do mesmo conceito em outros planejadores hierárquicos com mais tempo de publicação e maior adesão, como a linguagem utilizada pelo SHOP2 (Nau et al., 2003) e seus diversos desdobramentos.

Pesquisas futuras na mesma linha que a apresentada podem ser realizadas de diversas maneiras. Caso seja dado foco em aplicações comerciais imediatas pode-se realizar a melhoria da performance em ambiente real através de heurísticas dedicadas (Alford et al., 2014; Bercher et al., 2017) ou otimização do código. Na área de extensão conceitual, sua aplicabilidade pode ser ampliada através da inclusão de relaxamentos ao problema de planejamento abordado, tal como a aproximação ao conceito original do CASPER pela conversão do processo para planejamento

contínuo¹, se aproveitando da atribuição de ações abstratas para guiar o horizonte de cada etapa de planejamento. Outra possibilidade de pesquisa futura promissora seria o foco na característica de criação rápida de planos paralelos para a implementação em enxames similares às apresentadas em (Mufti et al., 2019; DeMedeiros e Wettergren, 2019). Tangencialmente, este trabalho também pode ter sua aplicabilidade estendida através de sua vinculação com estruturas de Ontologia para realizar a consolidação de informações sobre domínios mais diversos, de maneira similar às apresentadas em (Bihlmaier et al., 2016; Santofimia et al., 2017).

¹Planejamento em mundo aberto seria muito interessante, mas não foi encontrado durante a pesquisa nenhum precedente para o aprendizado de literais e ações abstratas em tempo de execução

REFERÊNCIAS

- Alford, R., Shivashankar, V., Kuter, U. e Nau, D. (2014). On the feasibility of planning graph style heuristics for htn planning. Em *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling*, páginas 2–10. AAAI Press.
- Alford, R., Shivashankar, V., Roberts, M., Frank, J. e Aha, D. W. (2016). Hierarchical planning: Relating task and goal decomposition with task sharing. Em *IJCAI*, páginas 3022–3029.
- Ambros-Ingerson, J. A. e Steel, S. (1988). Integrating planning, execution and monitoring. Em *AAAI*, volume 88, páginas 21–26.
- Barrett, A. e Weld, D. S. (1994). Partial-order planning: evaluating possible efficiency gains. *Artificial Intelligence*, 67(1):71–112.
- Bechon, P., Barbier, M., Infantes, G., Lesire, C. e Vidal, V. (2014). Hipop: Hierarchical partial-order planning. Em *STAIRS*, páginas 51–60.
- Behnke, G., Bercher, P., Biundo-Stephan, S., Glimm, B., Ponomaryov, D. K. e Schiller, M. R. G. (2015). Integrating ontologies and planning for cognitive systems. Em *Description Logics*.
- Bercher, P., Behnke, G., Höller, D. e Biundo, S. (2017). An admissible htn planning heuristic. Em *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, páginas 480–488. AAAI Press.
- Bercher, P., Keen, S. e Biundo, S. (2014). Hybrid planning heuristics based on task decomposition graphs. Em *Seventh Annual Symposium on Combinatorial Search*.
- Bihlmaier, A., Schreiter, L., Raczowsky, J. e Wörn, H. (2016). Hierarchical task networks as domain-specific language for planning surgical interventions. Em *Intelligent Autonomous Systems 13*, páginas 1095–1105. Springer.
- Bonet, B. e Geffner, H. (2000). Planning with incomplete information as heuristic search in belief space. Em *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems*, páginas 52–61. AAAI Press.
- Boutilier, C., Dean, T. e Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11(1):94.
- Burns, E., Kiesel, S. e Ruml, W. (2013). Experimental real-time heuristic search results in a video game. Em *Sixth Annual Symposium on Combinatorial Search*.
- Bylander, T. (1994). The computational complexity of propositional strips planning. *Artificial Intelligence*, 69(1-2):165–204.
- Chanel, C. P. C., Albore, A., T’Hooft, J., Lesire, C. e Teichteil-Königsbuch, F. (2019). Ample: an anytime planning and execution framework for dynamic and uncertain problems in robotics. *Autonomous Robots*, 43(1):37–62.
- Clearwater, S. H. (1996). *Market-based control: A paradigm for distributed resource allocation*. World Scientific.

