

UNIVERSIDADE FEDERAL DO PARANÁ

PAULO AMARO VELLOSO HENRIQUES DOS SANTOS

RESOLUÇÃO DO PROBLEMA DE CARREGAMENTO DE CONTAINER E DE
ROTEAMENTO DE VEÍCULOS UTILIZANDO ALGORITMOS GENÉTICOS

CURITIBA

2011

PAULO AMARO VELLOSO HENRIQUES DOS SANTOS

RESOLUÇÃO DO PROBLEMA DE CARREGAMENTO DE CONTAINER E DE
ROTEAMENTO DE VEÍCULOS UTILIZANDO ALGORITMOS GENÉTICOS

Dissertação apresentada ao Curso de Pós-Graduação em Métodos Numéricos em Engenharia, Área de Concentração em Programação Matemática, Setores de Tecnologia e Ciências Exatas, Universidade Federal do Paraná, como parte das exigências para a obtenção do título de Mestre em Ciências.

Orientador: Prof. Dr. Arinei Carlos Lindbeck da Silva.

CURITIBA

2011

TERMO DE APROVAÇÃO

PAULO AMARO VELLOSO HENRIQUES DOS SANTOS

RESOLUÇÃO DO PROBLEMA DE CARREGAMENTO DE CONTAINER E DE ROTEAMENTO DE VEÍCULOS UTILIZANDO ALGORITMOS GENÉTICOS

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Curso de Pós-Graduação em Métodos Numéricos em Engenharia, Setor de Tecnologia da Universidade Federal do Paraná, pela seguinte banca examinadora:

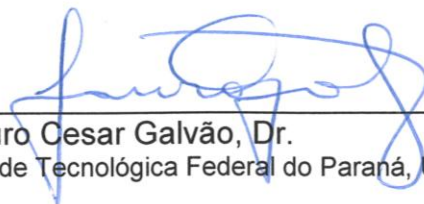
Orientador:



Prof. Arinei Carlos Lindbeck da Silva, Dr.
Programa de Pós-Graduação em Métodos Numéricos em Engenharia, UFPR



Prof. Volmir Eugênio Wilhelm, Dr.
Programa de Pós-Graduação em Métodos Numéricos em Engenharia, UFPR



Prof. Lauro Cesar Galvão, Dr.
Universidade Tecnológica Federal do Paraná, UTFPR

Curitiba, 30 de setembro de 2011

À Laininha e Laurinha
Vocês são os amores da minha vida
Foi por vocês!

AGRADECIMENTOS

A Deus, pela oportunidade de poder estudar e pela saúde e força recebidas mesmo nos momentos de cansaço.

A minha amada Elaine, pela compreensão nas muitas noites em que tive que me dedicar aos estudos e não fui me deitar ao seu lado, pelo apoio que nunca me faltou mesmo nas horas mais difíceis, por me acompanhar nesses últimos meses viajando de Joinville pra Curitiba quase todas as semanas e por acreditar em mim todos os nossos dias. TE AMO DEMAIS MINHA LINDA!

A minha filhinha querida Laura, por iluminar minha vida com seu sorriso de amor e suas conversas ainda 'incompreensíveis'. Princesinha, você é uma bênção na vida do papai.

A minha família querida, aos meus pais, Claudio e Ana, pela educação recebida através do amor que me deram e por sempre me incentivarem a fazer o meu melhor, e meus irmãos, Gabriel e João Pedro, pela amizade e amor que sempre nos une e por torcerem por mim.

Ao meu orientador, Prof. Arinei Carlos Lindbeck da Silva, pelos ensinamentos preciosos que superam os conteúdos das disciplinas, pela atenção concedida nas muitas conversas, pela compreensão frente aos problemas que surgiram no decorrer do caminho e pela amizade ao me receber nas muitas sextas-feiras pela manhã para orientação.

Aos professores Lauro e Volmir por contribuir no desenvolvimento desta dissertação com comentários e sugestões que enriqueceram muito o texto apresentado.

A secretária Maristela Bandil, por ser esta pessoa espetacular capaz de resolver todos os nossos problemas de alunos do PPGMNE num piscar de olhos, e sempre de bom humor.

Aos meus professores, Anselmo Chaves Neto, Liliana Madalena Gramani, Maria Teresinha Arns Steiner, Neida Maria Patias Volpi, e em especial ao Prof. Celso Carnieri por me encantar com as belezas da Pesquisa Operacional, em especial a Programação Linear.

Aos amigos Tomio, Maria e Cadu, pelo auxílio nas correções ortográficas e por engrandecer o conteúdo deste trabalho.

Ao Colégio Militar de Curitiba e ao IFSC Campus Joinville por me disponibilizar tempo para frequentar as aulas e orientações na UFPR.

Aos colegas de curso, Cassius, Gustavo, Francisco, Ângela e em especial ao amigo Paulo Cesar que me incentivou a dar início ao curso.

Aos colegas de trabalho e amigos Penna, Roberto, Marcus Vinícius, Roberto, Rodrigo, Valcides e Guilherme, do CMC, e Leandro, Fabio e Cristiano, do IFSC, que sempre trouxeram uma palavra de apoio em meio a correria do nosso cotidiano escolar.

Aos meus alunos que sempre me incentivaram a ser o melhor profissional que eu puder a me fizeram buscar a cada dia me qualificar mais para o trabalho. Que eu possa ser um exemplo de dedicação e estudo a eles, sempre.

E a todas as demais pessoas que de maneira direta ou indireta contribuíram nestes últimos anos para a realização deste trabalho.

Tudo posso naquele que me fortalece.

Paulo (*Fp.* 4:3)

RESUMO

Esta dissertação aborda uma proposta de metodologia de resolução de um problema de entregas que abrange a integração de dois problemas clássicos de Otimização Combinatória: o Problema de Carregamento de Container (PCC) e o Problema de Roteamento de Veículos (PRV). O problema específico analisado está na logística empregada no carregamento e entrega de eletrodomésticos (linha branca) vendidos à pessoa física. Para representar esta situação, assume-se um cenário fictício em que a empresa que vende os produtos possui um Centro de Distribuição de Produtos (CD) localizado na cidade de Curitiba e uma lista de doze possíveis produtos a serem vendidos. A partir desta lista foram gerados 160 pedidos diferentes para serem entregues em vinte endereços aleatórios localizados também na cidade de Curitiba. Para a resolução deste problema, apresenta-se uma metodologia baseada em formação de torres de caixas e um Algoritmo *Bottom-Left* para realizar o carregamento dos pedidos no compartimento de carga dos veículos e um Algoritmo Genético para realizar a otimização evolutiva da solução até que se encontre uma solução suficientemente próxima à solução ótima do problema, buscando diminuir, a cada geração, a distância total percorrida pelos veículos de entrega. Para demonstração e utilização desta metodologia, apresenta-se uma implementação dos algoritmos e técnicas de pesquisa operacional descritos acima para a resolução desenvolvida em linguagem de programação *Microsoft Visual Basic*. Utilizando-se esta implementação e o cenário construído para testes, obteve-se bons resultados em relação à distância total percorrida pelos veículos de entrega, com redução de 25% a 45% em relação às soluções iniciais aleatórias, sendo que em alguns casos, esta melhoria alcançou até 60%.

Palavras-Chave: Problema de Carregamento de Container. Problema de Roteamento de Veículos. Algoritmo *Bottom-Left*. Algoritmo Genético.

ABSTRACT

This work discusses a proposal of methodology of solving a problem of deliveries includes the integration of two classic problems of combinatorial optimization: the Container Loading Problem (CLP) and the Vehicle Routing Problem (VRP). The specific problem is analyzed in the logistics used in loading and delivery of household appliances (white goods) sold to individuals. To represent this situation, it assumes a fictional scenario in which the company selling the product has a Product Distribution Center (DC) located in the city of Curitiba and a list of twelve possible products to be sold. From this list 160 different orders were generated to be delivered in twenty random addresses also located within the city of Curitiba. To solve this problem, it presents a methodology based on the formation of the towers of boxes and a Bottom-Left Algorithm to perform the loading of applications in the cargo compartment of vehicles and a Genetic Algorithm to perform the evolutionary optimization of the solution until it is a solution close enough to the optimal solution of the problem, seeking to diminish with each generation the total distance traveled by delivery vehicles. For demonstration and use of this methodology, it presents an implementation of algorithms and operations research techniques described above in relation to the solve, developed in Microsoft Visual Basic programming language. Using this implementation and the scenario built for testing, we achieved good results in relation to the total distance traveled by delivery vehicles, with average improvement of 25% to 45% compared to random initial solutions, and in some cases, this improvement reached up to 60%.

Keywords: Container Loading Problem. Vehicle Routing Problem. Bottom-Left Algorithm. Genetic Algorithm.

LISTA DE ILUSTRAÇÕES

FIGURA 3.1	–	CONTAINER COM DIMENSÕES $L \times W \times H$ E CAIXA COM DIMENSÕES $ln \times wn \times hn$24
FIGURA 3.2	–	EXEMPLO DE UM CONTAINER COM A ORIGEM LOCALIZADA NO CANTO INFERIOR FRONTAL ESQUERDO.....24
FIGURA 3.3	–	EXEMPLO DE POSICIONAMENTO DE UMA CAIXA DO TIPO n DENTRO DO CONTAINER, BASEADO EM JUNQUEIRA (2010)26
FIGURA 3.4	–	CONTAINER CARREGADO COM CAMADAS DE CAIXAS COM DIMENSÕES $W \times H \times d$28
FIGURA 3.5	–	REPRESENTAÇÃO DAS SEIS ORIENTAÇÕES QUE UMA CAIXA PODE ASSUMIR29
FIGURA 3.6	–	REPRESENTAÇÃO DA PRIMEIRA CAMADA VERTICAL DE UM CONTAINER31
FIGURA 3.7	–	EXEMPLO DE PILHA (TORRE) DE CAIXAS32
FIGURA 3.8	–	EXEMPLOS DE PADRÕES DE CORTE TRIDIMENSIONAL GUILHOTINADO (A) E NÃO GUILHOTINADO (B).....33
FIGURA 3.9	–	EXEMPLO DE CUBÓIDE $ctki = 2,3,3$33
FIGURA 3.10	–	EXEMPLO DE ROTEAMENTO DE VEÍCULO A PARTIR DE UM CENTRO DE DISTRIBUIÇÃO (CD).....34
FIGURA 3.11	–	CARREGAMENTO FACTÍVEL PELA PROPOSTA DE GENDRAU <i>ET AL.</i> (2006)38
FIGURA 3.12	–	REPRESENTAÇÃO DE UM CROMOSSOMO EM AG'S39
FIGURA 3.13	–	REPRESENTAÇÃO DE UMA OPERAÇÃO DE CRUZAMENTO40
FIGURA 3.14	–	REPRESENTAÇÃO DE UMA OPERAÇÃO DE MUTAÇÃO.....41
FIGURA 3.15	–	CORTE BIDIMENSIONAL COM O ABL.43
FIGURA 4.1	–	RECONDICIONAMENTO DE PEDIDO QUE EXTRAPOLA AS DIMENSÕES DO CONTAINER.....49
FIGURA 5.1	–	INTERFACE INICIAL DO SOFTWARE PARA CADASTRAR PRODUTOS E PEDIDOS.....54

FIGURA 5.2	–	INTERFACE PARA CADASTRO DAS INFORMAÇÕES DE CADA PRODUTO.....	55
FIGURA 5.3	–	INTERFACE PARA CARREGAR O ARQUIVO DE UMA AMOSTRA A PARTIR DA BASE DE DADOS.	56
FIGURA 5.4	–	INTERFACE PARA CADASTRO DAS INFORMAÇÕES DE UM PEDIDO.....	57
FIGURA 5.5	–	ARQUIVO DE PRODUTOS SALVO NA BASE DE DADOS.....	57
FIGURA 5.6	–	EXEMPLO DE ARQUIVO DE UM PEDIDO SALVO NA BASE DE DADOS.....	58
FIGURA 5.7	–	INTERFACE DO SOFTWARE DE RESOLUÇÃO DE PROBLEMA DE ENTREGAS.....	59
FIGURA 5.8	–	PLANILHA COM INFORMAÇÕES DOS PEDIDOS SELECIONADOS.	60
FIGURA 5.9	–	JANELA COM INFORMAÇÕES DA RESOLUÇÃO DO PROBLEMA.....	61
FIGURA 5.10	–	ARQUIVO COM RESPOSTA SIMPLES DO PROBLEMA DE ENTREGAS.....	61
FIGURA 5.11	–	ARQUIVO COM INFORMAÇÕES COMPLETAS SOBRE A RESOLUÇÃO DO PROBLEMA DE ENTREGAS.	62
FIGURA 5.12	–	INTERFACE DO SOFTWARE PARA CONSTRUÇÃO DAS RESPOSTAS.....	63
FIGURA 5.13	–	VISUALIZAÇÃO DAS TORRES DE UM DETERMINADO PEDIDO.....	64
FIGURA 5.14	–	EXEMPLO DE ARQUIVO COM AS INFORMAÇÕES PARA OS RESPONSÁVEIS PELO ESTOQUE E ORGANIZAÇÃO DE MERCADORIAS.....	65
FIGURA 5.15	–	EXEMPLO DE ARQUIVOS COM AS INFORMAÇÕES ENVIADAS AOS RESPONSÁVEIS POR CARREGAR OS VEÍCULOS.	66
FIGURA 5.16	–	EXEMPLO DE MAPA COM O ROTEAMENTO DE ENTREGA, ENVIADO AO MOTORISTA DO VEÍCULO RESPONSÁVEL POR ESTAS ENTREGAS.	67

FIGURA 5.17	–	EXEMPLO DE ARQUIVO FORNECIDO AOS MOTORISTAS CONTENDO AS INFORMAÇÕES DOS PEDIDOS TRANSPORTADOS POR SEU VEÍCULO.....	68
FIGURA 5.18	–	EXEMPLO DE ARQUIVO SÍNTESE CONTENDO AS INFORMAÇÕES PARA O SETOR DE CONTROLADORIA/AUDITORIA DA EMPRESA.....	69

LISTA DE QUADROS

QUADRO 2.1	– LISTA DE ELETRODOMÉSTICOS OFERECIDOS PELA EMPRESA FICTÍCIA	20
QUADRO 2.2	– ENDEREÇOS SELECIONADOS ALEATORIAMENTE PARA ENTREGA DOS PEDIDOS	21
QUADRO 2.3	– EXEMPLO DE PEDIDO GERADO NO CENÁRIO FICTÍCIO	21
QUADRO 4.1	– ALGORITMO DE FORMAÇÃO DE TORRES - AFT.....	47
QUADRO 4.2	– PSEUDO-CÓDIGO PARA O ABL UTILIZADO.....	49
QUADRO 4.3	– TAMANHO DA POPULAÇÃO EM FUNÇÃO DO NÚMERO DE PEDIDOS.	50
QUADRO 6.1	– RESUMO DO COMPORTAMENTO DA DISTÂNCIA TOTAL PERCORRIDA PELA MELHOR SOLUÇÃO A CADA GERAÇÃO, EM 250 GERAÇÕES, PARA UMA AMOSTRA ALEATÓRIA DE CADA CENÁRIO ESCOLHIDO.	70
QUADRO 6.2	– RESUMO DO COMPORTAMENTO DA DISTÂNCIA TOTAL PERCORRIDA PELA ELITE DA POPULAÇÃO (50% DAS MELHORES RESPOSTAS DA POPULAÇÃO) A CADA GERAÇÃO, EM 250 GERAÇÕES, PARA UMA AMOSTRA ALEATÓRIA DE CADA CENÁRIO ESCOLHIDO.	71

LISTA DE GRÁFICOS

GRÁFICO 6.1	–	COMPORTAMENTO DAS MELHORES SOLUÇÕES EM 250 GERAÇÕES PARA CINCO AMOSTRAS ESCOLHIDAS ALEATORIAMENTE, UMA PARA CADA CENÁRIO ESCOLHIDO.....	71
GRÁFICO 6.2	–	COMPORTAMENTO DAS SOLUÇÕES DA ELITE DA POPULAÇÃO EM 250 GERAÇÕES PARA CINCO AMOSTRAS ESCOLHIDAS ALEATORIAMENTE, UMA PARA CADA CENÁRIO ESCOLHIDO.	72
GRÁFICO 6.3	–	COMPORTAMENTO PERCENTUAL DAS MELHORES SOLUÇÕES EM 250 GERAÇÕES PARA CINCO AMOSTRAS ESCOLHIDAS ALEATORIAMENTE, UMA PARA CADA CENÁRIO ESCOLHIDO.	73
GRÁFICO 6.4	–	COMPORTAMENTO PERCENTUAL DAS SOLUÇÕES DA ELITE DA POPULAÇÃO EM 250 GERAÇÕES PARA CINCO AMOSTRAS ESCOLHIDAS ALEATORIAMENTE, UMA PARA CADA CENÁRIO ESCOLHIDO.	74
GRÁFICO 6.5	–	COMPARAÇÃO ENTRE A MELHOR SOLUÇÃO INICIAL E A MELHOR SOLUÇÃO FINAL, EM MÉDIA, PARA CADA CENÁRIO ESCOLHIDO ENTRE 50 E 350 PEDIDOS.....	75
GRÁFICO 6.6	–	MELHORIA PERCENTUAL MÉDIA PARA CADA CENÁRIO ESCOLHIDO ENTRE 50 E 350 PEDIDOS.	76
GRÁFICO 6.7	–	COMPARAÇÃO ENTRE A QUANTIDADE DE VEÍCULOS NECESSÁRIA PARA A MELHOR SOLUÇÃO INICIAL E FINAL, EM MÉDIA, PARA CADA CENÁRIO ESCOLHIDO ENTRE 50 E 350 PEDIDOS.....	77
GRÁFICO 6.8	–	COMPARAÇÃO ENTRE A ÁREA OCUPADA PELAS TORRES DE CAIXAS NO CONTAINER PARA A MELHOR SOLUÇÃO INICIAL E FINAL, EM MÉDIA, PARA CADA CENÁRIO ESCOLHIDO ENTRE 50 E 350 PEDIDOS.	77

LISTA DE SIGLAS

ABL	-	Algoritmo <i>Bottom-Left</i>
AFT	-	Algoritmo de Formação de Torres
AG	-	Algoritmo Genético
CD	-	Centro de Distribuição de Produtos
GRASP	-	<i>Greedy Randomized Adaptive Search Procedure</i>
GPS	-	<i>Global Position System</i>
OG	-	Operador Genético
PCC	-	Problema de Carregamento de Container
PRV	-	Problema de Roteamento de Veículos

SUMÁRIO

1 INTRODUÇÃO	16
1.1 OBJETIVOS DO TRABALHO	16
1.1.1 Objetivos Específicos.....	16
1.2 METODOLOGIA	17
1.3 LIMITAÇÕES	17
1.4 ESTRUTURA DO TRABALHO	18
2 DESCRIÇÃO DO PROBLEMA	19
2.1 CENÁRIO FICTÍCIO PARA O PROBLEMA ESTUDADO	20
3 REFERENCIAL TEÓRICO	23
3.1 O PROBLEMA DE CARREGAMENTO DE CONTAINER (PCC).....	23
3.1.1 Formulação Matemática do PCC.....	23
3.1.2 Diferentes abordagens para o PCC.....	27
3.1.3 Restrições práticas para o PCC.....	28
3.1.4 Padrão de carregamento	31
3.2 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)	34
3.3 ALGORITMO GENÉTICO (AG)	38
3.3.1 Indivíduos	39
3.3.2 População.....	39
3.3.3 Operadores Genéticos.....	40
3.3.3.1 Operador Genético de Reprodução.....	40
3.3.3.2 Operador Genético de Cruzamento.....	40
3.3.3.3 Operador Genético de Mutação	41
3.3.4 Funcionamento Básico de um AG	41
3.4 ALGORITMO <i>BOTTOM-LEFT</i> (ABL)	42
4 METODOLOGIA	45
4.1 RECEBIMENTO DOS PEDIDOS.....	45
4.2 SOLUÇÕES INICIAIS	48
4.3 APLICAÇÃO DO AG E EVOLUÇÃO DAS SOLUÇÕES.....	50
4.4 CONSTRUÇÃO DAS INFORMAÇÕES ENVIADAS AOS SETORES.....	51
5 IMPLEMENTAÇÃO	53
5.1 CRIAÇÃO DE CENÁRIO FICTÍCIO	53

5.2 SELEÇÃO DOS PEDIDOS E RESOLUÇÃO DO PROBLEMA	58
5.3 APRESENTAÇÃO DAS TORRES DE UM DETERMINADO PEDIDO E CONSTRUÇÃO DAS INFORMAÇÕES ESPECÍFICAS POR SETOR	63
6 RESULTADOS COMPUTACIONAIS	70
7 CONCLUSÕES.....	78
7.1 SUGESTÕES PARA TRABALHOS FUTUROS	80
REFERÊNCIAS.....	81

1 INTRODUÇÃO

Na área da logística, um problema está presente no cotidiano de empresas que necessitam entregar produtos ou encomendas a Pessoas Físicas, que envolve desde a organização dos produtos que deverão ser entregues ao carregamento destes produtos em veículos de carga e, também, o roteamento destes veículos para realizarem as entregas. Este problema apresenta uma aplicação da integração de dois problemas clássicos da Otimização Combinatória: o Problema de Carregamento de Container (PCC) e o Problema de Roteamento de Veículos (PRV). Nesta dissertação, este problema será, resumidamente, chamado de Problema de Entregas.

Ambos os problemas são considerados de complexidade não-polinomial (Oliveira, 2009 e Bazgan *et al.* 2005), ou seja, complexos de serem resolvidos de utilizando métodos exatos, como programação linear. Conseqüentemente, a integração dos dois problemas apresenta uma complexidade ainda maior do que resolver cada um em separado, já que pequenas alterações no carregamento dos produtos podem gerar grandes alterações no roteamento dos veículos. Decorrente desta complexidade, segundo Guimarães (2011), problemas integrados de carregamento e roteamento são uma das aplicações mais frutíferas da área de Programação Matemática, mais especificamente da Pesquisa Operacional, pois respostas cada vez mais precisas serão revertidas na diminuição dos custos envolvidos com o transporte de mercadorias.

1.1 OBJETIVOS DO TRABALHO

O Objetivo Geral da presente dissertação é apresentar uma proposta de resolução heurística para o problema de entregas, que envolve a integração do Problema de Carregamento de Container com o Problema de Roteamento de Veículos.

1.1.1 Objetivos Específicos

a) Apresentar uma metodologia eficiente para a resolução do Problema de Entregas;

b) Implementar um software de acordo com a proposta de resolução do problema integrado;

c) Avaliar a proposta de resolução do problema integrado através de testes com diferentes quantidades de pedidos recebidos pela empresa, mostrando que a proposta melhora significativamente a solução inicial adotada no problema;

d) Aprimorar a implementação que será realizada de modo que o software apresente as respostas para cada setor envolvido no Problema de Entregas.