- Contreras-Cruz, M. A., Lopez-Perez, J. J. e Ayala-Ramirez, V. (2017). Distributed path planning for multi-robot teams based on artificial bee colony. Em *2017 IEEE Congress on Evolutionary Computation (CEC)*, páginas 541–548. IEEE.
- Cox, J. S. e Durfee, E. H. (2005). An efficient algorithm for multiagent plan coordination. Em *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, páginas 828–835. ACM.
- DeMedeiros, K. D. e Wettergren, T. A. (2019). Control of multi-agent swarms with cooperative particle swarm optimization. Relatório técnico, NUWC Division Newport Newport.
- Durfee, E. H. (2001). Distributed problem solving and planning. Em *ECCAI Advanced Course on Artificial Intelligence*, páginas 118–149. Springer.
- Erol, K., Hendler, J. e Nau, D. S. (1994a). HTN planning: Complexity and expressivity. Em *AAAI*, volume 94, páginas 1123–1128.
- Erol, K., Hendler, J. A. e Nau, D. S. (1994b). UMCP: A sound and complete procedure for hierarchical task-network planning. Em *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems*, volume 94, páginas 249–254.
- Erol, K., Nau, D. S. e Subrahmanian, V. S. (1995). Complexity, decidability and undecidability results for domain-independent planning. *Artificial intelligence*, 76(1-2):75–88.
- Fikes, R. E. e Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208.
- Fox, M. e Long, D. (2003). PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research*.
- Fox, M., Long, D. e Magazzeni, D. (2017). Explainable planning. Em *In Proceedings of IJCAI Workshop on Explainable AI*.
- Fox, M. S. (1994). Isis: A retrospective. *Intelligent Scheduling*.
- Gerevini, A., Kuter, U., Nau, D., Saetti, A. e Waisbrot, N. (2008). Combining domain-independent planning and htn planning: The duet planner. Em *Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, páginas 573–577. IOS Press.
- Gerkey, B. P. e Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954.
- Ghallab, M., Howe, A., Knoblock, C., McDermott, D., Ram, A., Veloso, M., Weld, D. e Wilkins, D. (1998). Pddl—the planning domain definition language. *AIPS-98 planning committee*, 3:14.
- Ghallab, M., Nau, D. e Traverso, P. (2004). *Automated Planning: theory and practice*. Elsevier.
- Ghallab, M., Nau, D. e Traverso, P. (2014). The actor’s view of automated planning and acting: A position paper. *Artificial Intelligence*, 208:1–17.
- Ghallab, M., Nau, D. e Traverso, P. (2016). *Automated planning and acting*. Cambridge University Press.