1.2 METODOLOGIA

Neste trabalho, a aplicação proposta se constitui de uma empresa que vende produtos à Pessoa Física e necessita fazer a entrega de várias vendas partindo de um Centro de Distribuição de Produtos (CD) da empresa. Para representar esta situação foi montado um Cenário Fictício que será apresentado adiante na dissertação.

A proposta apresentada para a resolução do problema de entregas utiliza a otimização evolutiva através de Algoritmos Genéticos (AG's). Para carregar os produtos de cada pedido, segundo a sequência de carregamento gerada pelo AG, utiliza-se o Algoritmo *Bottom-Left* para encontrar o padrão de carregamento dos produtos.

1.3 LIMITAÇÕES

No Cenário apresentado nesta dissertação serão desconsideradas as restrições referentes à condição de tráfego urbano. Atualmente, na cidade de Curitiba, durante alguns intervalos de tempo é proibida a circulação de veículos grandes por determinadas vias públicas que apresentam normalmente uma grande quantidade de veículos trafegando. Esta restrição poderia impossibilitar os veículos de entrega de seguirem um roteiro pré-determinado. Com o objetivo de simplificar a resolução do problema, esta restrição será ignorada durante a elaboração da proposta de resolução.

1.4 ESTRUTURA DO TRABALHO

A presente dissertação está estruturada em sete capítulos. No primeiro capítulo, introduz-se o tema da dissertação assim como os seus objetivos.

O segundo capítulo faz uma descrição mais detalhada do Problema de Entregas e da aplicação que se deseja trabalhar além de apresentar o cenário fictício que servirá como base para a aplicação proposta.

O terceiro capítulo traz uma descrição da teoria envolvida no Problema de Entregas, apresentando detalhes sobre o Problema de Carregamento de Container e seus desdobramentos, sobre o Problema de Roteamento de Veículos e sua integração com o PCC, sobre o Algoritmo Genético (que servirá como ferramenta de otimização) e sobre o Algoritmo *Bottom-Left* (que servirá para carregar o container).

O quarto capítulo descreve a metodologia proposta para a resolução do problema, desde o recebimento do pedido até a organização e carregamento dos produtos no compartimento de carga do veículo chegando ao roteamento deste veículo para realizar as entregas. Apresenta ainda a utilização dos algoritmos propostos e a metodologia de otimização das soluções e, finalmente, descreve as respostas que serão apresentadas aos diferentes setores envolvidos.

O quinto capítulo apresenta a implementação da metodologia descrita no quarto capítulo apresentando o software desenvolvido para a resolução do Problema de Entregas.

O sexto capítulo expõe uma síntese dos resultados obtidos de diversos testes realizados com software apresentado no quinto capítulo.

Para finalizar, no sétimo capítulo apresentam-se as conclusões obtidas pelo autor e algumas sugestões para pesquisas futuras.

2 DESCRIÇÃO DO PROBLEMA

Esta dissertação trata da integração de dois problemas clássicos da otimização combinatória: o Problema de Carregamento de Container e o Problema de Roteamento de Veículos. A aplicação escolhida é a entrega de produtos eletrodomésticos (linha branca) a clientes Pessoa Física.

Nesta aplicação, detecta-se claramente o Problema de Carregamento de Container presente no momento de carregar os produtos de cada um dos pedidos nos veículos que farão as entregas no Centro de Distribuição. As possibilidades na ordem de carregamento de um determinado número de pedidos podem ser descritas através da permutação dos pedidos (p pedidos $\rightarrow p!$ possibilidades de carregamento). Se, além disso, assumir que cada pedido possui em média n produtos (nesta dissertação cada produto será representado por uma caixa com largura, profundidade, altura, peso, pressão e resistência pré-determinados), então conclui-se que o número de possibilidades de carregamento destes produtos, assumindo que os produtos de um mesmo pedido devem ser agrupados, é $n!^p \cdot p!$. Este número elevado de possibilidades justifica a utilização de métodos heurísticos na busca da melhor sequência de carregamento.

Esta escolha da sequência de carregamento interfere diretamente na sequência de entrega dos pedidos, o que caracteriza o Problema de Roteamento de Veículos. Segundo Belfiore (2006) pode-se elencar uma série de variáveis que poderiam ser estudadas, por exemplo:

- a) o tempo gasto para realizar as entregas;
- b) o consumo de combustível dos veículos;
- c) a quantidade de veículos utilizada;
- d) a distância percorrida pelos veículos;
- e) condições de tráfego Urbano e Rural.

Neste trabalho foi escolhida como variável de decisão a distância percorrida por todos os veículos que farão as entregas. Reduzindo-se a distância percorrida por todos os veículos conseqüentemente ocorre a redução do tempo gasto para realizar as entregas e também a economia no consumo de combustível. Com a otimização do carregamento dos caminhões, mesmo que indiretamente a tendência é reduzir a quantidade de veículos utilizados.

Para o desenvolvimento do trabalho, implementar a metodologia e realizar testes computacionais, criou-se um cenário fictício que será tratado a seguir.

2.1 CENÁRIO FICTÍCIO PARA O PROBLEMA ESTUDADO

O cenário fictício criado envolve uma empresa que vende doze eletrodomésticos diferentes da linha branca, conforme Quadro 2.1 abaixo.

Seq.	Produto	Fabricante	Dimensões (cm)			Peso (kg)	Pressão (kg/cm ²)	Resist. (kg/cm ²)
			Larg	Prof	Alt			
1	Fogão 4 bocas Coliseum	Atlas	48	57	85	21	7,52E-03	1,50E-02
2	Fogão 4 bocas Ibiza Glass	Esmaltec	50	57	85	22	7,78E-03	1,56E-02
3	10kg Clean	Brastemp	67	71	105	43	9,07E-03	1,81E-02
4	10,4kg lava e seca	Samsung	65	79	94	88	1,72E-02	3,45E-02
5	439L BRM49	Brastemp	71	74	188	87	1,66E-02	1,25E-02
6	553L DF80	Electrolux	81	75	190	107	1,76E-02	1,32E-02
7	25L CMN34AB	Consul	57	47	37	18	6,84E-03	2,05E-02
8	28L NN-ST568	Panasonic	56	44	36	16	6,24E-03	1,87E-02
9	7500BTUs CCN07B	Consul	57	60	43	28	8,23E-03	1,65E-02
10	Multiclimatizador Q/F	Komeco	39	24	67	7	7,80E-03	1,56E-02
11	Forno 28L Super Cook	Mondial	56	72	44	6	1,49E-03	3,72E-03
12	Forno 23L	Suggar	28	35	46	6	5,82E-03	1,45E-02

QUADRO 2.1 – LISTA DE ELETRODOMÉSTICOS OFERECIDOS PELA EMPRESA FICTÍCIA
Fonte: O Autor

Esta empresa possui um Centro de Distribuição onde serão carregados todos os veículos e, conseqüentemente, de onde sairão todas as entregas de pedidos recebidos dos clientes. Este Centro de Distribuição está situado à Rua Celestino Menssing de Siqueira, 411, na Cidade Industrial de Curitiba, Curitiba/PR. Neste Centro de Distribuição não existe um limite de veículos a serem utilizados nas entregas. Todos os veículos de entrega desta empresa fictícia são caminhões do tipo 'Baú' com as seguintes dimensões: 250cm de largura, 600cm de profundidade e 265cm de altura, além de que não se leva em consideração o limite de peso que pode ser carregado no veículo.

Além deste Centro de Distribuição e destas características da empresa fictícia que fará as entregas dos produtos, foram escolhidos vinte endereços aleatórios na cidade de Curitiba, conforme Quadro 2.2 abaixo, para onde serão encaminhados os pedidos. Cada pedido recebido pela empresa possui um endereço de entrega entre estes vinte endereços listados.

Seq.	Logradouro	Número	Cidade/UF
1	Alameda Augusto Stellfeld	1200	Curitiba/PR
2	Avenida Getúlio Vargas	3225	Curitiba/PR
3	Avenida Manoel Ribas	480	Curitiba/PR
4	Estrada do Santa Cândida	1200	Curitiba/PR
5	Praça Conselheiro Thomás Coelho	1	Curitiba/PR
6	Rua Anne Frank	4501	Curitiba/PR
7	Rua Ari José Valle	660	Curitiba/PR
8	Rua Barão do Rio Branco	320	Curitiba/PR
9	Rua Coronel Joaquim Lacerda	140	Curitiba/PR
10	Rua Costa Rica	700	Curitiba/PR
11	Rua Engenheiro José Orontes Pires	134	Curitiba/PR
12	Rua Estados Unidos	212	Curitiba/PR
13	Rua Gastão Câmara	600	Curitiba/PR
14	Rua Hayton da Silva Pereira	134	Curitiba/PR
15	Rua João Bettega	2380	Curitiba/PR
16	Rua João Dembinski	1230	Curitiba/PR
17	Rua Lysimaco Ferreira da Costa	1000	Curitiba/PR
18	Rua Mato Grosso	30	Curitiba/PR
19	Rua Rosamélia de Oliveira	1000	Curitiba/PR
20	Rua Ubaldino do Amaral	412	Curitiba/PR

QUADRO 2.2 – ENDEREÇOS SELECIONADOS ALEATORIAMENTE PARA ENTREGA DOS PEDIDOS

Fonte: O Autor

Para cada endereço deste foram criados oito pedidos diferentes, contendo aleatoriamente de três a dez produtos dentre os doze mostrados no Quadro 2.2, totalizando 160 pedidos diferentes. Um dos pedidos gerados neste cenário fictício é demonstrado abaixo:

<p><i>Pedido 16 com 6 produtos</i></p> <p><i>Endereço: Rua Estados Unidos, 212 – Curitiba/PR</i></p> <p><i>Produtos:</i></p> <p><i>1 unidade do produto 5 (439L BRM49 - Brastemp)</i></p> <p><i>2 unidades do produto 6 (553L DF80 - Electrolux)</i></p> <p><i>1 unidade do produto 10 (Multiclimatizador Q/F - Komeco)</i></p> <p><i>1 unidade do produto 11 (Forno 28L Super Cook - Mondial)</i></p> <p><i>1 unidade do produto 12 (Forno 23L - Suggar)</i></p>

QUADRO 2.3 – EXEMPLO DE PEDIDO GERADO NO CENÁRIO FICTÍCIO

Fonte: O Autor

Baseado neste cenário, foram criadas instâncias de testes onde se escolhem aleatoriamente entre 50 e 350 pedidos e, então, resolve-se o problema de encontrar a melhor sequência para carregar todos estes pedidos, em quantos

veículos forem necessários, de forma a percorrer a menor distância possível e, conseqüentemente, utilizar a menor quantidade de veículos possível e também o melhor padrão de carregamento possível.

Este cenário, e também estas instâncias, foram criados a fim de ilustrar de maneira mais clara a aplicação específica do problema que se deseja estudar e, assim, tornar o problema mais concreto e mais fácil de ser visualizado e compreendido.

3 REFERENCIAL TEÓRICO

Este capítulo apresenta os problemas que foram abordados nesta dissertação, bem como métodos heurísticos que serão utilizados na resolução dos problemas. Inicialmente, descreve-se o Problema de Carregamento de Container e seus desdobramentos, em seguida o Problema de Roteamento de Veículos e seus desdobramentos. Então, discutem-se algumas considerações sobre a integração dos dois problemas. Além destes problemas, descreve-se também, o funcionamento de um Algoritmo Genético e de um Algoritmo *Bottom-Left*.

3.1 O PROBLEMA DE CARREGAMENTO DE CONTAINER (PCC)

O PCC é um problema muito discutido nas últimas décadas e foi tema de diversas publicações como George & Robinson (1980), Morabito & Arenales (1994;1997), Bischoff & Ratcliff (1995), Gehring & Bortfeldt (1997), Eley (2002), Pisinger (2002), Cecílio & Morabito (2004), Araújo (2006), Junqueira, Morabito & Yamashita (2010), Guimarães (2011). É um problema clássico de Otimização Combinatória e possui uma abordagem semelhante ao Problema da Mochila. Genericamente, pode ser descrito como o problema de posicionar caixas, em formato de prismas retangulares retos de diferentes dimensões, em um compartimento (container) que também é um prisma retangular reto, substancialmente maior, buscando maximizar a ocupação do espaço do container.

3.1.1 Formulação Matemática do PCC

Junqueira, Morabito & Yamashita (2010) propõem uma formulação matemática interessante para o PCC, semelhante à formulação abaixo.

Seja um container de profundidade L , largura W e altura H que deve ser carregado com caixas de n tipos diferentes. Cada tipo de caixa possui uma profundidade l_n , largura w_n , altura h_n , quantidade b_n e valor v_n , com $n = 1, \dots, m$, que pode ser representado pelo volume da caixa se $v_n = (l_n \cdot w_n \cdot h_n)$. A figura 3.1 apresenta uma representação do container e de uma caixa do tipo n .

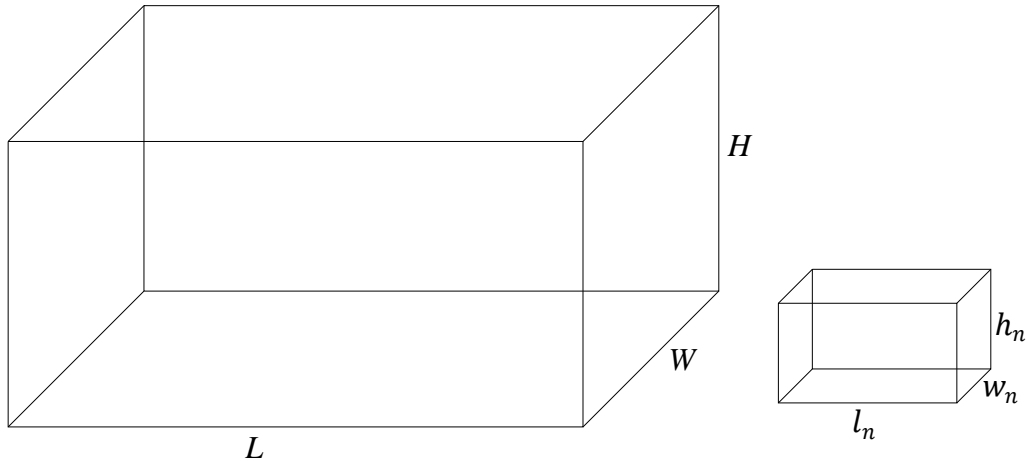


FIGURA 3.1 – CONTAINER COM DIMENSÕES $L \times W \times H$ E CAIXA COM DIMENSÕES $l_n \times w_n \times h_n$
 Fonte: O autor

Admite-se que as dimensões são inteiras, que as caixas só podem ser empacotadas ortogonalmente e que as suas orientações são fixas (esta última restrição pode ser relaxada trabalhando com um modelo semelhante a este, mas é mantida pela simplicidade de apresentação das formulações).

Adotando-se um sistema de coordenadas cartesianas com origem no canto inferior frontal esquerdo do container a tripla (p, q, r) representará a posição onde o canto inferior frontal esquerdo de uma determinada caixa é colocado (Figura 3.2).

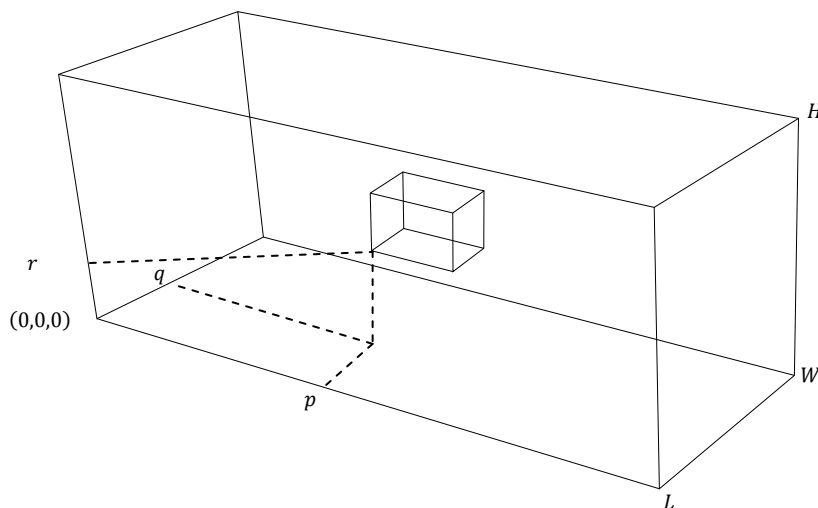


FIGURA 3.2 – EXEMPLO DE UM CONTAINER COM A ORIGEM LOCALIZADA NO CANTO INFERIOR FRONTAL ESQUERDO
 Fonte: O autor

Logo, as possíveis posições, ao longo do comprimento L , da largura W e da altura H do container, onde cada caixa pode ser colocada, podem ser definidas por meio dos seguintes conjuntos:

$$X = \left\{ p \mid 0 \leq p \leq L - \min_n(l_n) \text{ e } p \in \mathbb{Z}, n = 1, \dots, m \right\} \quad (1)$$

$$Y = \left\{ q \mid 0 \leq q \leq W - \min_n(w_n) \text{ e } q \in \mathbb{Z}, n = 1, \dots, m \right\} \quad (2)$$

$$Z = \left\{ r \mid 0 \leq r \leq H - \min_n(h_n) \text{ e } r \in \mathbb{Z}, n = 1, \dots, m \right\} \quad (3)$$

Foi baseado no trabalho de Christofides & Whitlock (1977) que Junqueira, Morabito & Yamashita (2010) reduziram os conjuntos X , Y e Z para os conjuntos abaixo:

$$X = \left\{ p \mid p = \sum_{n=1}^m \beta_n \cdot l_n, 0 \leq p \leq L - \min_n(l_n), 0 \leq \beta_n \leq b_n \text{ e } p \in \mathbb{Z}, n = 1, \dots, m \right\} \quad (4)$$

$$Y = \left\{ q \mid q = \sum_{n=1}^m \beta_n \cdot w_n, 0 \leq q \leq W - \min_n(w_n), 0 \leq \beta_n \leq b_n \text{ e } q \in \mathbb{Z}, n = 1, \dots, m \right\} \quad (5)$$

$$Z = \left\{ r \mid r = \sum_{n=1}^m \beta_n \cdot h_n, 0 \leq r \leq H - \min_n(h_n), 0 \leq \beta_n \leq b_n \text{ e } r \in \mathbb{Z}, n = 1, \dots, m \right\} \quad (6)$$

De modo a descrever a não sobreposição de caixas dentro do container definiu-se a função $a_{npqrstu}$, com $n = 1, \dots, m$, além de $p, s \in X$ e $q, t \in Y$ e $r, u \in Z$, que possui um resultado binário definido por

$$a_{npqrstu} = \begin{cases} 1, & \text{se uma caixa do tipo } n, \text{ quando empacotada com seu canto} \\ & \text{inferior frontal esquerdo na posição } (p, q, r) \text{ não permite} \\ & \text{que outra caixa qualquer ocupe a posição } (s, t, u); \\ 0, & \text{caso contrário.} \end{cases} \quad (7)$$

porém, nota-se que a função $a_{npqrstu}$ não se apresenta como uma variável de decisão do modelo podendo ser computada *a priori* da seguinte maneira:

$$a_{npqrstu} = \begin{cases} 1, & \text{se } 0 \leq p \leq s \leq p + l_n - 1 \leq L - 1 \\ & \text{e } 0 \leq q \leq t \leq q + w_n - 1 \leq W - 1 \\ & \text{e } 0 \leq r \leq u \leq r + h_n - 1 \leq H - 1; \\ 0, & \text{caso contrário.} \end{cases} \quad (8)$$

ou seja, se $a_{npqrstu} = 1$, significa que a posição (s, t, u) está ocupada por uma caixa do tipo n colocada com seu canto inferior frontal esquerdo na posição (p, q, r) do container (Figura 3.3). Caso contrário, se $a_{npqrstu} = 0$, significa que esta posição não está ocupada pela caixa.

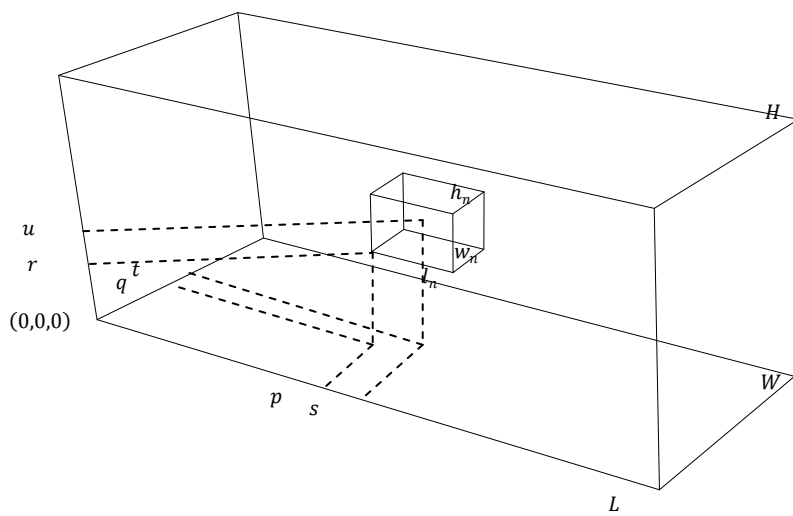


FIGURA 3.3 – EXEMPLO DE POSICIONAMENTO DE UMA CAIXA DO TIPO n DENTRO DO CONTAINER, BASEADO EM JUNQUEIRA (2010)
Fonte: O autor

Já as variáveis de decisão do modelo (x_{npqr}) são definidas como:

$$x_{npqr} = \begin{cases} 1, & \text{se uma caixa do tipo } n \text{ é empacotada com seu canto} \\ & \text{inferior frontal esquerdo na posição } (p, q, r) \text{ tal que} \\ & 0 \leq p \leq L - l_n, 0 \leq q \leq W - w_n \text{ e } 0 \leq r \leq H - h_n; \\ 0, & \text{caso contrário.} \end{cases} \quad (9)$$

Sejam ainda:

$$X_n = \{p \in X | 0 \leq p \leq L - l_n, n = 1, \dots, m\} \quad (10)$$

$$Y_n = \{q \in Y | 0 \leq q \leq W - w_n, n = 1, \dots, m\} \quad (11)$$

$$Z_n = \{r \in Z | 0 \leq r \leq H - h_n, n = 1, \dots, m\} \quad (12)$$

Então o problema de carregamento de caixas dentro de um único container pode ser formulado como uma extensão direta do modelo de programação linear inteira 0-1 proposto em Beasley (1985) para o problema de corte bidimensional não guilhotinado restrito:

$$\max \sum_{n=1}^m \sum_{p \in X_n} \sum_{q \in Y_n} \sum_{r \in Z_n} v_n \cdot x_{npqr} \quad (13)$$

$$\text{Sujeito a: } \begin{cases} \sum_{n=1}^m \sum_{p \in X_n} \sum_{q \in Y_n} \sum_{r \in Z_n} a_{npqrstu} \cdot x_{npqr} \leq 1 & s \in X, t \in Y, u \in Z & (14) \\ \sum_{p \in X_n} \sum_{q \in Y_n} \sum_{r \in Z_n} x_{npqr} \leq b_n & n = 1, \dots, m & (15) \\ x_{npqr} \in \{0,1\} & n = 1, \dots, m \\ & p \in X_n, q \in Y_n, r \in Z_n & (16) \end{cases}$$

Nesta formulação a função objetivo visa maximizar o valor total das caixas empacotadas dentro do container, enquanto as restrições impedem que haja sobreposição de caixas, que sejam carregadas mais caixas do que as quantidades disponíveis de cada tipo e definem o domínio das variáveis de decisão x_{npqr} .

3.1.2 Diferentes abordagens para o PCC

Segundo Pisinger (2002) este problema pode ser dividido em quatro modelos de abordagem distintos:

a) *Strip Packing Problem*, onde o container é considerado com duas dimensões fixas (largura e altura) e uma dimensão livre (profundidade). Nesta abordagem opta-se por deixar a profundidade do container livre, e o objetivo é carregar as caixas de forma a minimizar a profundidade utilizada do container, mesmo sem conhecer esta dimensão do container, ou seja, objetiva-se 'compactar' o carregamento das caixas no fundo do container. O trabalho de George & Robinson (1980) apresenta um algoritmo de construção de paredes ou camadas de caixas que exemplifica esta abordagem (Figura 3.4).

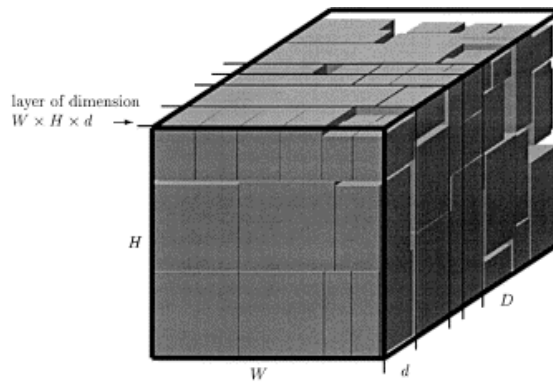


FIGURA 3.4 – CONTAINER CARREGADO COM CAMADAS DE CAIXAS COM DIMENSÕES $W \times H \times d$
 Fonte: Pisinger (2002)

b) *Knapsack Loading Problem*, onde se tem um único container que possui as três dimensões fixas. Esta abordagem consiste em posicionar as caixas dentro de um único container de maneira a carregar a maior quantidade possível de caixas dentro deste container, ou seja, maximizar a quantidade de caixas carregadas pelo container.

c) *Bin Packing Problem*, onde se deve carregar um conjunto de caixas que não necessariamente cabe num único container e, para isso, tem-se uma quantidade indefinida de containeres iguais (de mesmas dimensões). Esta abordagem objetiva minimizar o número destes containeres utilizados e, para tanto, esta será a abordagem utilizada na presente dissertação.

d) *Multi-Container Loading Problem*, que apresenta uma abordagem parecida com o *Bin Packing Problem*, porém utiliza containeres com diferentes dimensões, ou seja, além de procurar o melhor posicionamento das caixas dentro do container possibilita escolher qual o melhor container para cada subconjunto de caixas.

3.1.3 Restrições práticas para o PCC

O trabalho de Bischoff & Ratcliff (1995) merece um destaque pois analisa diferentes técnicas de resolução do problema e mostra que cada diferente técnica considera um conjunto de particularidades que se adequam somente a uma parte de todas as considerações possíveis sobre o PCC. Neste trabalho os autores elencaram doze restrições de cunho prático que podem ser atendidas ou

descartadas durante a resolução do PCC de acordo com a característica específica do problema que se quer resolver. São elas:

Restrição 1 – Orientação: as caixas podem ser rotacionadas em sentidos específicos para facilitar o carregamento (Figura 3.5).

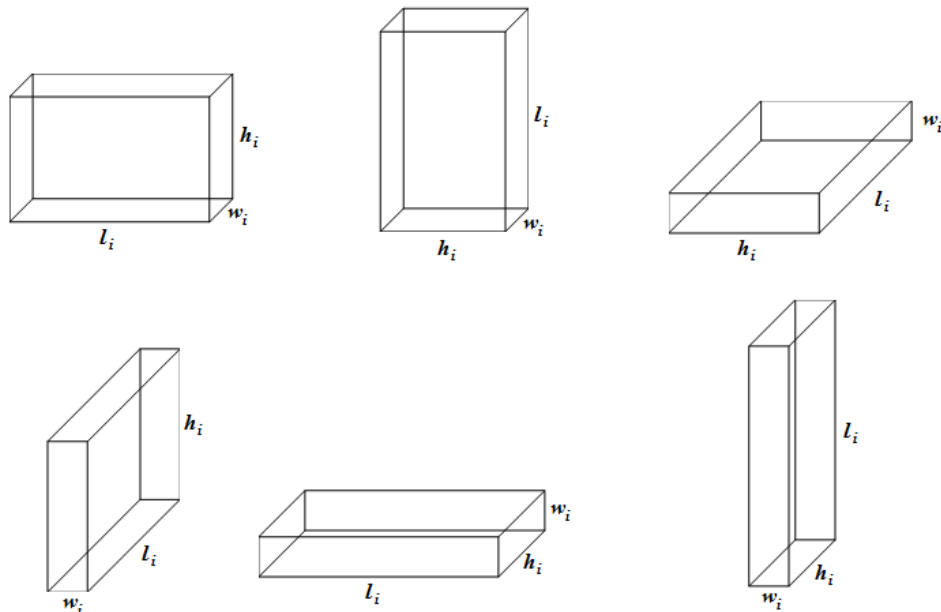


FIGURA 3.5 – REPRESENTAÇÃO DAS SEIS ORIENTAÇÕES QUE UMA CAIXA PODE ASSUMIR
Fonte: O Autor

Restrição 2 – Empilhamento: a caixa possui um limite de peso que pode ser empilhado sobre ela, ou ainda, a caixa possui um limite de pressão total exercida sobre a sua face superior, para que não ocorram deformações na caixa. Exemplos desta restrição aparecem em Scheithauer *et al.* (1996), Ratcliff & Bischoff (1998), Bortfeldt & Gehring (2001), Gehring & Bortfeldt (2002), Bischoff (2006), Gendrau *et al.* (2006) Lin *et al.* (2006), Christensen & Rousøe (2007).

Restrição 3 – Manuseio: algumas caixas, devido suas dimensões, seu peso ou ao equipamento que realiza a carga/descarga do container, devem estar posicionadas em lugares específicos do interior do container.

Restrição 4 – Estabilidade: as caixas que serão empilhadas a partir da segunda camada devem possuir uma porção das faces inferiores apoiadas sobre outras caixas de camadas anteriores ou sobre o piso do container, ou ainda, as faces laterais de uma caixa devem estar apoiadas nas faces laterais de outra caixa ou nas laterais do container. Exemplos desta restrição aparecem em Bischoff & Ratcliff (1995), Scheithauer *et al.* (1996), Davies & Bischoff (1999), Terno *et al.*

(2000), Bortfeldt & Gehring (2001), Eley (2002, 2003), Pisinger (2002), Bortfeldt *et al.* (2003), Silva & Soma (2003), Silva *et al.* (2003), Jin *et al.* (2004), Mack *et al.* (2004), Moura & Oliveira (2005), Yeung & Tang (2005), Gendrau *et al.* (2006) Lin *et al.* (2006), Araújo & Armentano (2007).

Restrição 5 – Agrupamento de Itens: as caixas com um destino comum ou de um mesmo tipo devem ser carregadas próximas dentro do container. Exemplos aparecem em Bischoff & Ratcliff (1995), Terno *et al.* (2000), Eley (2003).

Restrição 6 – Múltiplos Destinos: as caixas devem ser carregadas no container levando em consideração a ordem de entrega dos pedidos. Exemplos aparecem em Bischoff & Ratcliff (1995), Scheithauer *et al.* (1996), Terno *et al.* (2000), Jin *et al.* (2004), Moura & Oliveira (2005), Araújo (2006), Gendrau *et al.* (2006) Lin *et al.* (2006), Christensen & Rousøe (2007), Campos (2008).

Restrição 7 – Separação de Itens: caixas que não podem estar em contato umas com as outras devem ser carregadas separadas dentro do container (umidade, campo magnético, etc.). Um exemplo aparece em Eley (2003).

Restrição 8 – Carregamento Completo de Grupos de Itens: as caixas contendo todos os componentes que fazem parte de uma mesma entidade funcional (exemplo: caixas com as partes de um roupeiro), ou ainda caixas de um mesmo pedido, devem estar carregados no mesmo container. Um exemplo aparece em Terno *et al.* (2000).

Restrição 9 – Prioridades: pode-se levar em consideração a data de entrega ou prazo de validade de determinadas caixas e, assim, dar prioridade ao carregamento destas caixas em detrimento do carregamento de outras caixas de menor prioridade.

Restrição 10 – Complexidade do Padrão de Empacotamento: caixas presentes em padrões de carregamentos complexos podem demandar, por exemplo, maior esforço de manuseio devido a limitações do equipamento que realiza a carga/descarga das caixas.

Restrição 11 – Limite de Peso: a somatória dos pesos das caixas carregadas dentro do container não pode exceder o peso máximo que o container pode suportar, por exemplo, peso máximo para que um caminhão possa trafegar em uma via expressa. Exemplos aparecem em Scheithauer *et al.* (1996), Terno *et al.* (2000), Bortfeldt & Gehring (2001), Gehring & Bortfeldt (2002), Gendrau *et al.* (2006).

Restrição 12 – Distribuição do Peso dentro do Container: o centro de gravidade de um container carregado deve estar próximo ao centro de gravidade do plano que define a base do container. Exemplos aparecem em Davies & Bischoff (1999), Bortfeldt & Gehring (2001), Eley (2002, 2003).

3.1.4 Padrão de carregamento

Segundo Pisinger (2002), o método utilizado na construção do padrão de carregamento de um container classifica quais as abordagens heurísticas que mais se enquadram. Já o trabalho de Araújo (2006) descreve quatro métodos de construção de padrões de carregamento como se descreve a seguir:

i) Paredes Virtuais: consiste em construir camadas verticais (ou horizontais) que proporcionam uma redução no espaço de soluções tornando a implementação do algoritmo mais simples. O container é preenchido camada por camada. Uma camada vertical j é definida por uma seção do container que contém todos os espaços vazios que pertencem ao conjunto $CV_j = \{(x, y, z) | x_j \leq x \leq x_{j+1}, 0 \leq y \leq W, 0 \leq z \leq H\}$ onde x_j e x_{j+1} representam as posições das camadas verticais j e $j + 1$. No caso de camadas horizontais a definição é análoga, porém com uma troca de papéis entre o eixo horizontal x e o eixo vertical z .

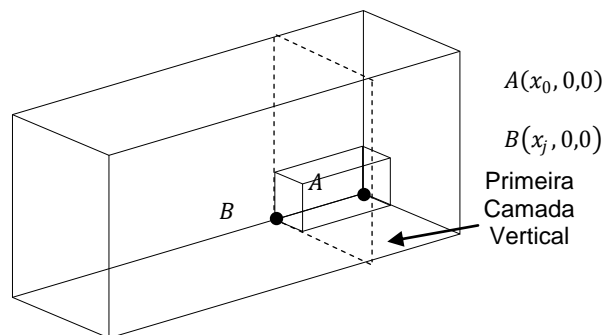


FIGURA 3.6 – REPRESENTAÇÃO DA PRIMEIRA CAMADA VERTICAL DE UM CONTAINER
Fonte: O Autor

ii) Pilhas de Caixas (ou Torres de caixas): este método possibilita a decomposição do carregamento tridimensional do container em dois subproblemas, o problema tridimensional de construção da pilha (torre) de caixas e o problema bidimensional de alocação das pilhas (torres) no container. A construção da pilha (torre) inicia com a escolha da caixa que será posicionada no piso do container,

chamada de caixa base, a seguir as demais caixas devem sempre ser colocadas sobre um caixa da pilha (torre). Uma vez que todas as caixas foram distribuídas em pilhas (torres) o próximo passo consiste em dispor estas pilhas no piso do container. Este procedimento favorece o tratamento de algumas restrições, como a fragilidade das caixas, mas, em geral, produz padrões de carregamento com pouca estabilidade horizontal e baixa ocupação do volume do container, principalmente quando as caixas são heterogêneas. O problema da estabilidade horizontal pode ser amenizado com o 'envelopamento' da pilha (torre) usando plástico filme.

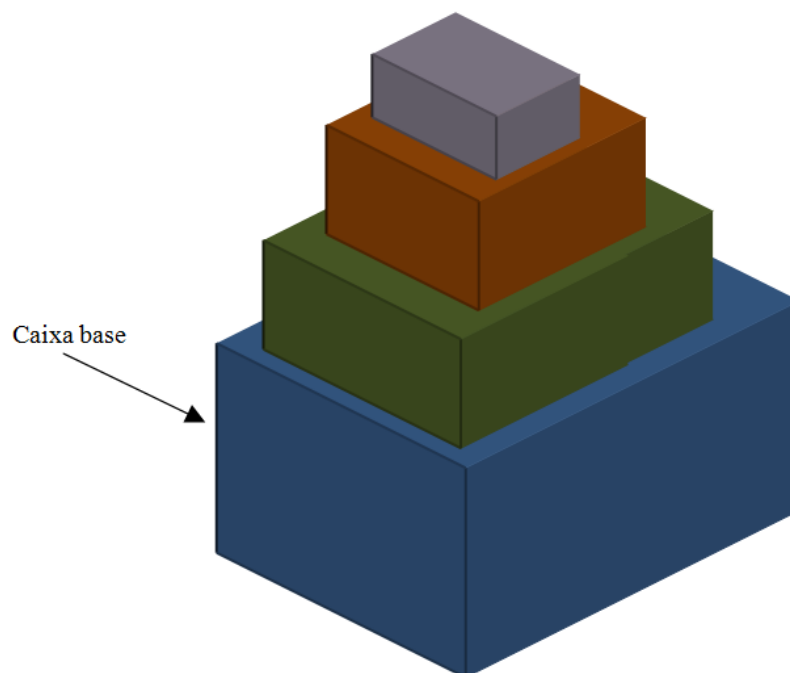


FIGURA 3.7 – EXEMPLO DE PILHA (TORRE) DE CAIXAS
Fonte: O Autor

iii) Cortes Guilhotinados: são classificados como guilhotinados os cortes que se fazem sobre um paralelepípedo produzindo dois novos paralelepípedos. Neste modelo de padrão de carregamento todas as caixas podem ser separadas por uma sequência de cortes guilhotinados. A restrição de cortes guilhotinados é muito mais comum em problemas bidimensionais do que em problemas tridimensionais. Abaixo vê-se um exemplo de um padrão com corte guilhotinado e outro padrão com corte não guilhotinado.

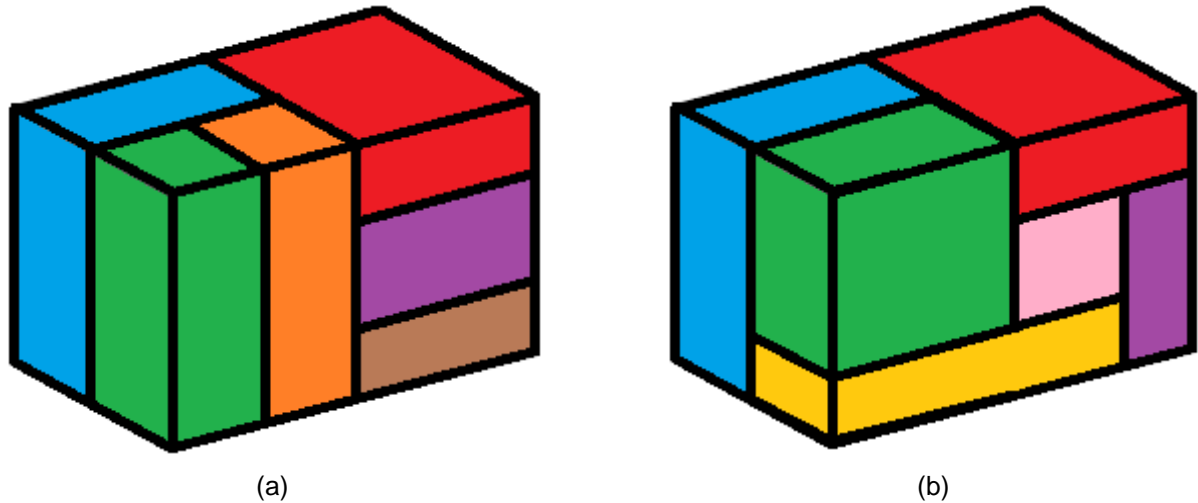


FIGURA 3.8 – EXEMPLOS DE PADRÕES DE CORTE TRIDIMENSIONAL GUILHOTINADO (A) E NÃO GUILHOTINADO (B)

Fonte: O Autor

iv) Cubóides: são blocos homogêneos formados por caixas do mesmo tipo e com a mesma orientação. As caixas que formam o cubóide devem estar dispostas ao longo das três dimensões do container de modo a formarem um paralelepípedo, podendo ser representado por uma tripla onde cada elemento representa a quantidade de caixas em cada eixo, por exemplo: $c_{tki} = (2,3,3)$, um cubóide formado por caixas do tipo t na orientação k ocupando o espaço vazio i do container que possui duas caixas no sentido da largura do container e três caixas nos sentidos da profundidade e da altura do container (Figura 3.9).

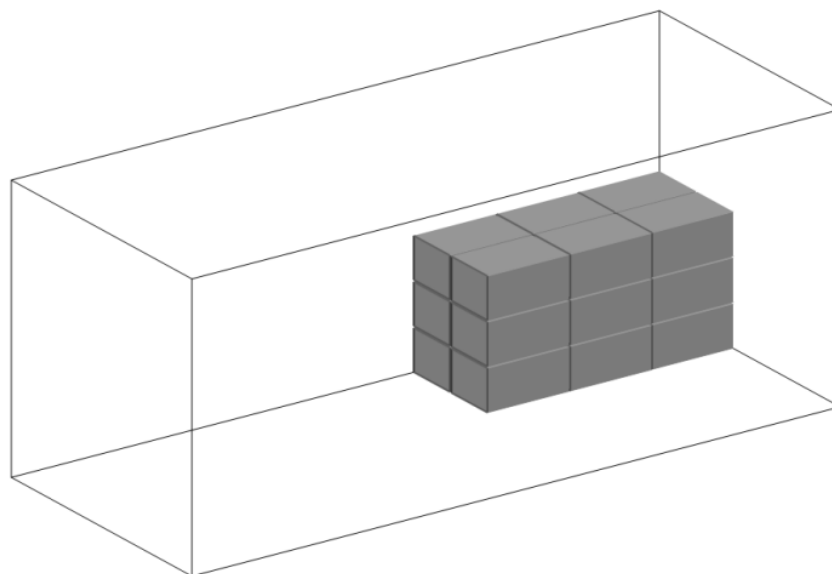


FIGURA 3.9 – EXEMPLO DE CUBÓIDE $c_{tki} = (2,3,3)$

Fonte: O Autor

3.2 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)

O PRV é um problema comum para empresas que trabalham com transporte, logística e distribuição de produtos, onde todos os dias veículos precisam sair de um Centro de Distribuição carregados com produtos e, muitas vezes, entregar estes produtos para clientes diferentes. Este problema pode ser representado por um grafo orientado, como mostra a Figura 3.10 abaixo, onde um veículo deve sair do Centro de Distribuição (nó CD), entregar produtos para cinco clientes diferentes (nós M, N, O, P e Q) e retornar ao Centro de Distribuição.

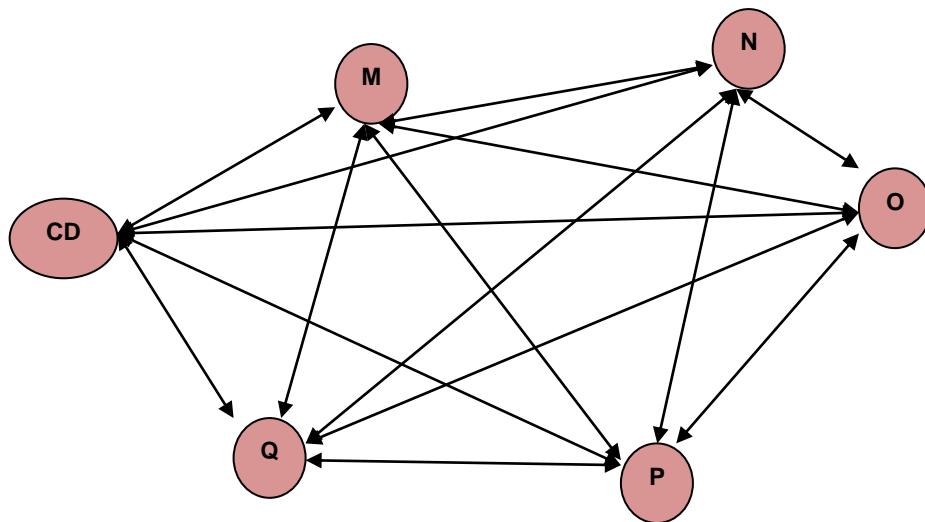


FIGURA 3.10 – EXEMPLO DE ROTEAMENTO DE VEÍCULO A PARTIR DE UM CENTRO DE DISTRIBUIÇÃO (CD)

Fonte: O Autor

Segundo Araujo (2006, p. 44), o objetivo padrão de um PRV “é minimizar os custos das rotas de uma frota de veículos que satisfaçam as demandas dadas pelos clientes e um conjunto de restrições”, onde este custo pode ser mensurado pela distância, condições de tráfego, consumo de combustível, pedágios, etc.