- Giunchiglia, E. e Maratea, M. (2007). Sat-based planning with minimal-#actions plans and “soft” goals. Em Basili, R. e Pazienza, M. T., editores, *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, páginas 422–433, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Göbelbecker, M., Keller, T., Eyerich, P., Brenner, M. e Nebel, B. (2010). Coming up with good excuses: What to do when no plan can be found. Em *Twentieth International Conference on Automated Planning and Scheduling*, volume 20, páginas 81–88. AAAI Press.
- Godoy, J., Karamouzas, I., Guy, S. J. e Gini, M. L. (2016). Implicit coordination in crowded multi-agent navigation. Em *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, páginas 2487–2493. AAAI press.
- Gréa, A., Matignon, L. e Aknine, S. (2018). HEART: HiErarchical Abstraction for Real-Time Partial Order Causal Link Planning. Em *1st Workshop on Hierarchical Planning at 28th International Conference on Automated Planning and Scheduling (ICAPS)*, páginas 17–25.
- Grea, A., Matignon, L. e Aknine, S. (2018). How explainable plans can make planning faster. Em *Workshop on Explainable Artificial Intelligence*, páginas 58–64.
- Gunning, D. (2019). Darpa’s explainable artificial intelligence (xai) program. Em *Proceedings of the 24th International Conference on Intelligent User Interfaces*, páginas ii–ii. ACM.
- Hernández, C. e Baier, J. A. (2012). Avoiding and escaping depressions in real-time heuristic search. *Journal of Artificial Intelligence Research*, 43:523–570.
- Hoang, H., Lee-Urban, S. e Muñoz-Avila, H. (2005). Hierarchical plan representations for encoding strategic game ai. Em *Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, páginas 63–68. AAAI Press.
- Höller, D., Bercher, P., Behnke, G. e Biundo, S. (2018). Htn plan repair using unmodified planning systems. Em *Proc. of the First ICAPS Workshop on Hierarchical Planning*, páginas 26–30.
- IJtsma, M., Lassiter, W., Feigh, K. M., Savelsbergh, M. e Pritchett, A. R. (2019). An integrated system for mixed-initiative planning of manned spaceflight operations. Em *2019 IEEE Aerospace Conference*, páginas 1–8. IEEE.
- Ingrand, F. e Ghallab, M. (2017). Deliberation for autonomous robots: A survey. *Artificial Intelligence*, 247:10–44.
- Jennings, N. R. (1999). Agent-based computing: Promise and perils. Em *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, páginas 1429–1436. Morgan Kaufmann Publishers Inc.
- Kim, J., Kim, H., Lakshmanan, K. e Rajkumar, R. R. (2013). Parallel scheduling for cyber-physical systems: Analysis and case study on a self-driving car. Em *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*, páginas 31–40. ACM.
- Knight, S., Rabideau, G., Chien, S., Engelhardt, B. e Sherwood, R. (2001). Casper: Space exploration through continuous planning. *IEEE Intelligent Systems*, 16(5):70–75.

- Komenda, A., Stolba, M. e Kovacs, D. L. (2016). The international competition of distributed and multiagent planners (codmap). *AI Magazine*, 37(3):109–115.
- Korf, R. E. (1990). Real-time heuristic search. *Artificial intelligence*, 42(2-3):189–211.
- Kruijff, G.-J. M., Janíček, M., Keshavdas, S., Larochelle, B., Zender, H., Smets, N. J., Mioch, T., Neerincx, M. A., Diggelen, J. V., Colas, F. et al. (2014). Experience in system design for human-robot teaming in urban search and rescue. Em *Field and Service Robotics*, páginas 111–125. Springer.
- Lotem, A. e Nau, D. S. (2000). New advances in GraphHTN: identifying independent subproblems in large HTN domains. Em *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems*, páginas 206–215. AAAI Press.
- McAllester, D. e Rosenblitt, D. (1991). Systematic nonlinear planning. Em *Proceedings of the ninth National conference on Artificial intelligence-Volume 2*, páginas 634–639. AAAI Press.
- McDermott, D. M. (2000). The 1998 AI planning systems competition. *AI magazine*, 21(2):35.
- Montavon, G., Binder, A., Lapuschkin, S., Samek, W. e Müller, K.-R. (2019). Layer-wise relevance propagation: an overview. Em *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, páginas 193–209. Springer.
- Mufti, S., Roberge, V. e Tarbouchi, M. (2019). *Path Planning Algorithm for Multiple Unmanned Aerial Vehicles with Parallel Implementation on a Graphics Processing Unit*. Tese de doutorado, Royal Military College of Canada.
- Nau, D. S., Au, T.-C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D. e Yaman, F. (2003). SHOP2: An HTN planning system. *Journal of artificial intelligence research*, 20:379–404.
- Newell, A., Shaw, J. C. e Simon, H. A. (1959). Report on a general problem solving program. Em *IFIP congress*, volume 256, página 64. Pittsburgh, PA.
- Palacios, H. e Geffner, H. (2009). Compiling uncertainty away in conformant planning problems with bounded width. *Journal of Artificial Intelligence Research*, páginas 623–675.
- Penberthy, J. S. e Weld, D. S. (1992). Ucpop: a sound, complete, partial order planner for adl. Em *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, páginas 103–114. Morgan Kaufmann Publishers Inc.
- Penberthy, J. S. e Weld, D. S. (1994). Temporal planning with continuous change. Em *AAAI*, volume 94, página 1010.
- Rabideau, G., Knight, R., Chien, S., Fukunaga, A. e Govindjee, A. (1999). Iterative repair planning for spacecraft operations using the ASPEN system. Em *Artificial Intelligence, Robotics and Automation in Space*, volume 440, página 99.
- Ramirez, M. e Geffner, H. (2009). Plan recognition as planning. Em *21st International Joint Conference on Artificial Intelligence*, páginas 1778–1783. The Association for the Advancement of Artificial Intelligence.
- Ramoul, A. (2018). *Système de planification en mixed-initiative pour l'assistance à la gestion des systèmes informatisés complexes*. Tese de doutorado, Université Grenoble Alpes.