Assim como o Problema de Carregamento de Container, o PRV também pode ser classificado em diversas categorias levando em consideração as características de carga, frota utilizada, tipo de operação, localização dos clientes, conjuntos de restrições, entre outros fatores. Em sua tese de doutorado Belfiore (2006) propõe uma taxonomia para o PRV e suas variações através de 4 categorias: Função Objetivo; Restrições; Variáveis de Decisão e por fim Hipótese/Recurso e as Características do Problema. Reproduz-se abaixo esta classificação:

- **Função Objetivo**
 - ◆ Minimização dos custos totais de distribuição, englobando custos fixos (aquisição do veículo, remuneração dos motoristas, seguros, etc.) e variáveis (combustível, depreciação, etc.);
 - ◆ Minimização da distância total percorrida;
 - ◆ Minimização da duração das rotas (tempo de percurso);
 - ◆ Maximização do nível de serviço;
- **Restrições**
 - ◆ Capacidade de carregamento dos veículos;
 - ◆ Tipo de carga transportada;
 - ◆ Operação de carga e descarga;
 - ◆ Quantidade e tipos de veículos disponíveis;
 - ◆ Janela de tempo de atendimento de cada cliente;
 - ◆ Atendimento total ou parcial das demandas;
 - ◆ Prioridade no atendimento de cliente;
 - ◆ Tempo máximo para carga e descarga;
 - ◆ Restrição de fixação do serviço em algum dia da semana;
 - ◆ Horário de início e término das viagens;
 - ◆ Tempo máximo de viagem de um veículo;
 - ◆ Distância máxima percorrida (autonomia);
 - ◆ Locais de paradas fixas;
- **Variáveis de Decisão**
 - ◆ Roteiro a ser percorrido por cada veículo;
 - ◆ Designação de certo veículo para certo cliente;
 - ◆ Tipo e quantidade de carga transportada para cada cliente da rota;
 - ◆ Tempo de início de atendimento do primeiro cliente da rota;
- **Hipótese/Recurso e Características do Problema**
 - ◆ Referente à operação:
 - Coleta;
 - Entrega;
 - Coleta e entrega simultâneas;
 - Coleta (ou entrega) com carga de retorno;

- ◆ Referente ao carregamento:
 - Carga única ou de lotação;
 - Múltiplas cargas ou carga fracionada;
- ◆ Referente ao tipo de demanda:
 - Determinística;
 - Estocástica;
 - Dinâmica;
- ◆ Referente à localização de demanda:
 - Localizada somente em nós;
 - Localizada somente em arcos;
 - Localizada em nós e arcos;
- ◆ Referente ao tamanho da frota:
 - Finito;
 - Infinito;
- ◆ Referente ao tipo de frota:
 - Homogênea;
 - Heterogênea;

Ainda na mesma tese a autora denomina diversos problemas a partir dos parâmetros descritos acima. Segue abaixo esta denominação.

- ◆ Problema do Caixeiro Viajante;
- ◆ Problema do Carteiro Chinês;
- ◆ Problema dos Múltiplos Caixeiros Viajantes;
- ◆ Problema de Roteamento de Veículos;
- ◆ Problema de Roteamento de Veículos com Demandas Estocástica;
- ◆ Problema de Roteamento de Veículos com Entregas Fracionadas;
- ◆ Problema de Roteamento e Dimensionamento de Veículos com Frota Homogênea;
- ◆ Problema de Roteamento de Veículos com Frota Heterogênea Fixa;
- ◆ Problema de Roteamento e Dimensionamento de Veículos com Frota Heterogênea;
- ◆ Problema de Roteamento de Veículos Periódico;

- ◆ Problema de Roteamento de Veículos com Tempo Dependente;
- ◆ Problema de Roteamento de Veículos com Janelas de Tempo;
- ◆ Problema de Roteamento de Veículos com Janelas de Tempo Flexíveis;
- ◆ Problema de Roteamento de Veículos com Coleta e Entrega;

Para uma análise mais detalhada sobre cada uma das abordagens do PRV apresentadas acima veja Toth & Vigo (2002), Braysy & Gendreau (2005a), Braysy & Gendreau (2005b), além da tese de Belfiore (2006) citada anteriormente.

Vê-se então que campo de pesquisa dos problemas de roteamento se estende por uma série de ramificações e uma destas é a integração entre o PRV e o Problema de Carregamento de Container, onde as demandas são compostas por caixas e, para fazer o roteamento, deve-se considerar as restrições inerentes ao carregamento das demandas no veículo.

Esta ramificação específica da integração do PRV e do PCC tem sido mais abordada durante os últimos 10 anos e, além da tese de doutorado de Araújo (2006) citado anteriormente, destacam-se ainda o trabalho de Iori *et al.* (2006) e a dissertação de mestrado de Guimarães (2011). O trabalho de Iori apresenta um caso específico de empacotamento bidimensional. A frota é composta por veículos idênticos e cada veículo possui uma superfície bidimensional para armazenar as demandas e uma capacidade máxima de peso. Seu objetivo é encontrar uma partição de clientes em rotas de custo total mínimo, sendo que o peso de carregamento em cada veículo não deve ser excedido e algumas restrições de carregamento como: agrupamento dos itens de um mesmo cliente, orientação fixa das caixas e múltiplos destinos para cada veículo, devem ser atendidas. O autor propõe um Algoritmo *Branch-and-Cut* e um Algoritmo de Busca Tabu para a resolução do problema.

Outro trabalho importante sobre o assunto é o de Gendreau *et al.* (2006), nele os autores consideram um carregamento tridimensional com uma frota fixa de veículos. Conseqüentemente, por se tratar de uma proposta tridimensional, os autores relaxaram a restrição quanto à orientação das caixas, fixando apenas a altura das mesmas considerando restrições de Empilhamento, onde as caixas foram classificadas como frágeis e não frágeis, e restrições de Estabilidade, onde a área

de suporte de uma caixa não pode ser menor que 75% de sua área de base (Figura 3.11).

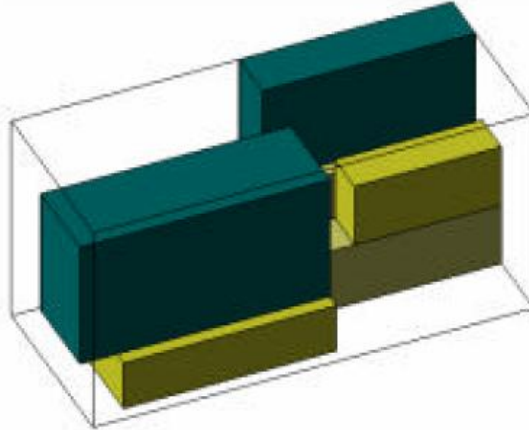


FIGURA 3.11 – CARREGAMENTO FACTÍVEL PELA PROPOSTA DE GENDRAU *ET AL.* (2006)
Fonte: Araújo (2006)

3.3 ALGORITMO GENÉTICO (AG)

Na integração do PCC com o PRV integrado será utilizado um Algoritmo Genético (AG). Algoritmos Genéticos são algoritmos computacionais baseados nos mecanismos de evolução natural e na genética. Os AG's nasceram dos trabalhos de John Holland, na década de 60, sobre algoritmos de otimização de inspiração biológica que resultou na publicação do livro *Adaptation in Natural and Artificial Systems* (HOLLAND, 1975). Com a publicação de um livro tutorial sobre AG's por um aluno de doutorado de Holland, David Goldber, os AG's ganharam popularidade (GOLDBERG, 1986). Abaixo tem-se uma definição interessante para os AG's retirada de um artigo do II Congresso Brasileiro de Redes Neurais (1995):

Algoritmos Genéticos (AG's) são métodos computacionais de busca baseados nos mecanismos de evolução natural e na genética. Em AG's, uma população de possíveis soluções para o problema em questão evolui de acordo com operadores probabilísticos concebidos a partir de metáforas biológicas, de modo que há uma tendência de que, na média, os indivíduos representem soluções cada vez melhores à medida que o processo evolutivo continua. (TANOMARU, 1995, p. 3)

Os AG's utilizam uma série de conceitos da biologia como indivíduos, populações, reprodução, cruzamento e mutação, entre outros. A seguir serão descritos alguns dos conceitos mais importantes.

3.3.1 Indivíduos

Cada indivíduo é representado por um cromossomo, uma sequência de genes. Cada gene ocupa um local fixo no cromossomo e pode assumir um valor que representa uma característica do indivíduo.

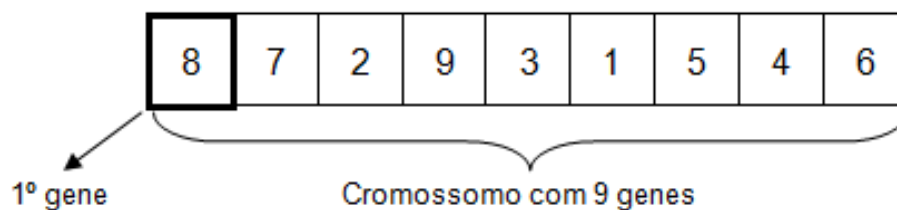


FIGURA 3.12 – REPRESENTAÇÃO DE UM CROMOSSOMO EM AG'S
Fonte: O Autor

Cada indivíduo possui uma adaptabilidade, ou *fitness* do inglês, que é o valor que representa a qualidade deste indivíduo, como por exemplo, a distância total percorrida por todos os veículos que farão as entregas dos pedidos aos clientes seguindo a sequência apresentada no cromossomo. Quanto maior a adaptabilidade (*fitness*) maior será a probabilidade do indivíduo sobreviver ao ambiente, reproduzir-se e gerar descendentes passando partes do seu material genético às próximas gerações.

3.3.2 População

Uma população reúne os indivíduos mais adaptados da geração anterior, com exceção da população inicial que é gerada aleatoriamente ou por uma heurística própria. O tamanho da população é um fator que influencia na resolução do problema, já que populações muito grandes tornam o AG 'pesado' consumindo muito tempo de processamento para formar cada geração. Segundo Corrêa (2000), populações entre 50 e 200 cromossomos resolvem a maior parte dos problemas.

3.3.3 Operadores Genéticos

Existem basicamente três tipos de Operadores Genéticos (OG) comuns em AG's: a Reprodução, o Cruzamento e a Mutação.

3.3.3.1 Operador Genético de Reprodução

O OG de Reprodução apenas copia um indivíduo de uma geração para a próxima geração. Este operador é aplicado a uma parcela dos indivíduos de uma população de forma a garantir que estes indivíduos apareçam na próxima geração. Em outras palavras, o OG de Reprodução é a clonagem de alguns indivíduos.

3.3.3.2 Operador Genético de Cruzamento

O OG de Cruzamento seleciona dois indivíduos de uma geração (pais) e um ponto de cruzamento aleatório. Então, gera dois novos indivíduos para uma nova geração (filhos) utilizando em cada um dos filhos uma parte do cromossomo de cada pai (Figura 3.13).

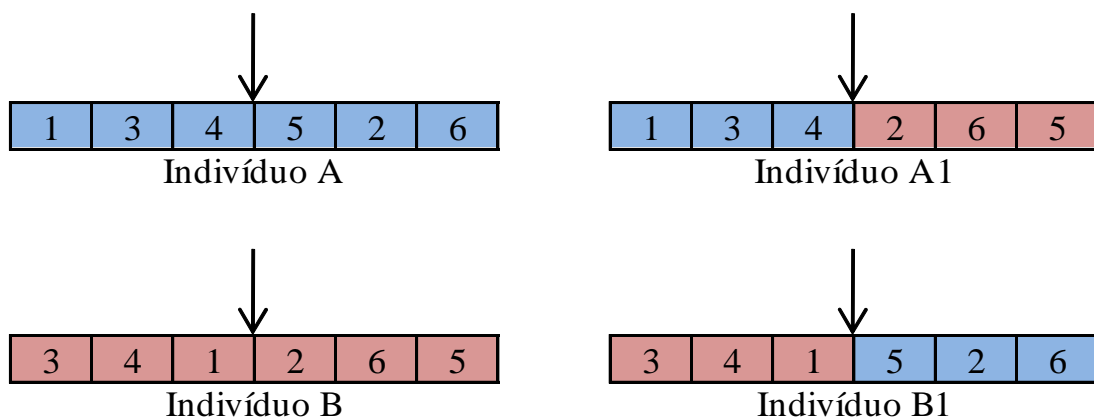


FIGURA 3.13 – REPRESENTAÇÃO DE UMA OPERAÇÃO DE CRUZAMENTO

Fonte: O Autor

A tendência é que os dois novos indivíduos (filhos) após a realização do OG de Cruzamento possuam características bem diferentes, já que um herdou a parte inicial de seu cromossomo do 1º indivíduo original (pai) e a parte final de seu cromossomo do 2º indivíduo original (pai), enquanto o outro recebeu exatamente o

contrário. Como na seleção natural, existe a tendência de um filho apresentar maior adaptabilidade (*fitness*) que o outro e por este motivo ele possui maior probabilidade de se reproduzir nas próximas gerações, enquanto o filho com menor adaptabilidade (*fitness*) possui a tendência de desaparecer em algumas gerações.

3.3.3.3 Operador Genético de Mutação

O OG de Mutação seleciona um indivíduo e dois genes deste indivíduo. Então, gera um novo indivíduo trocando estes dois genes de posição (Figura 3.14).

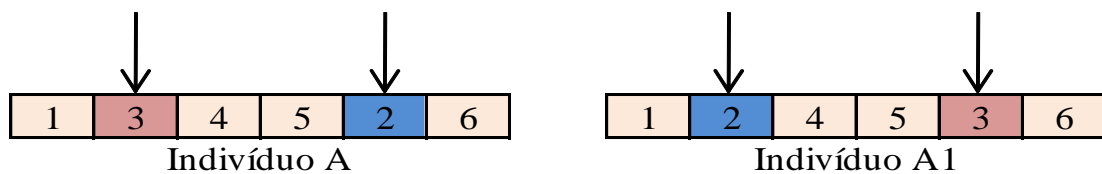


FIGURA 3.14 – REPRESENTAÇÃO DE UMA OPERAÇÃO DE MUTAÇÃO
Fonte: O Autor

O OG de Mutação possui um papel fundamental na evolução populacional pois é capaz de introduzir, ou restaurar, características novas aos indivíduos, além de evitar a convergência prematura para ótimos locais. Segundo Vendramini (2007) a implementação do OG de Mutação deve ser realizada com precaução, pois pode acarretar num custo computacional elevado.

3.3.4 Funcionamento Básico de um AG

De maneira simples, pode-se descrever um AG através da seguinte sequência de procedimentos:

i) toma-se uma população inicial de soluções factíveis para o problema e ordena-se esta população de acordo com a qualidade de cada solução (adaptabilidade ou *fitness*);

ii) selecionam-se, através de probabilidade, alguns indivíduos desta população e com estes realizam-se os OG de reprodução, ou cruzamento ou mutação a fim de gerar novos indivíduos, que por sua vez formaram uma nova geração;

iii) a partir desta nova geração, tem-se então, uma nova população, com esta população o processo reinicia até que determinada condição seja satisfeita.

Esta condição pode ser escolhida pelo usuário do AG: o tempo computacional, o número de gerações, uma melhoria percentual dos indivíduos de cada geração, a estagnação dos indivíduos de uma geração, entre outros. Quando esta condição é então satisfeita o AG finaliza e, então, o primeiro cromossomo da última geração encontrada é tomado como solução para o problema.

3.4 ALGORITMO *BOTTOM-LEFT* (ABL)

Durante a resolução do problema proposto, para cada cromossomo encontrado no AG, será utilizado um Algoritmo *Bottom-Left* (ABL). Este algoritmo determinará de que forma serão acondicionados os produtos que cada veículo deverá transportar. O ABL trata este problema como um problema de corte bidimensional não guilhotinado sobre o piso do compartimento de carga do veículo.

Um trabalho pioneiro sobre o assunto de cortes foi realizado por Gilmore & Gomory (1961), eles estudaram o problema de corte unidimensional (*cutting-stock*). Após este trabalho, diversos outros foram desenvolvidos para cortes unidimensionais, bidimensionais e até tridimensionais. Rocha (1997) utilizou métodos de geração de colunas enquanto Ciurlizza & Chávez (1998) *Simulated Annealing* e Hopper & Turton (1999) utilizaram o AG combinado com o ABL que têm a função de realizar o encaixe das peças na chapa bidimensional.

O funcionamento do ABL pode ser considerado de simples compreensão já que, em síntese, o algoritmo busca encaixar diversas as peças numa chapa retangular mais ao fundo e mais à esquerda possível. No problema tratado as peças serão representadas pela base das torres de caixas e a chapa retangular será a superfície interna do container. A partir de uma lista de peças a serem encaixadas, o algoritmo então toma a primeira peça e a encaixa na parte inferior-esquerda da chapa. Após o encaixe da primeira peça, as peças seguintes da lista serão encaixadas mais ao fundo e à esquerda possível. A figura 3.15 abaixo ilustra o corte de uma chapa (superfície interna do container) bidimensional usando o ABL, em quatro imagens. A imagem (a) possui três peças já posicionadas a partir do canto inferior esquerdo da chapa e uma quarta peça sendo posicionada com o primeiro movimento em direção ao fundo da chapa. A imagem (b) mostra o segundo

movimento da quarta peça, agora para à esquerda. A figura (c) mostra que a quarta peça ainda pode ser deslocada em direção ao fundo da chapa e não mais para a esquerda. A figura (d) mostra a chapa com doze peças posicionadas para o corte não guilhotinado, através do ABL.

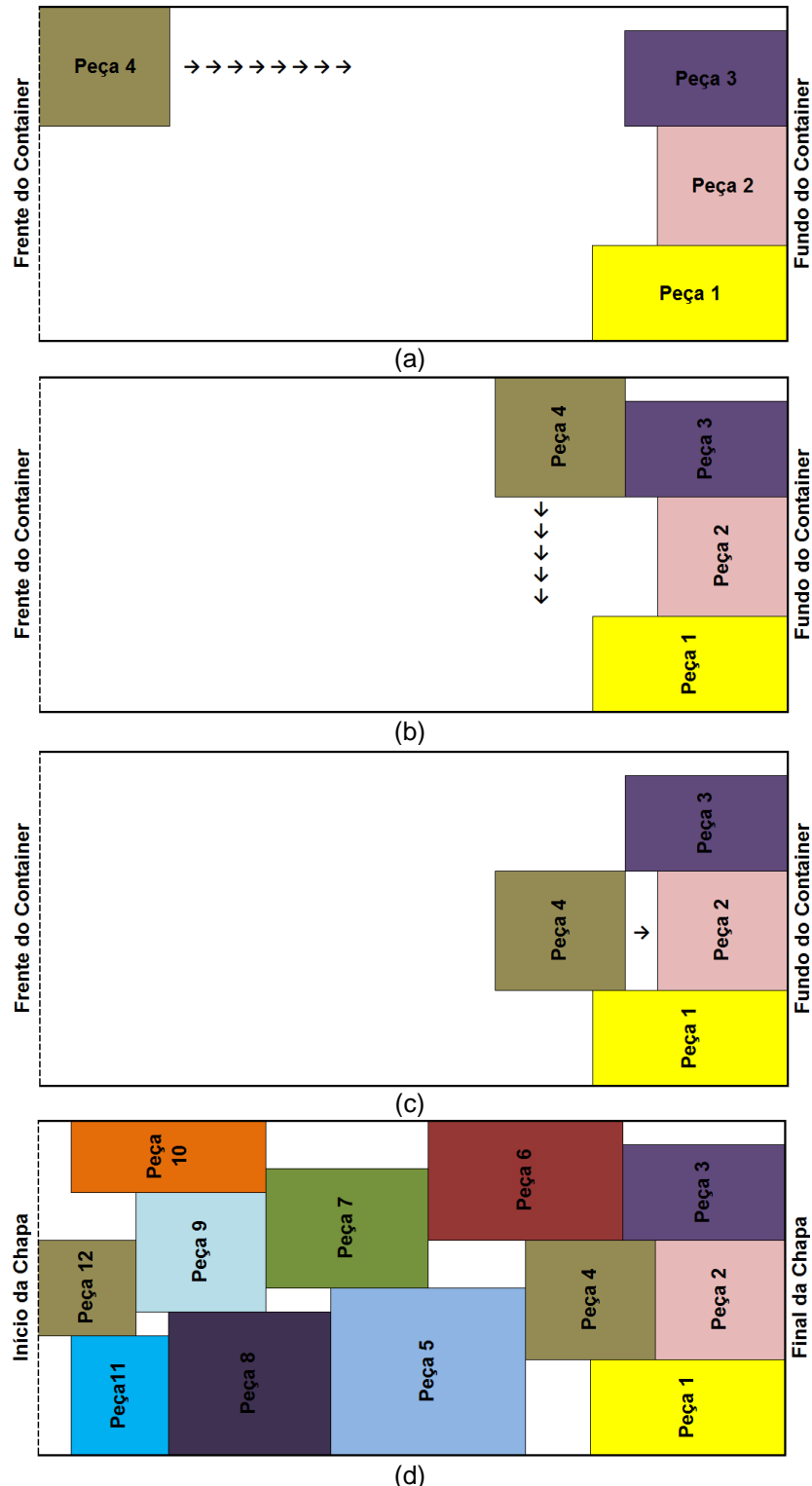


FIGURA 3.15 – CORTE BIDIMENSIONAL COM O ABL.
 Fonte: O Autor

Com isso, pode-se facilmente perceber que se com o ABL os cortes são feitos na chapa (superfície interna do container) seguindo a sequência Fundo-Esquerda, estes mesmos cortes poderão ser desfeitos um a um na sequência inversa de que foram feitos na chapa, seguindo a sequência Direita-Frente. Esta consequência justifica a escolha deste algoritmo no arranjo das torres de caixas dentro do container, já que garante que todas as torres poderão ser descarregadas na ordem inversa de carregamento sem que seja necessário movimentar outra torre dentro do container. Assim, seguindo a sequência inversa do roteamento para se carregar o container, tem-se justamente a sequência do roteamento para fazer o descarregamento das torres de caixas

A seguir será detalhada a metodologia de resolução do problema proposto por este artigo.

4 METODOLOGIA

O capítulo apresenta uma proposta para resolver o problema de entregas. A metodologia de resolução do problema foi dividida em quatro etapas para facilitar seu desenvolvimento:

- i) o recebimento dos pedidos e formação das torres de produtos de cada pedido;
- ii) a geração de soluções iniciais do carregamento e do roteamento, com utilização do ABL para cálculo da adaptabilidade das soluções;
- iii) a aplicação do AG para otimização através de evolução, com uso do ABL mais uma vez para o cálculo da adaptabilidade, até que um critério seja atendido, e então, a melhor resposta encontrada é tomada como solução do problema;
- iv) a construção das informações enviadas a cada um dos diferentes setores envolvidos na prática do problema:
 - a) Como são formadas/carregadas as torres de produtos?
 - b) Quantos veículos serão necessários?
 - c) Qual a sequência de carregamento de cada veículo?
 - d) Qual o roteiro de entrega e a distância percorrida por cada veículo?
 - e) Quantos quilômetros cada veículo irá percorrer?

Cada uma destas etapas será detalhada a seguir.

4.1 RECEBIMENTO DOS PEDIDOS

A entrega de eletrodomésticos da linha branca vendidos à pessoa física é um problema comum para grandes redes varejistas. Estas empresas possuem diversas lojas (físicas e virtuais) e Centros de Distribuição de Produtos (CD). Os Centros de Distribuição geralmente são localizados próximos aos grandes centros urbanos e é destes que partem as entregas dos produtos comprados nas lojas desta empresa. Cada CD fica responsável por entregar os produtos comprados por clientes de determinada região.

Uma série de pedidos recebidos pela empresa são enviados ao Centro de Distribuição que fará a entrega. No CD os produtos de cada pedido são organizados

em torres (pilhas) de caixas buscando maximizar o espaço que sobra sobre uma caixa do pedido e, assim, minimizar a superfície ocupada do container.

A formação das torres de caixas se dará através de um Algoritmo de Formação de Torres (AFT), baseado nos trabalhos de Gehring & Bortfeldt (1997) e Liang *et al.* (2007), que leva em consideração as seguintes restrições (páginas 28, 29 e 30):

R1 – Orientação: as caixas só poderão ser rotacionadas em torno do eixo vertical, ou seja, trocar a dimensão da largura pela da profundidade, e vice-versa.

R2 – Empilhamento: uma caixa só poderá ser adicionada à torre se a pressão que ela exerce for menor ou igual à resistência da torre;

R4 – Estabilidade: cada caixa deve ser totalmente apoiada na caixa imediatamente abaixo dela ou no piso do container, ou seja, uma caixa que não é a base da torre só poderá ser adicionada à torre se sua base possuir dimensões menores ou iguais às dimensões do topo da última caixa alocada na torre;

R6 – Múltiplos Destinos: esta restrição justifica também a construção de torres, pois cada torre só possui produtos de um mesmo destino, e também faz com que as torres de um mesmo pedido sejam carregadas próximas umas das outras, dentro de um mesmo container.

Além destas restrições, também será considerada uma restrição em relação à altura da torre: a altura da torre deve ser menor ou igual à altura do container, ou seja, uma torre não poderá ser carregada se a soma de sua altura com a altura da torre ultrapassar a altura do container. O AFT está descrito abaixo:

Passo1
Dado um determinado pedido, formar uma lista com as q caixas, levando em conta os diferentes tipos e quantidades de cada caixa, em ordem decrescente de largura: L(qcaixas) – Em caso de empate na largura, fazer ordem decrescente de profundidade; em caso de novo empate, fazer ordem decrescente de pressão da base. Zerar quantidade de torres: qtor = 0
Passo 2
Testar $q > 0$? Se sim Prosseguir. Se não Fim do algoritmo.
Passo 3
Fazer $i = 0$ Atualizar a quantidade de torres: $qtor = qtor + 1$ Fazer o primeiro elemento da lista igual ao primeiro elemento da torre. Fazer a profundidade da torre igual a profundidade do primeiro elemento da lista. Fazer a altura da torre igual a altura do primeiro elemento da lista. Fazer a resistência da torre igual a resistência do primeiro elemento da lista. Fazer $j = 2$

Passo 4
<p>Comparar profundidade do item L(j) da lista com a profundidade da torre: L(j).prof <= torre(qtor).prof ?</p> <p>Se sim Prosseguir.</p> <p>Se não Fazer j = j+1 Testar j <= qcaixas ? Se sim Voltar ao passo 4 Se não Eliminar de L as caixas utilizadas nesta torre Finalizar a torre e atualizar L e qcaixas Voltar ao Passo 2</p>
Passo 5
<p>Comparar a altura do item L(j) da lista com a altura da torre e a altura do container: L(j).alt + torre(qtor).alt <= container.alt ?</p> <p>Se sim Prosseguir.</p> <p>Se não Fazer j = j+1 Testar j <= qcaixas ? Se sim Voltar ao passo 4 Se não Eliminar de L as caixas utilizadas nesta torre Finalizar a torre e atualizar L e qcaixas Voltar ao Passo 2</p>
Passo 6
<p>Comparar a pressão do item L(j) da lista com a resistência da torre: L(j).pressão <= torre(qtor).res ?</p> <p>Se sim Prosseguir.</p> <p>Se não Fazer j = j+1 Testar j <= qcaixas ? Se sim Voltar ao passo 4 Se não Eliminar de L as caixas utilizadas nesta torre Finalizar a torre e atualizar L e qcaixas Voltar ao Passo 2</p>
Passo 7
<p>Fazer i = i + 1 Fazer a caixa j de L: L(j), a caixa i da torre qtor. Fazer a profundidade da torre igual a profundidade do elemento j da lista. Fazer a altura da torre igual a altura atual da torre mais a altura do elemento j da lista. Fazer a resistência da torre igual ao mínimo entre a resistência da torre menos a pressão do elemento j da lista, ou a resistência do elemento j da lista. Fazer j = j+1 Testar j <= qcaixas ? Se sim Voltar ao passo 4 Se não Eliminar de L as caixas utilizadas nesta torre Finalizar a torre e atualizar L e qcaixas Voltar ao Passo 2</p>

QUADRO 4.1 – ALGORITMO DE FORMAÇÃO DE TORRES - AFT.
Fonte: O Autor

Vale lembrar que, como se admite a rotação da caixa sobre o eixo z , uma caixa que não seja carregada por ter sua profundidade maior que a profundidade da torre pode ser rotacionada. Então, testa-se esta profundidade da caixa com a largura da torre e a largura da caixa com a profundidade da torre. Caso seja possível o carregamento desta caixa rotacionada, o número que representa esta caixa perante a base de dados será adicionado de mil unidades (exemplo: o sétimo produto da lista foi carregado rotacionado, logo será classificado como “Produto 1007” ao invés de “Produto 7”).

Após receber todos os pedidos são geradas soluções iniciais de acordo com a metodologia descrita a seguir.

4.2 SOLUÇÕES INICIAIS

As soluções iniciais para o problema de entregas são geradas aleatoriamente. Todos os pedidos recebidos são numerados de 1 a p , então um número aleatório menor ou igual a p é escolhido, este é o primeiro pedido da solução aleatória. Este pedido é retirado da sequência, que fica então com $p - 1$ elementos, desta nova sequência é extraído o próximo pedido, semelhantemente ao primeiro pedido. Quando o último pedido for colocado na sequência tem-se uma solução para o seqüenciamento das entregas, que tende a ser uma resposta muito ruim, já que é aleatória num universo de soluções com $p!$ possibilidades.

E com esta solução inicial aleatória, cria-se uma família de soluções iniciais fazendo um rodízio nesta primeira solução encontrada, ou seja, a segunda solução iniciará pelo segundo elemento da primeira solução, seguido pelo terceiro elemento da primeira solução e assim por diante, finalizando a sequência com o primeiro elemento da primeira solução. A terceira solução inicial, assim como todas as próximas, será formada de modo semelhante à segunda, garantindo assim que nesta família de soluções iniciais, nenhuma solução seja igual a outra.

Para cada uma destas soluções iniciais é realizado um ABL para carregar as torres dos pedidos na sequência da solução. Assim, encontra-se a quantidade de veículos necessários para realizar a entrega nesta sequência, quais pedidos serão carregados em cada um dos veículos e a distância percorrida por cada veículo. A soma das distâncias percorridas por cada veículo é a distância total percorrida pela

solução e será utilizada como função de adaptabilidade no AG, ou seja, quanto menor for esta distância total melhor será a solução.

Passo 1
A primeira torre de caixas carregada no container deve ser colocada no fundo do container e à esquerda. A partir da segunda torre de caixas, seguir passo 2.
Passo 2
Empurrar a torre de caixas para o fundo do container até que encoste o fundo ou até que encoste noutra torre que já esteja carregada
Passo 3
Empurrar esta torre para a esquerda até que encoste na lateral do container ou até que encoste noutra torre que já esteja carregada
Passo 4
Tentar repetir o passo 2 e depois o passo 3, até que um não seja mais possível.

QUADRO 4.2 – PSEUDO-CÓDIGO PARA O ABL UTILIZADO.

Fonte: O Autor

É importante ressaltar que um pedido deve ser carregado integralmente num único container (página 30 – R6). Se durante o carregamento de determinado pedido pelo menos um produto deste pedido não puder ser acondicionado dentro do container este pedido deverá ser descarregado do mesmo e carregado num próximo container (Figura 4.1).

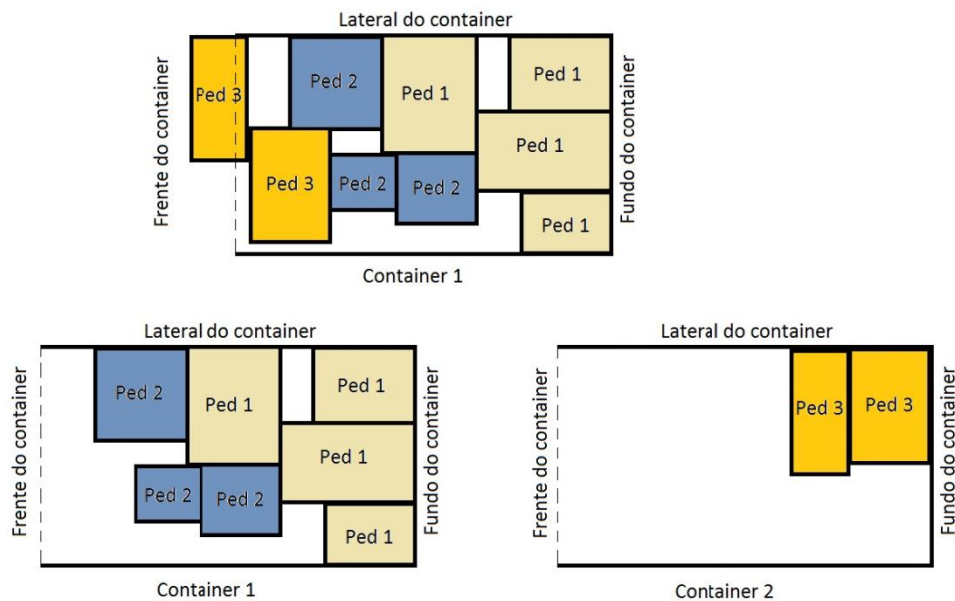


FIGURA 4.1 – RECONDICIONAMENTO DE PEDIDO QUE EXTRAPOLA AS DIMENSÕES DO CONTAINER.

Fonte: O Autor

Encontrada então esta família de soluções iniciais aleatórias tem-se um conjunto inicial para aplicar o AG a fim de otimizar as soluções através de processos evolutivos, como será descrito a seguir.

4.3 APLICAÇÃO DO AG E EVOLUÇÃO DAS SOLUÇÕES

Formado o conjunto de soluções iniciais, aplica-se então, um AG com o objetivo de procurar soluções com a menor distância total percorrida (função de adaptabilidade ou *fitness*). Na presente dissertação vê-se que na maioria dos casos a quantidade de containeres utilizados também acaba sendo minimizada. Os indivíduos serão representados por cromossomos formados pela sequência dos pedidos (página 39). Com estes indivíduos serão realizados os Operadores Genéticos de Reprodução, Cruzamento e Mutação descritos (páginas 40 e 41) a fim de encontrar uma nova geração de indivíduos. Testes empíricos realizados por DeJong (1975) mostraram que proporções em torno de 70% para o OG de Cruzamento apresentaram bons resultados. Este mesmo autor sugere também uma proporção baixa para o OG de Mutação, que na presente dissertação irá variar entre 5% e 10%. O restante dos indivíduos da população participa do OG de Reprodução.

Ao final das operações genéticas, são selecionadas desta população as melhores soluções encontradas e algumas soluções aleatórias, sem se importar com a adaptabilidade, com o intuito de evitar o elitismo. Estes formam então uma nova população que servirá de base para a próxima geração. O tamanho da população (página 38) interfere na velocidade computacional e também no direcionamento da evolução. Por isso, foram realizados testes com populações de 50 a 200 indivíduos, sempre considerando a quantidade de pedidos (Quadro 4.3).

Número de Pedidos	Tamanho da População
Até 60 pedidos	Igual ao Número de Pedidos
De 61 a 100 pedidos	De 50 a 60 indivíduos
De 101 a 150 pedidos	De 60 a 100 indivíduos
De 151 a 300 pedidos	De 100 a 150 indivíduos
A partir de 301 pedidos	De 150 a 200 indivíduos

QUADRO 4.3 – TAMANHO DA POPULAÇÃO EM FUNÇÃO DO NÚMERO DE PEDIDOS.

Fonte: O Autor

O critério de parada do AG escolhido nesta proposta é constituído por dois parâmetros classificados como “Estabilidade na População” e “Tempo Computacional”. O primeiro parâmetro leva em conta que uma parcela (50% das melhores soluções – Elite da população) sofre alterações praticamente imperceptíveis durante uma série de gerações (20 à 30 gerações). O segundo parâmetro utilizado leva em conta o tempo de processamento do AG (em alguns casos o limite foi fixado em 240 segundos) em conjunto com a condição de que deve haver uma melhoria percentual significativa nas soluções (mínimo de 16% de melhoria). Quando um destes parâmetros é obedecido o AG finaliza e a melhor solução da última geração será utilizada como solução do problema.

Vale ressaltar que, numa aplicação real o critério “Tempo Computacional” pode ser extremamente relaxado isto porque a empresa não fará entregas durante a madrugada para Pessoas Física. Sendo assim, a empresa pode facilmente despende de minutos ou até horas para que a resolução do problema encontre soluções ainda melhores. Tendo este resultado no meio da madrugada, os funcionários encarregados do carregamento já começam a separar os produtos que serão carregados, formar as torres e carregar os veículos, de modo que, próximo às 08:00 horas todos os produtos estejam carregados nos veículos e estes estejam prontos para realizar as entregas. Ou ainda, pode-se assumir que os produtos sejam carregados no período da manhã e entregues no período da tarde.

4.4 CONSTRUÇÃO DAS INFORMAÇÕES ENVIADAS AOS SETORES

Esta é uma etapa muito simples da metodologia, que não envolve nenhum algoritmo ou método matemático, porém importante, principalmente na implantação da metodologia. É fundamental que, após resolver o problema integrado entre o Problema de Carregamento de Container e o Problema de Roteamento de Veículos, cada setor da empresa responsável pela logística de entrega do produto ao cliente saiba como desempenhar sua função. Logo serão repassadas as seguintes informações, por setor envolvido, ao final do problema:

a) Responsáveis por Estoque e Organização de Mercadorias: os funcionários deste setor receberão a sequência de carregamento de cada veículo por pedidos e a sequência das caixas dos produtos na construção das torres de cada pedido. Sendo assim, capaz de preparar as torres com as mercadorias,

‘envolver’ as torres com o plástico filme e encaminhar as torres em sequência de carregamento para cada veículo.

b) Carregadores dos Veículos: este setor receberá, por veículo, a sequência de carregamento das torres de produtos de cada pedido que será acondicionada naquele veículo e a sequência de pedidos a serem entregues por este veículo.

c) Motoristas dos Veículos: cada motorista receberá um mapa (gerado através do site: www.maps.google.br) apresentando uma trajetória para o motorista realizar as entregas dos pedidos que foram carregados no seu veículo de acordo com a ordem de carregamento dos mesmos.

d) Controladoria/Auditoria: os responsáveis por fiscalizar o trabalho dos setores citados acima receberão como resposta o número de veículos que serão necessários, a distância aproximada percorrida por cada veículo e os pedidos atendidos por cada veículo. Com base nestas informações pode-se fazer o controle da rodagem de cada veículo, estudar possibilidades e necessidades de aumento de frota, entre outras atividades de controle e planejamento.

Este direcionamento de informações tende a trazer uma melhoria no funcionamento diário do Centro de Distribuição de Produtos da empresa. O capítulo seguinte abordará a implementação desta metodologia e os resultados dos testes propostos para tamanho de população e critério de parada, utilizando o cenário fictício apresentado (páginas 20, 21 e 22).

5 IMPLEMENTAÇÃO

O capítulo anterior da presente dissertação propôs uma metodologia para a resolução do problema de entregas, integrando os problemas de Carregamento de Container e de Roteamento de Veículos. Esta proposta utiliza um AG para otimizar as soluções e um ABL para acondicionar os produtos que devem ser transportados em cada um dos veículos de entrega.

Assim, este capítulo apresenta a implementação computacional desta proposta de resolução do problema. Tal implementação computacional foi desenvolvida em linguagem de programação *Microsoft Visual Basic 6.0*©. O computador utilizado na implementação possui processador Intel® Core™2 Duo T5450 1.67GHz com 2.96GB de memória RAM.

Para facilitar a implementação da proposta e simplificar o funcionamento do software para o usuário, este software foi desenvolvido em três etapas de utilização:

i) na primeira etapa o usuário insere as informações do cenário fictício na base de dados do software;

ii) na segunda etapa o usuário pode escolher entre três métodos de resolução do problema – seleção manual dos pedidos, seleção automática dos pedidos e diversas seleções automáticas dos pedidos. De acordo com a opção do usuário, o software resolverá o problema de entregas para os pedidos selecionados;

iii) na terceira etapa o usuário pode escolher se deseja que o software lhe apresente as torres de um determinado pedido e/ou construa os arquivos com as informações específicas que serão enviadas a cada um dos setores da empresa envolvidos no problema de entregas.

A seguir, será apresentado a forma pela qual o usuário interage com o software desenvolvido, em cada uma das três etapas de utilização, com figuras que apresentam a interface deste software.

5.1 CRIAÇÃO DO CENÁRIO FICTÍCIO

A implementação computacional da metodologia de resolução do problema de entregas, proposta no capítulo anterior, tem a finalidade de realizar testes nesta metodologia. Para que o software desenvolvido possa realizar estes testes é necessário que a base de dados deste contenha as instâncias do cenário fictício

citado (páginas 20, 21 e 22). Estas instâncias de teste foram criadas aleatoriamente e inseridas na base de dados do software através da primeira etapa desta implementação. Foi desenvolvida uma interface, apresentada nas figuras a seguir, para que o usuário possa cadastrar os produtos e pedidos do cenário fictício.

A figura 5.1 apresenta a interface inicial do software, onde o usuário poderá escolher um nome para a amostra de produtos (caixas) e a quantidade de tipos de produtos que serão inseridas na base de dados. Além desta função, a interface inicial do software apresenta ao usuário duas outras funções: carregar um arquivo com produtos e carregar um arquivo com um pedido, sendo que o usuário terá a possibilidade de escolher entre os arquivos que já estão salvos na base de dados.

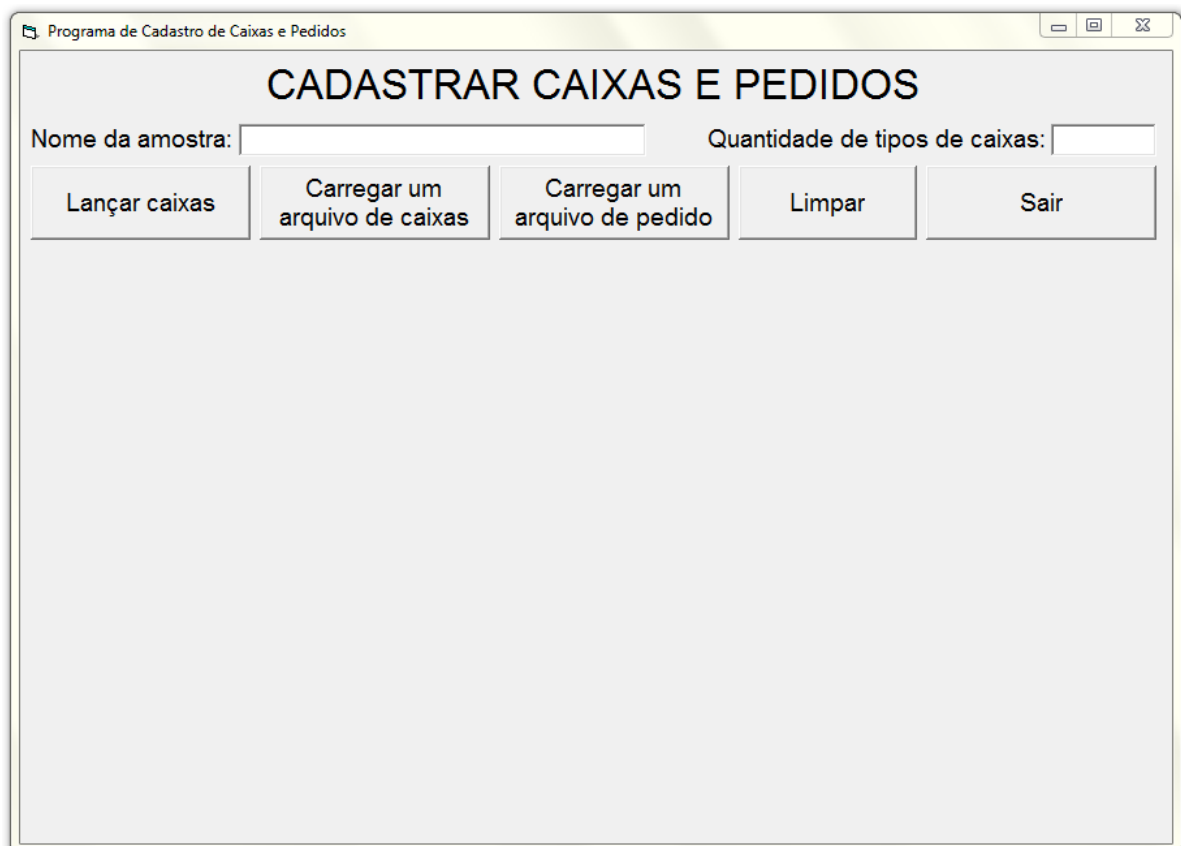


FIGURA 5.1 – INTERFACE INICIAL DO SOFTWARE PARA CADASTRAR PRODUTOS E PEDIDOS.
Fonte: O Autor

Se então o usuário optar por criar uma amostra nova de produtos, o software apresenta uma segunda interface (Figura 5.2) onde o usuário cadastra todas as informações de cada um dos produtos. Após cadastrar todas as informações de um produto, o usuário “salva” este produto e suas informações aparecem numa planilha. Quando o usuário finaliza o cadastro dos produtos é

possível verificar se todos estão corretos na planilha e, caso haja alguma incorreção, o software oferece uma função para correção de um produto específico.

Programa de Cadastro de Caixas e Pedidos

CADASTRAR CAIXAS E PEDIDOS

Nome da amostra: Quantidade de tipos de caixas:

Tipo:

Produto:

Dimensões (em cm) - Largura: Profundidade: Altura:

Peso (em kg): Quantas caixas iguais a esta ela suporta:

Tipo	Descrição do Produto	Larg.	Prof.	Alt.	Peso	Pressão kg/cm2	Resistência
1	Fogão 4 bocas - Atlas	50	60	85	21	0,007	0,014
2	fogão 4 bocas - Esmaltec	50	60	85	22	7,333333E-03	1,466667E-02
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							

FIGURA 5.2 – INTERFACE PARA CADASTRO DAS INFORMAÇÕES DE CADA PRODUTO.