- Riabov, S. S. A. V. e Udrea, O. (2016). Plan recognition as planning revisited. Em *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 63, páginas 3258–3264.
- Sacerdoti, E. D. (1975). The nonlinear nature of plans. Relatório técnico, Stanford Research Institute.
- Santofimia, M. J., Martinez-del Rincon, J., Hong, X., Zhou, H., Miller, P., Villa, D. e Lopez, J. C. (2017). Hierarchical task network planning with common-sense reasoning for multiple-people behaviour analysis. *Expert Systems with Applications*, 69:118–134.
- Seegebarth, B., Müller, F., Schattenberg, B. e Biundo, S. (2012). Making hybrid plans more clear to human users—a formal approach for generating sound explanations. Em *Twenty-Second International Conference on Automated Planning and Scheduling*, páginas 225–233. AAAI Press.
- Sharon, G., Sturtevant, N. e Felner, A. (2013). Online detection of dead states in real-time agent-centered search. Em *Sixth Annual Symposium on Combinatorial Search*.
- Shehory, O. e Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial intelligence*, 101(1-2):165–200.
- Sherwood, R., Mishkin, A., Chien, S., Estlin, T., Backes, P., Cooper, B., Rabideau, G. e Engelhardt, B. (2014). An integrated planning and scheduling prototype for automated mars rover command generation. Em *Sixth European Conference on Planning*.
- Shimbo, M. e Ishida, T. (2003). Controlling the learning process of real-time heuristic search. *Artificial Intelligence*, 146(1):1–41.
- Shortliffe, E. (2012). *Computer-based medical consultations: MYCIN*, volume 2. Elsevier.
- Sreedharan, S., Srivastava, S., Smith, D. e Kambhampati, S. (2019). Why can't you do that hal? explaining unsolvability of planning tasks. Em *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, páginas 1422–1430. AAAI Press.
- Talamadupula, K., Benton, J., Kambhampati, S., Schermerhorn, P. e Scheutz, M. (2010). Planning for human-robot teaming in open worlds. *ACM Trans. Intell. Syst. Technol.*, 1(2):14:1–14:24.
- Tate, A. (1977). Generating project networks. Em *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 2*, páginas 888–893. Morgan Kaufmann Publishers Inc.
- Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixia, I. L., Ruess, F., Suppa, M. e Burschka, D. (2012). Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *IEEE Robotics Automation Magazine*, 19(3):46–56.
- Weld, D. S. (1994). An introduction to least commitment planning. *AI magazine*, 15(4):27–27.
- Weld, D. S. (1999). Recent advances in ai planning. *AI magazine*, 20(2):93.
- Wellman, M. P., Walsh, W. E., Wurman, P. R. e MacKie-Mason, J. K. (2001). Auction protocols for decentralized scheduling. *Games and economic behavior*, 35(1-2):271–303.

Winikoff, M., Dignum, V. e Dignum, F. (2018). Why bad coffee? explaining agent plans with valuations. Em *International Conference on Computer Safety, Reliability, and Security*, páginas 521–534. Springer.

Zilberstein, S. e Russell, S. (1995). Approximate reasoning using anytime algorithms. Em *Imprecise and approximate computation*, páginas 43–62. Springer.