Fonte: O Autor

Após o usuário cadastrar uma amostra de produtos, o software “salva” esta amostra em um arquivo na base de dados. Este arquivo pode ser carregado pela interface inicial do software, como mostra a Figura 5.3.

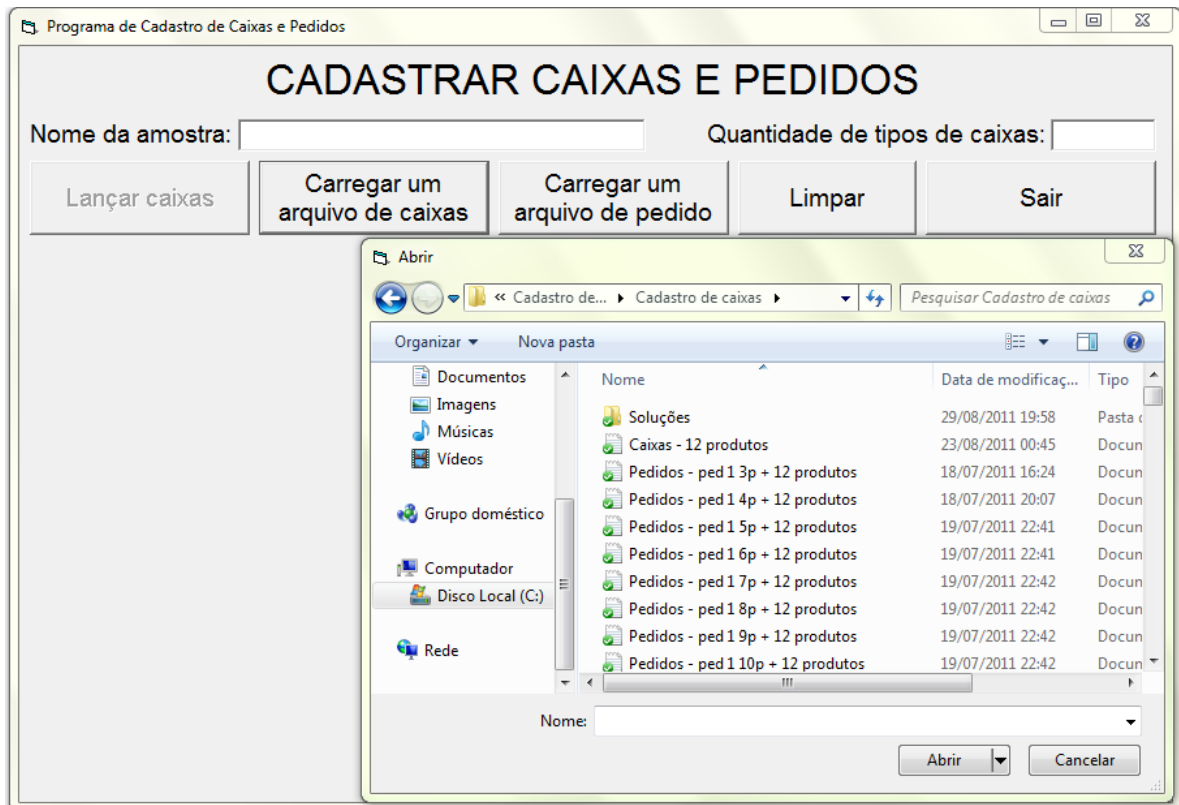


FIGURA 5.3 – INTERFACE PARA CARREGAR O ARQUIVO DE UMA AMOSTRA A PARTIR DA BASE DE DADOS.

Fonte: O Autor

Finalmente, nesta primeira etapa da implementação o usuário ainda pode cadastrar as informações de um pedido com base num arquivo de produtos que já esteja “salvo” na base de dados. Este pedido pode ser cadastrado logo após o cadastramento dos produtos ou o usuário pode optar por abrir um arquivo de produtos da base de dados e cadastrar o pedido baseado neste arquivo de produtos. A Figura 5.4 apresenta a interface do software desenvolvido durante o cadastramento de um pedido pelo usuário.

Programa de Cadastro de Caixas e Pedidos

CADASTRAR CAIXAS E PEDIDOS

Nome da amostra: Quantidade de tipos de caixas:

Tipo	Descrição do Produto	Larg.	Prof.	Alt.	Peso	Pressão k	
1	Fogão 4 bocas - Atlas	50	60	85	21		
2	fogão 4 bocas - Esmaltec	50	60	85	22	7,3333	
3	Lavadora 10kg - Brastemp	70	70	105	43	8,775	
4	Lavadora 10.4kg - Samsung	65	80	95	88	1,6923	0,006
5	Refrigerador 439L - Brastemp	70	75	190	87	1,657143E-02	1,242857E-02
6	Refrigerador 553L - Electrolux	80	75	190	107	1,783333E-02	0,013375
7	Microondas 25L - Consul	60	50	40	18		0,006
8	Microondas 28L - Panasonic	55	45	40	16	6,464647E-03	1,939394E-02
9	Climatizador 7500BTU - Consul	60	60	45	28	7,777778E-03	1,555556E-02
10	Climatizador Portátil - Komeco	40	25	70	7		0,007
11	Forno Elétrico 28L - Mondial	55	70	45	6	1,558442E-03	3,896104E-03
12	Forno Elétrico 23L - Suggar	30	35	45	6	5,714286E-03	1,428571E-02

Problema de Entregas

Digite um nome para o pedido:

FIGURA 5.4 – INTERFACE PARA CADASTRO DAS INFORMAÇÕES DE UM PEDIDO.

Fonte: O Autor

O Cenário Fictício proposto nesta dissertação apresenta uma lista com as informações de 12 produtos que a empresa vende (página 20). Estes produtos foram cadastrados na base de dados e o software gerou um arquivo de texto com as informações dos produtos (Figura 5.5).

Caixas - Amostra com 12 caixas - Bloco de notas

```

Arquivo Editar Formatar Exibir Ajuda
Amostra com 12 caixas
12
Fogão 4 bocas - Atlas,50,60,85,21,0.007,2,0.014
Fogão 4 bocas - Esmaltec,50,60,85,22,7.333333E-03,2,1.466667E-02
Lavadora 10kg - Brastemp,70,70,105,43,8.77551E-03,2,1.755102E-02
Lavadora 10.4kg - Samsung,65,80,95,88,1.692308E-02,2,3.384615E-02
Refrigerador 439L - Brastemp,70,75,190,87,1.657143E-02,0.75,1.242857E-02
Refrigerador 553L - Electrolux,80,75,190,107,1.783333E-02,0.75,0.013375
Microondas 25L - Consul,60,50,40,18,0.006,3,0.018
Microondas 28L - Panasonic,55,45,40,16,6.464647E-03,3,1.939394E-02
Climatizador 7500BTU - Consul,60,60,45,28,7.777778E-03,2,1.555556E-02
Climatizador Portátil - Komeco,40,25,70,7,0.007,2,0.014
Forno Elétrico 28L - Mondial,55,70,45,6,1.558442E-03,2,5,3.896104E-03
Forno Elétrico 23L - Suggar,30,35,45,6,5.714286E-03,2,5,1.428571E-02

```

FIGURA 5.5 – ARQUIVO DE PRODUTOS SALVO NA BASE DE DADOS.

Fonte: O Autor

Com base neste arquivo de produtos foram gerados neste cenário fictício 160 pedidos aleatórios diferentes (página 21). Cada um destes pedidos foi cadastrado através do software, gerando um arquivo de texto para cada um destes pedidos, que contenha as informações destes (Figura 5.6).

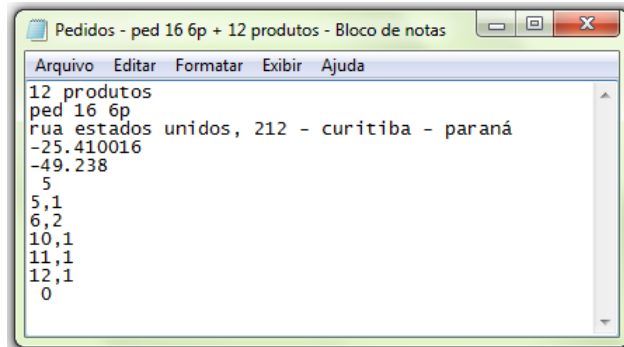


FIGURA 5.6 – EXEMPLO DE ARQUIVO DE UM PEDIDO SALVO NA BASE DE DADOS.

Fonte: O Autor

Cabe ressaltar que quando o usuário cadastra o endereço de entrega de um pedido, o software acessa a internet, mais especificamente o site www.maps.google.com.br, e extrai as informações de latitude e longitude deste endereço. Estas informações serão utilizadas no cálculo das distâncias entre os endereços de entrega e o endereço do Centro de Distribuição.

5.2 SELEÇÃO DOS PEDIDOS E RESOLUÇÃO DO PROBLEMA

Após realizar todos os cadastros do cenário fictício na primeira etapa da implementação, a segunda etapa apresenta ao usuário um software onde este poderá escolher como deseja resolver o problema de entregas (Figura 5.7). Este software apresenta em sua interface três comandos.

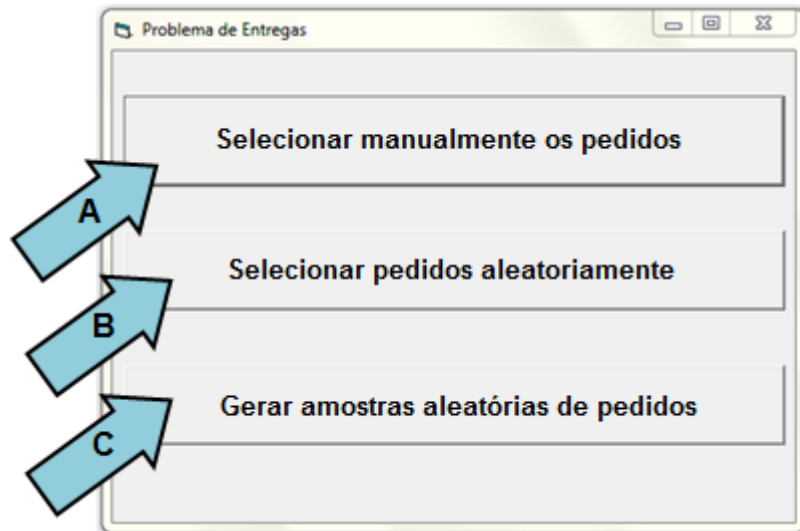


FIGURA 5.7 – INTERFACE DO SOFTWARE DE RESOLUÇÃO DE PROBLEMA DE ENTREGAS.
Fonte: O Autor

No primeiro comando (Botão **A**) o usuário tem liberdade para escolher quantos pedidos serão enviados ao Centro de Distribuição e selecionar manualmente no banco de dados os pedidos que deseja. Desta forma o usuário informa ao software todos os dados do problema de entregas específico que deseja resolver.

No segundo comando (Botão **B**) o usuário informa ao software apenas quantos pedidos deverão ser selecionados do banco de dados e enviados ao Centro de Distribuição. O software seleciona estes pedidos do banco de dados de maneira aleatória e resolve o problema de entregas para estes pedidos selecionados. A escolha aleatória dos pedidos evita que o usuário utilize amostras tendenciosas.

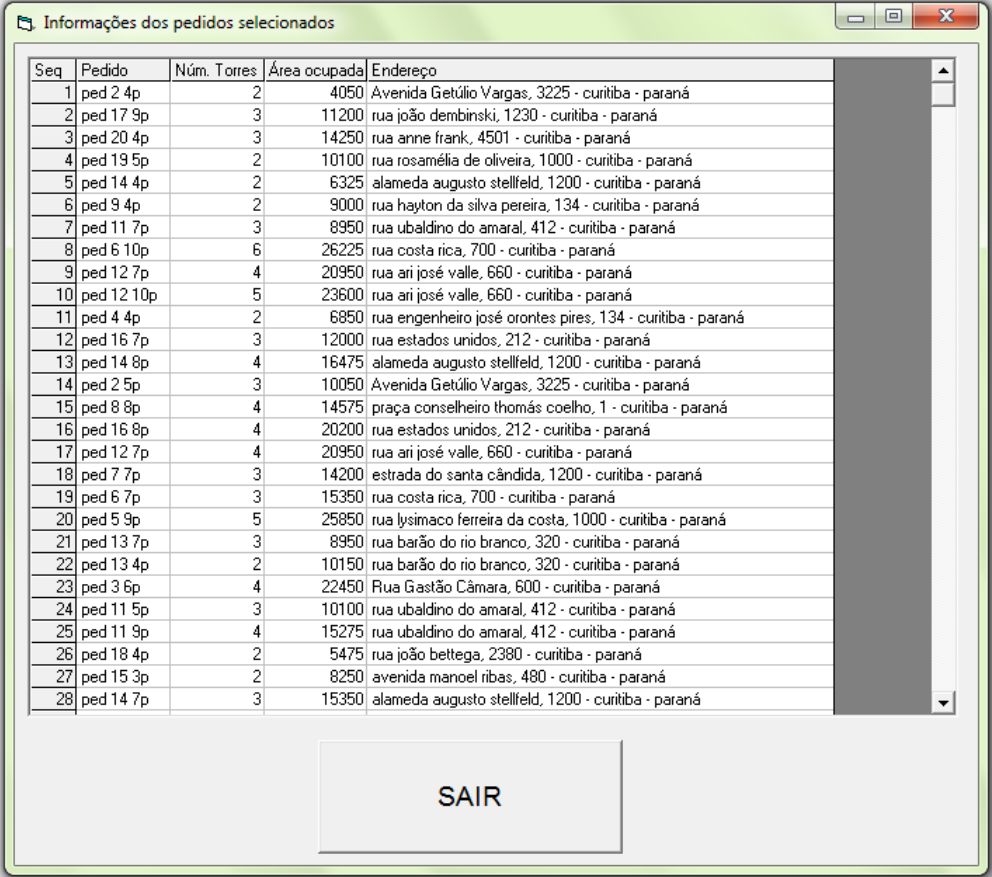
O terceiro comando (Botão **C**) apresenta uma função muito semelhante ao segundo comando. Neste caso o usuário também informa ao software a quantidade de pedidos que deverá ser selecionados aleatoriamente, mas a diferença é que o usuário também informa quantas amostras aleatórias com esta quantidade deverão ser formadas. Logo, utilizando este comando o usuário faz com que o software resolva não um problema de entregas, mas sim vários problemas de entregas diferentes com a mesma quantidade de pedidos selecionados.

Com o intuito de evitar a escolha de amostras tendenciosas de pedidos, na grande maioria (em torno de 90%) dos testes realizados, utilizou-se amostras aleatórias de pedidos.

Para exemplificar esta situação foi realizado um teste com uma amostra aleatória de 50 pedidos. O software então seleciona 50 pedidos dentre os 160

existentes no banco de dados. Para cada pedido selecionado, constroem-se as torres de caixas. Após a seleção de todos os produtos, forma-se uma solução inicial aleatória e, a partir desta, uma família de soluções iniciais. Com esta família de soluções dá-se início ao AG que otimizará a solução até que o critério de parada seja satisfeito. Ao final do AG, o melhor indivíduo da última geração de soluções será escolhido como resposta para o problema de entregas.

Após resolver o problema de entregas, ainda nesta etapa da implementação, o software apresenta algumas informações sobre o problema resolvido. Primeiro o software apresenta uma planilha contendo as seguintes informações dos pedidos selecionados: nome do arquivo, número de torres, área ocupada pelas bases das torres e endereço de entrega (Figura 5.8).



Seq	Pedido	Núm. Torres	Área ocupada	Endereço
1	ped 2 4p	2	4050	Avenida Getúlio Vargas, 3225 - curitiba - paraná
2	ped 17 9p	3	11200	rua joão dembinski, 1230 - curitiba - paraná
3	ped 20 4p	3	14250	rua anne frank, 4501 - curitiba - paraná
4	ped 19 5p	2	10100	rua rosamélia de oliveira, 1000 - curitiba - paraná
5	ped 14 4p	2	6325	alameda augusto stellfeld, 1200 - curitiba - paraná
6	ped 9 4p	2	9000	rua hayton da silva pereira, 134 - curitiba - paraná
7	ped 11 7p	3	8950	rua ubaldino do amaral, 412 - curitiba - paraná
8	ped 6 10p	6	26225	rua costa rica, 700 - curitiba - paraná
9	ped 12 7p	4	20950	rua ari josé valle, 660 - curitiba - paraná
10	ped 12 10p	5	23600	rua ari josé valle, 660 - curitiba - paraná
11	ped 4 4p	2	6850	rua engenheiro josé orontes pires, 134 - curitiba - paraná
12	ped 16 7p	3	12000	rua estados unidos, 212 - curitiba - paraná
13	ped 14 8p	4	16475	alameda augusto stellfeld, 1200 - curitiba - paraná
14	ped 2 5p	3	10050	Avenida Getúlio Vargas, 3225 - curitiba - paraná
15	ped 8 8p	4	14575	praça conselheiro thomás coelho, 1 - curitiba - paraná
16	ped 16 8p	4	20200	rua estados unidos, 212 - curitiba - paraná
17	ped 12 7p	4	20950	rua ari josé valle, 660 - curitiba - paraná
18	ped 7 7p	3	14200	estrada do santa cândida, 1200 - curitiba - paraná
19	ped 6 7p	3	15350	rua costa rica, 700 - curitiba - paraná
20	ped 5 9p	5	25850	rua lysimaco ferreira da costa, 1000 - curitiba - paraná
21	ped 13 7p	3	8950	rua barão do rio branco, 320 - curitiba - paraná
22	ped 13 4p	2	10150	rua barão do rio branco, 320 - curitiba - paraná
23	ped 3 6p	4	22450	Rua Gastão Câmara, 600 - curitiba - paraná
24	ped 11 5p	3	10100	rua ubaldino do amaral, 412 - curitiba - paraná
25	ped 11 9p	4	15275	rua ubaldino do amaral, 412 - curitiba - paraná
26	ped 18 4p	2	5475	rua joão bettega, 2380 - curitiba - paraná
27	ped 15 3p	2	8250	avenida manóel ribas, 480 - curitiba - paraná
28	ped 14 7p	3	15350	alameda augusto stellfeld, 1200 - curitiba - paraná

FIGURA 5.8 – PLANILHA COM INFORMAÇÕES DOS PEDIDOS SELECIONADOS.

Fonte: O Autor

Em seguida, o software apresenta uma janela contendo informações sobre a resolução do problema: o número de gerações realizadas, a distância total percorrida pela melhor solução encontrada e o tempo computacional gasto pelo software durante a resolução (Figura 5.9).

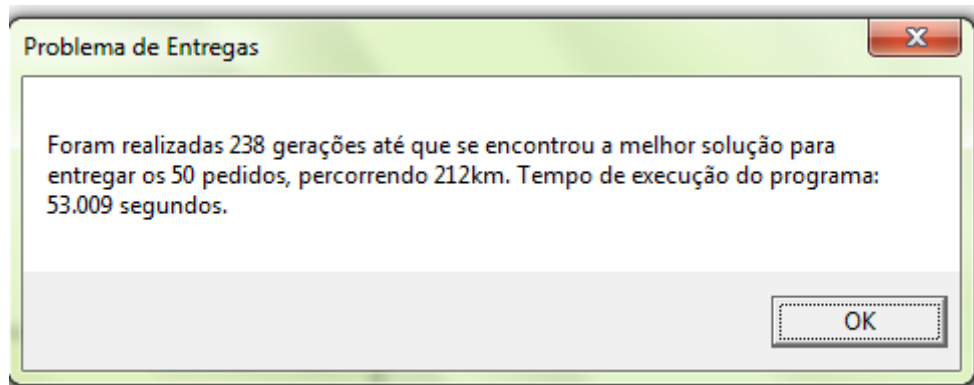


FIGURA 5.9 – JANELA COM INFORMAÇÕES DA RESOLUÇÃO DO PROBLEMA.
Fonte: O Autor

Além destas informações apresentadas diretamente ao usuário, o software gera dois arquivos de texto contendo informações sobre a resolução do problema de entregas e a solução do mesmo. Estes arquivos foram utilizados para apresentar a melhoria nos resultados obtidos que serão demonstrados no capítulo 6 da presente dissertação.

O primeiro arquivo gerado, denominado pelo autor de “Solução simples”, contém somente a distância total percorrida pela melhor solução de cada geração e a soma das distâncias totais percorridas pelas soluções da elite (50% dos melhores indivíduos da população) de cada geração (Figura 5.10).

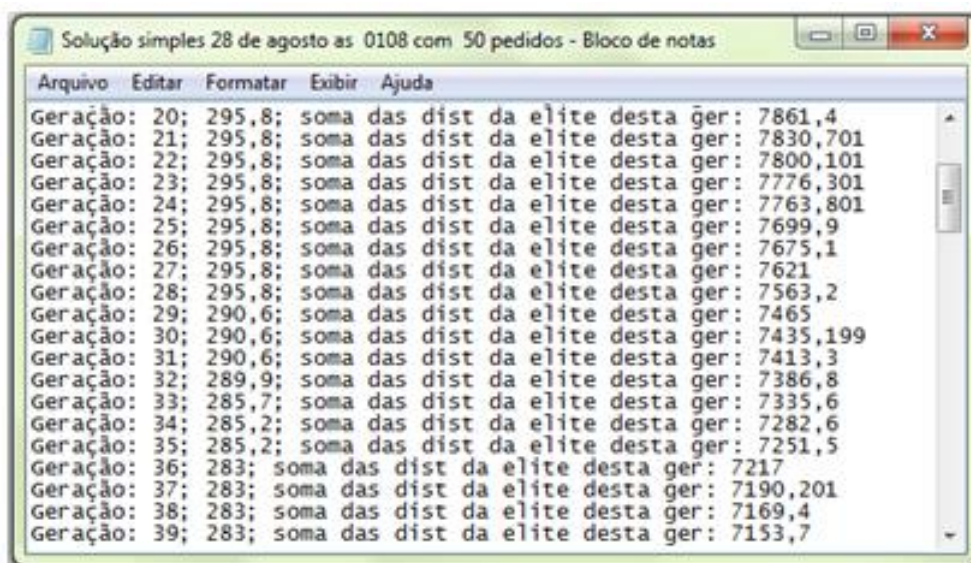


FIGURA 5.10 – ARQUIVO COM RESPOSTA SIMPLES DO PROBLEMA DE ENTREGAS.
Fonte: O Autor

O segundo arquivo gerado é mais completo que o primeiro. Este arquivo contém o tempo computacional para a resolução do problema, o número de gerações necessárias durante o AG até que o critério de parada fosse atingido e a distância total percorrida pela solução encontrada pelo software. Além disso, o arquivo apresenta uma série de informações sobre os três melhores indivíduos de todas as gerações realizadas, com exceção da última geração. No caso da última geração realizada pelo AG o arquivo possui as informações de todos os indivíduos da população desta geração. Entre estas informações sobre os indivíduos estão: a quantidade de veículos necessária, quais pedidos serão atendidos por cada um dos veículos e qual a distância percorrida por cada um dos veículos (Figura 5.11).

```

Solução do dia 28 de agosto as 0108 com 50caixas - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
50
28 de agosto
0108

Tempo de solução: 53.009

238 gerações

MELHOR SOLUÇÃO
212km.

Container: 1
26,41,23,34,6,8,12,3.
Dist perc pelo container: 43.8

Container: 2
15,16,7,38,46,47,45,21,32,29.
Dist perc pelo container: 30.1

Container: 3
4,13,5,37,36,42,25,22,33.
Dist perc pelo container: 25.9

Container: 4
2,10,35,18,44,31,11,17.
Dist perc pelo container: 49.3

Container: 5
50,14,24,19,20,48,1,30.
Dist perc pelo container: 30.6

Container: 6
9,27,39,28,40,49,43.
Dist perc pelo container: 32.3

```

FIGURA 5.11 – ARQUIVO COM INFORMAÇÕES COMPLETAS SOBRE A RESOLUÇÃO DO PROBLEMA DE ENTREGAS.

Fonte: O Autor

5.3 APRESENTAÇÃO DAS TORRES DE UM DETERMINADO PEDIDO E CONSTRUÇÃO DAS INFORMAÇÕES ESPECÍFICAS POR SETOR

Finalizada a etapa de resolução do problema de entregas o usuário passa então a uma etapa de construção de respostas do problema. Nesta etapa o usuário pode optar por dois comandos que apresentarão informações sobre o problema resolvido (Figura 5.12).

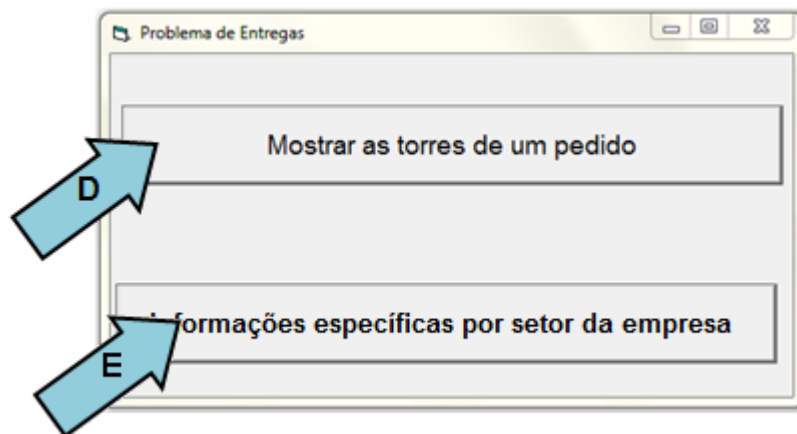


FIGURA 5.12 – INTERFACE DO SOFTWARE PARA CONSTRUÇÃO DAS RESPOSTAS.
Fonte: O Autor

Selecionando o primeiro comando (Botão **D**) o usuário pode visualizar a sequência de formação das torres de produtos de um determinado pedido (Figura 5.13). O software apresenta ao usuário uma janela onde este escolhe de qual pedido gostaria de visualizar as torres (a). Com esta informação o software primeiramente apresenta a quantidade de torres deste pedido selecionado pelo usuário (b) e depois apresenta uma planilha contendo a sequência de formação (empilhamento dos produtos) de cada uma destas torres (c).

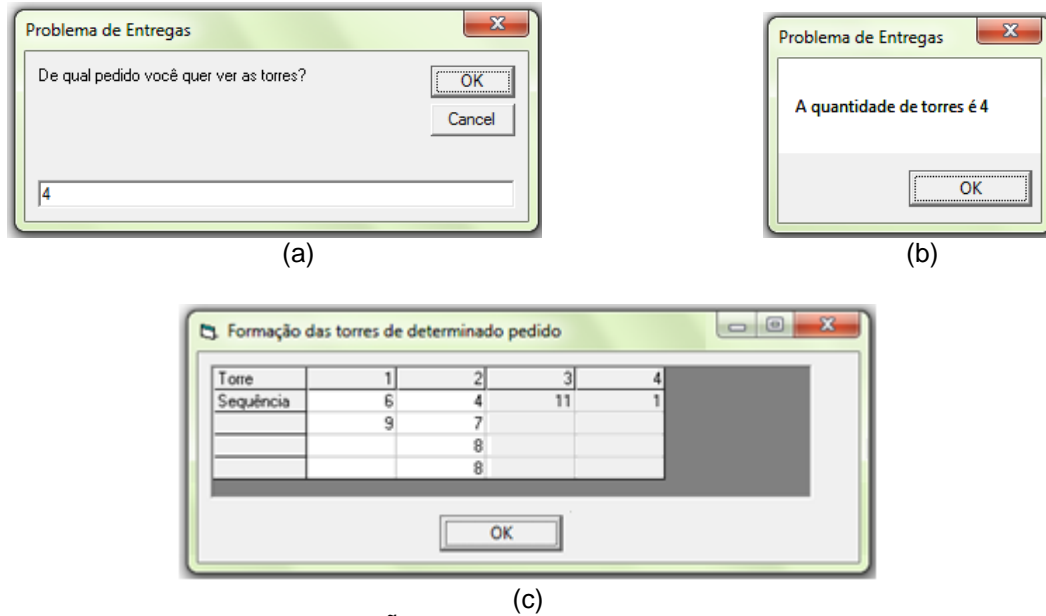


FIGURA 5.13 – VISUALIZAÇÃO DAS TORRES DE UM DETERMINADO PEDIDO.

Fonte: O Autor

Já o segundo comando do software (Botão **E**) gera para o usuário uma série de arquivos com informações específicas para cada um dos setores da empresa envolvidos no problema de entregas (páginas 51 e 52).

O primeiro setor envolvido no problema de entregas é o setor responsável pelo estoque e organização de mercadorias. Para estes funcionários o software gera um arquivo em texto que informa as quantidades e os tipos de produto que serão transportados por cada um dos veículos (Figura 5.14).

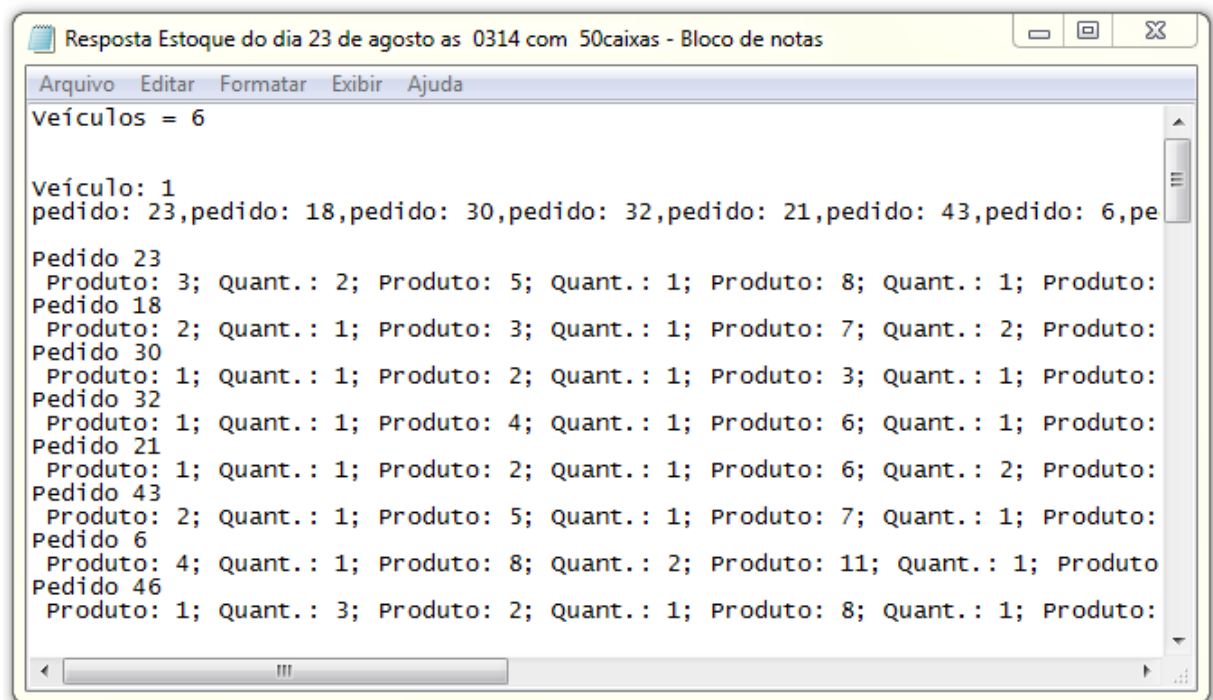
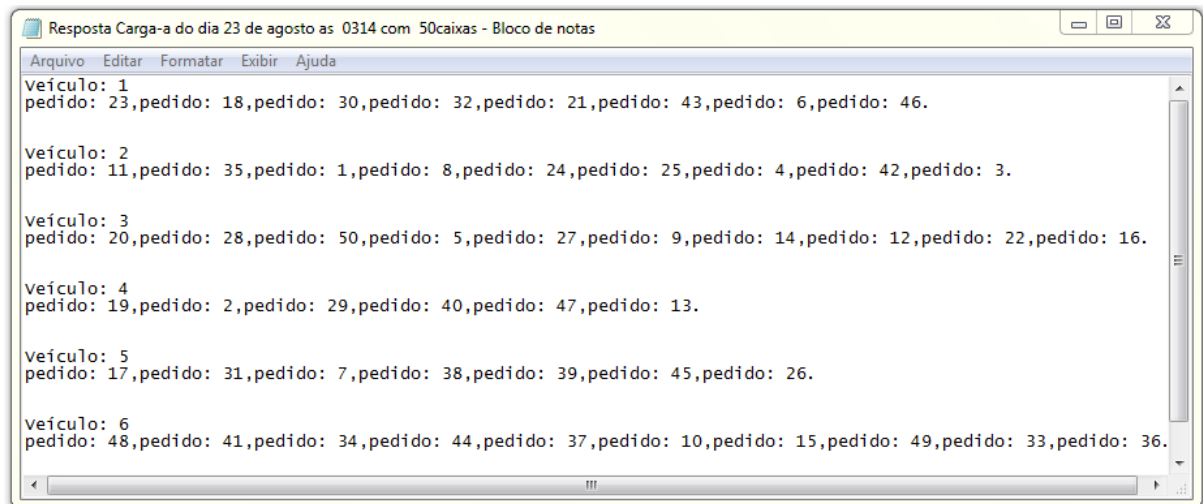


FIGURA 5.14 – EXEMPLO DE ARQUIVO COM AS INFORMAÇÕES PARA OS RESPONSÁVEIS PELO ESTOQUE E ORGANIZAÇÃO DE MERCADORIAS.

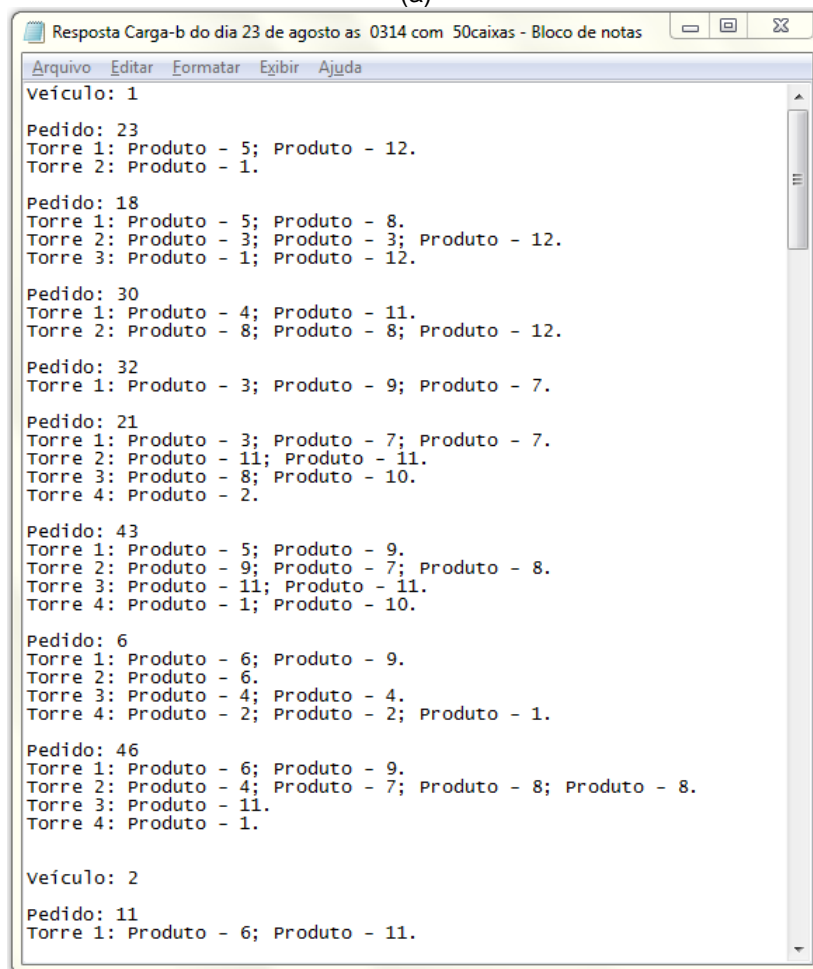
Fonte: O Autor

O segundo setor envolvido no problema de entregas é o setor responsável por realizar o carregamento dos produtos no interior dos veículos. Neste caso o software gera dois arquivos de texto (Figura 5.15): um arquivo que contém a sequência que se deve carregar os pedidos no interior do veículo (a) e um segundo arquivo que apresenta a formação das torres¹ de produtos de todos os pedidos que serão carregados em cada um dos veículos (b).

¹ Note que um produto pode aparecer na sequência de formação de torres com o seu número de cadastro adicionado de 1000 unidades (Exemplo: 1001) significa que este produto deve ser carregado girado na torre, isto é, a largura deve ser carregada como profundidade e vice-versa.



(a)



(b)

FIGURA 5.15 – EXEMPLO DE ARQUIVOS COM AS INFORMAÇÕES ENVIADAS AOS RESPONSÁVEIS POR CARREGAR OS VEÍCULOS.

Fonte: O Autor

O terceiro setor da empresa envolvido diretamente no problema de entregas é o setor responsável justamente por realizar as entregas aos clientes. Para simplificar, na presente dissertação vamos imaginar este setor centralizado na figura

do motorista do veículo de entrega. Cada um dos motoristas dos veículos que farão as entregas receberá dois arquivos gerados pelo software. O primeiro arquivo é um mapa gerado pelo site www.maps.google.com.br. Ao finalizar a resolução do problema de entregas o software mais uma vez acessa a internet e utiliza o site citado para construir mapas com o roteamento que cada veículo deve seguir para realizar suas entregas (Figura 5.16).

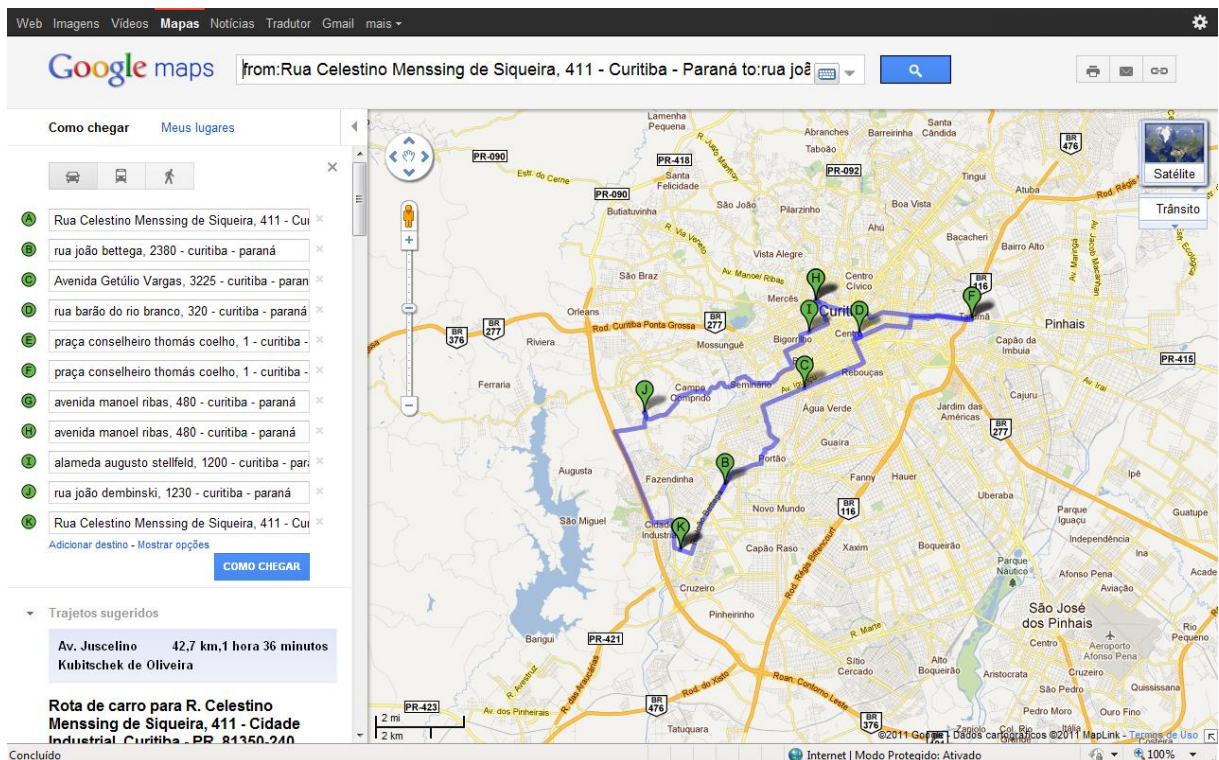


FIGURA 5.16 – EXEMPLO DE MAPA COM O ROTEAMENTO DE ENTREGA, ENVIADO AO MOTORISTA DO VEÍCULO RESPONSÁVEL POR ESTAS ENTREGAS.

Fonte: O Autor

O segundo arquivo é muito parecido com o arquivo enviado ao setor responsável pelo estoque e organização de mercadorias, informando as quantidades e os tipos de produto de cada pedido que seu veículo transporta, de forma a facilitar na hora de descarregar os produtos nos endereços dos respectivos clientes (Figura 5.17).

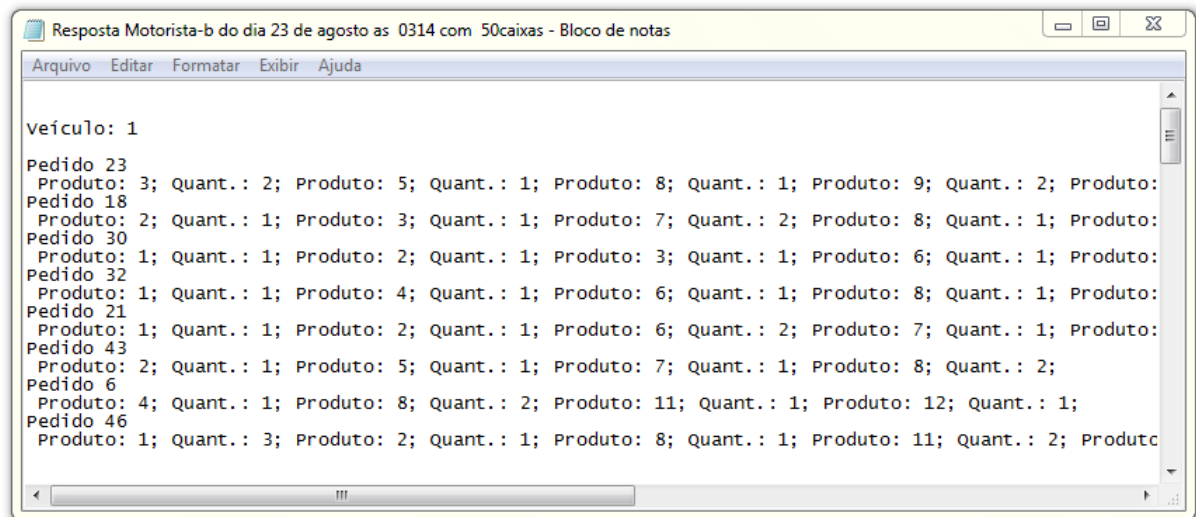


FIGURA 5.17 – EXEMPLO DE ARQUIVO FORNECIDO AOS MOTORISTAS CONTENDO AS INFORMAÇÕES DOS PEDIDOS TRANSPORTADOS POR SEU VEÍCULO.

Fonte: O Autor

Além destes três setores, que estão diretamente ligados ao carregamento e entrega das mercadorias, o software desenvolvido apresenta também um arquivo com informações resumidas sobre a solução do problema para o setor responsável por fazer a controladoria/auditoria da empresa. Os funcionários deste setor receberão um arquivo síntese que apresenta a quantidade de veículos utilizados, a distância aproximada percorrida por cada um dos veículos e os pedidos que cada veículo deve transportar (Figura 5.18). Cabe ressaltar que os funcionários responsáveis pela controladoria/auditoria da empresa terão acesso aos arquivos gerados para cada um dos outros setores citados.

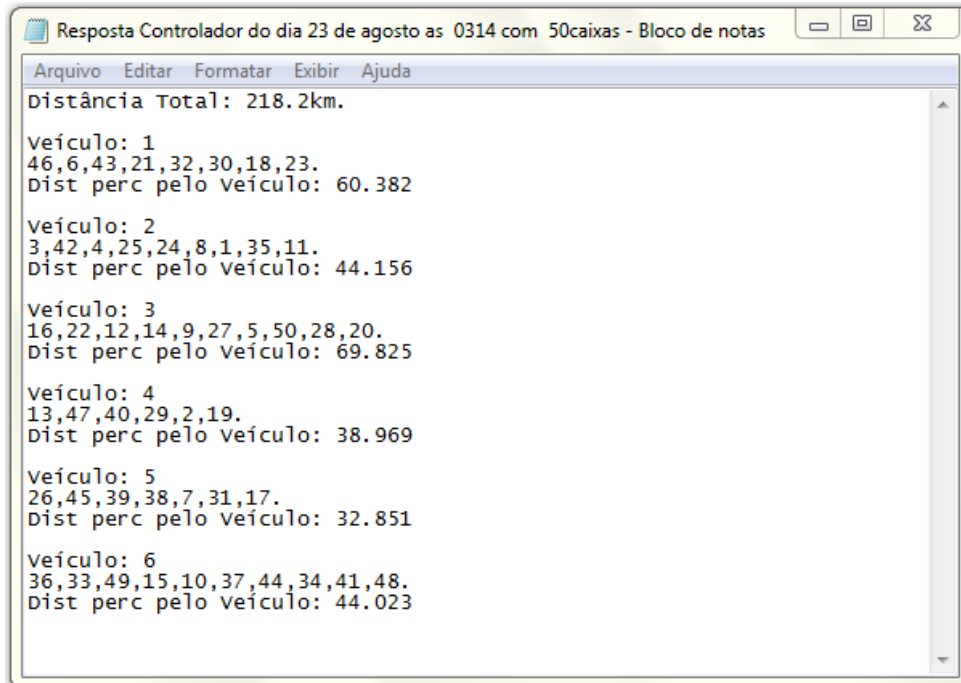


FIGURA 5.18 – EXEMPLO DE ARQUIVO SÍNTESE CONTENDO AS INFORMAÇÕES PARA O SETOR DE CONTROLADORIA/AUDITORIA DA EMPRESA.

Fonte: O Autor

Todas as notações de produtos de cada um dos pedidos se referem ao arquivo da base de dados que contém o cadastro de todos os produtos existentes na empresa do cenário fictício.

O próximo capítulo apresentará alguns dos resultados obtidos após diversos testes realizados com este software desenvolvido, utilizando o cenário fictício, com amostras entre 50 e 350 pedidos.

6 RESULTADOS COMPUTACIONAIS

Este capítulo da presente dissertação apresenta uma série de resultados encontrados após vários testes realizados. Nestes testes foram utilizados os dados do cenário fictício citado na dissertação, com amostras contendo cinco quantidades de pedidos diferentes: 50 pedidos, 75 pedidos, 140 pedidos, 200 pedidos e 350 pedidos. Para cada cenário escolhido foram realizados diversos testes com amostras aleatórias diferentes e, a partir destes testes, construiu-se uma situação média para cada um destes cenários. Os quadros e as tabelas a seguir mostram o comportamento destes cenários e destas amostras.

Os dois quadros abaixo apresentam o comportamento de cinco amostras, uma de cada cenário diferente, nas primeiras 250 gerações do AG.

Cenário	50 pedidos	75 pedidos	140 pedidos	200 pedidos	350 pedidos
Geração	Distância Total Percorrida pela Melhor Solução	Distância Total Percorrida pela Melhor Solução	Distância Total Percorrida pela Melhor Solução	Distância Total Percorrida pela Melhor Solução	Distância Total Percorrida pela Melhor Solução
1	405,2 km	521,2 km	1067,7 km	1578,5 km	2759,0 km
25	331,1 km	443,6 km	1000,0 km	1496,4 km	2603,3 km
50	294,6 km	405,4 km	926,4 km	1428,5 km	2505,1 km
75	277,7 km	387,2 km	865,1 km	1362,7 km	2455,0 km
100	269,3 km	370,9 km	827,6 km	1296,1 km	2356,0 km
125	256,3 km	349,4 km	801,4 km	1247,4 km	2295,2 km
150	250,0 km	341,7 km	784,5 km	1204,9 km	2239,9 km
175	239,9 km	341,3 km	762,3 km	1165,3 km	2198,1 km
200	238,8 km	317,0 km	755,2 km	1132,0 km	2151,7 km
225	231,5 km	310,7 km	736,5 km	1110,2 km	2107,5 km
250	230,0 km	310,2 km	728,5 km	1079,1 km	2067,0 km

QUADRO 6.1 – RESUMO DO COMPORTAMENTO DA DISTÂNCIA TOTAL PERCORRIDA PELA MELHOR SOLUÇÃO A CADA GERAÇÃO, EM 250 GERAÇÕES, PARA UMA AMOSTRA ALEATÓRIA DE CADA CENÁRIO ESCOLHIDO.

Fonte: O Autor

Cenário	50 pedidos	75 pedidos	140 pedidos	200 pedidos	350 pedidos
Geração	Distância Total Percorrida pela Elite da População	Distância Total Percorrida pela Elite da População	Distância Total Percorrida pela Elite da População	Distância Total Percorrida pela Elite da População	Distância Total Percorrida pela Elite da População
1	10932,1 km	15997,1 km	39475,9 km	105627,6 km	255503,4 km
25	8766,4 km	13639,3 km	36971,6 km	98562,4 km	237653,1 km
50	7495,7 km	12381,1 km	33966,7 km	92887,9 km	229062,4 km
75	7041,0 km	11805,2 km	31757,8 km	88853,9 km	222357,8 km
100	6779,2 km	11187,0 km	30045,9 km	84303,5 km	214031,1 km

125	6448,5 km	10695,8 km	29039,8 km	80802,8 km	208551,2 km
150	6250,0 km	10255,5 km	28326,9 km	78390,4 km	203223,3 km
175	5999,1 km	10250,6 km	27606,1 km	75014,0 km	199142,0 km
200	5970,0 km	9568,8 km	27287,0 km	73375,4 km	195190,8 km
225	5826,6 km	9494,4 km	26575,7 km	71502,2 km	190504,3 km
250	5755,6 km	9326,6 km	26362,7 km	69618,5 km	187783,8 km

QUADRO 6.2 – RESUMO DO COMPORTAMENTO DA DISTÂNCIA TOTAL PERCORRIDA PELA ELITE DA POPULAÇÃO (50% DAS MELHORES RESPOSTAS DA POPULAÇÃO) A CADA GERAÇÃO, EM 250 GERAÇÕES, PARA UMA AMOSTRA ALEATÓRIA DE CADA CENÁRIO ESCOLHIDO.

Fonte: O Autor

O gráfico abaixo representa a evolução das melhores soluções de cinco amostras aleatórias selecionadas uma para cada um dos cenários escolhidos.

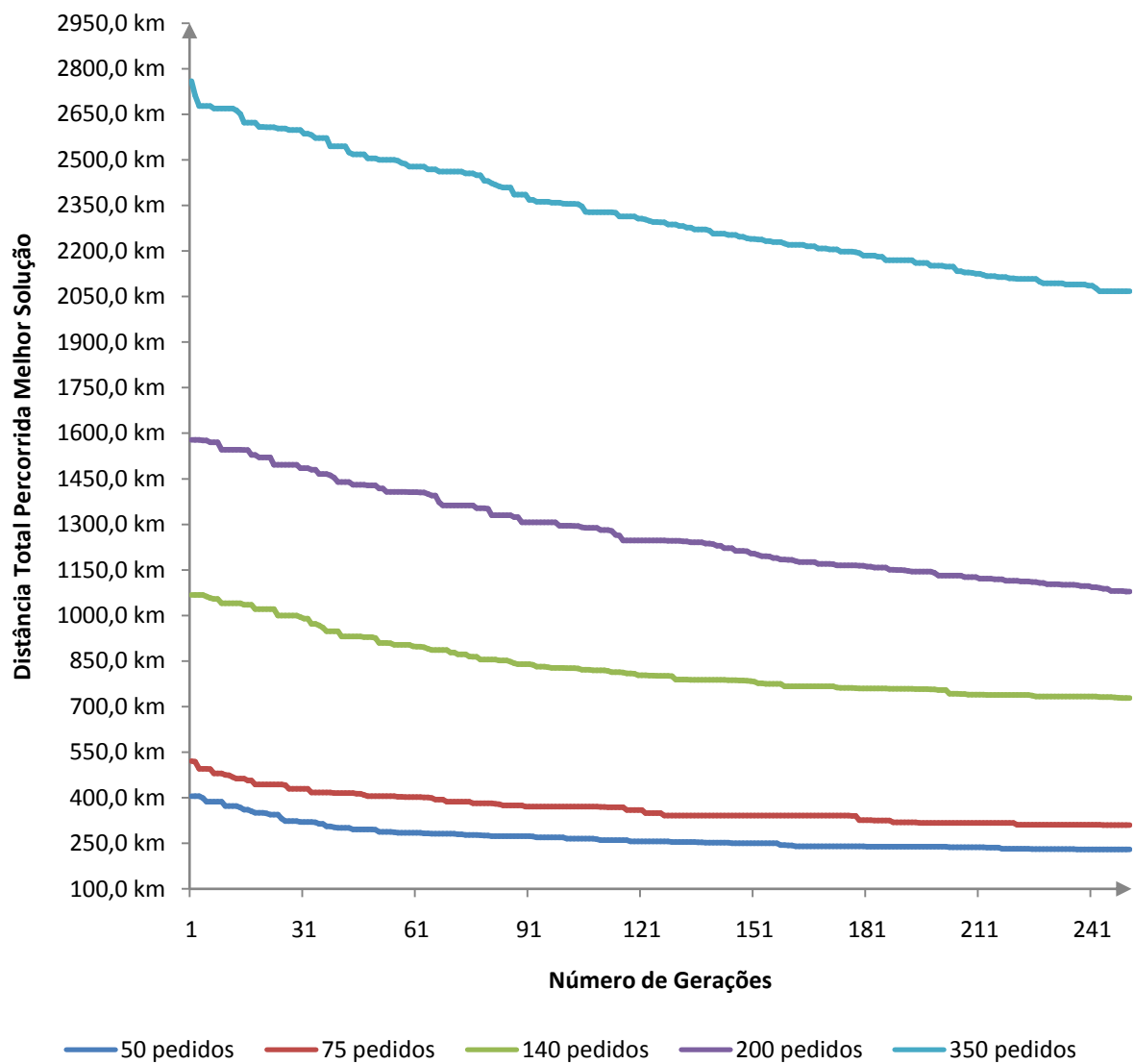


GRÁFICO 6.1 – COMPORTAMENTO DAS MELHORES SOLUÇÕES EM 250 GERAÇÕES PARA CINCO AMOSTRAS ESCOLHIDAS ALEATORIAMENTE, UMA PARA CADA CENÁRIO ESCOLHIDO.

Fonte: O Autor

O gráfico abaixo representa a evolução da soma das melhores soluções (elite da população) de cinco amostras aleatórias selecionadas uma para cada um dos cenários escolhidos.

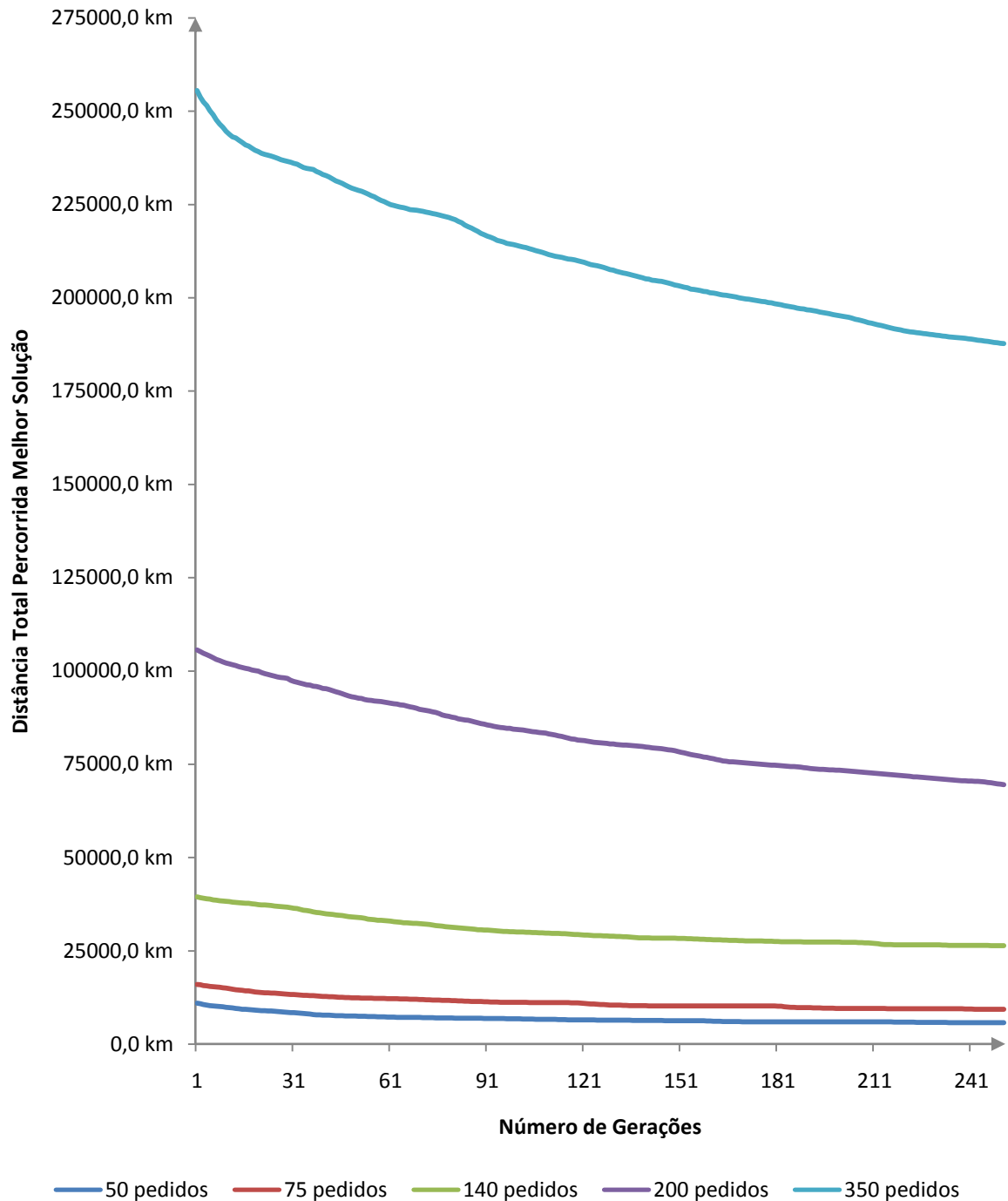


GRÁFICO 6.2 – COMPORTAMENTO DAS SOLUÇÕES DA ELITE DA POPULAÇÃO EM 250 GERAÇÕES PARA CINCO AMOSTRAS ESCOLHIDAS ALEATORIAMENTE, UMA PARA CADA CENÁRIO ESCOLHIDO.

Fonte: O Autor

Os próximos dois gráficos reúnem as informações contidas nos gráficos já mostrados, porém trata a evolução de maneira percentual para poder fazer uma melhor comparação entre estes dados. O primeiro dos dois gráficos trata da evolução percentual das melhores soluções das cinco amostras selecionadas aleatoriamente de cada um dos cenários escolhidos nas primeiras 250 gerações, enquanto o segundo gráfico trata da evolução percentual da soma das melhores soluções (elite da população) das cinco amostras selecionadas aleatoriamente de cada um dos cenários escolhidos, também nas primeiras 250 gerações.

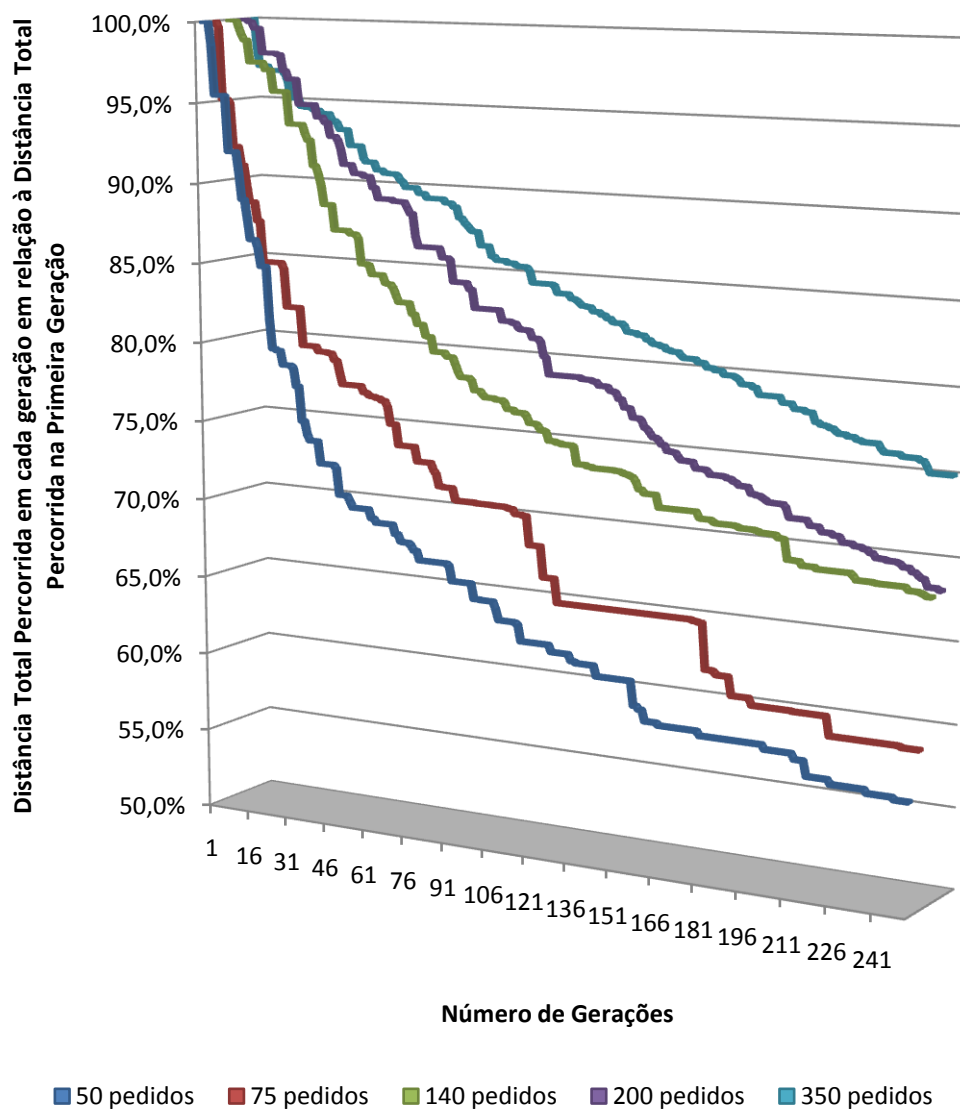


GRÁFICO 6.3 – COMPORTAMENTO PERCENTUAL DAS MELHORES SOLUÇÕES EM 250 GERAÇÕES PARA CINCO AMOSTRAS ESCOLHIDAS ALEATORIAMENTE, UMA PARA CADA CENÁRIO ESCOLHIDO.

Fonte: O Autor

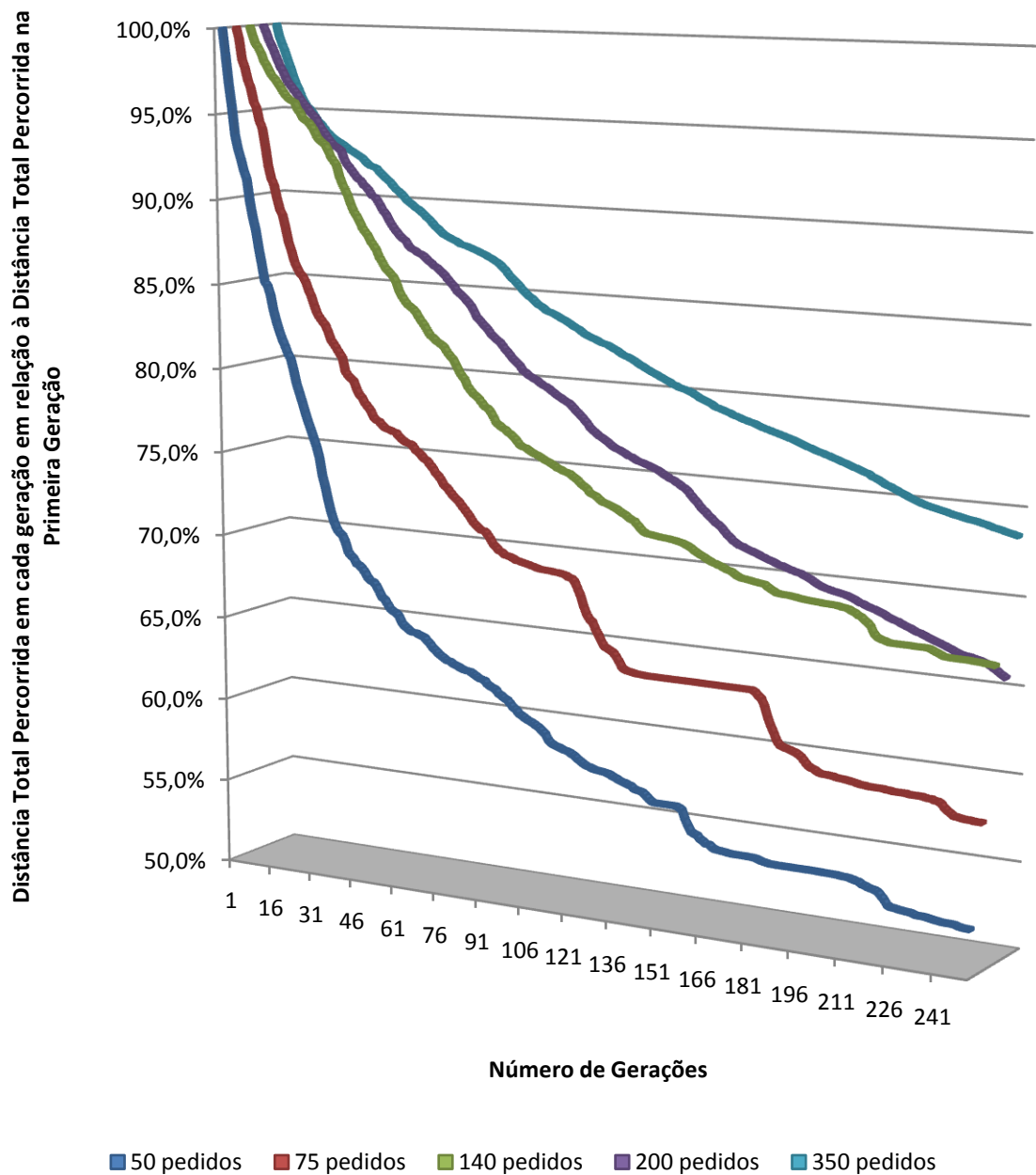


GRÁFICO 6.4 – COMPORTAMENTO PERCENTUAL DAS SOLUÇÕES DA ELITE DA POPULAÇÃO EM 250 GERAÇÕES PARA CINCO AMOSTRAS ESCOLHIDAS ALEATORIAMENTE, UMA PARA CADA CENÁRIO ESCOLHIDO.

Fonte: O Autor

Baseado no comportamento evolutivo da média da melhor solução, tem-se o gráfico abaixo onde se faz um comparativo mais simples, levando em conta apenas a diferença entre o melhor resultado encontrado entre a família de soluções iniciais aleatórias e o melhor resultado encontrado após a realização do AG.

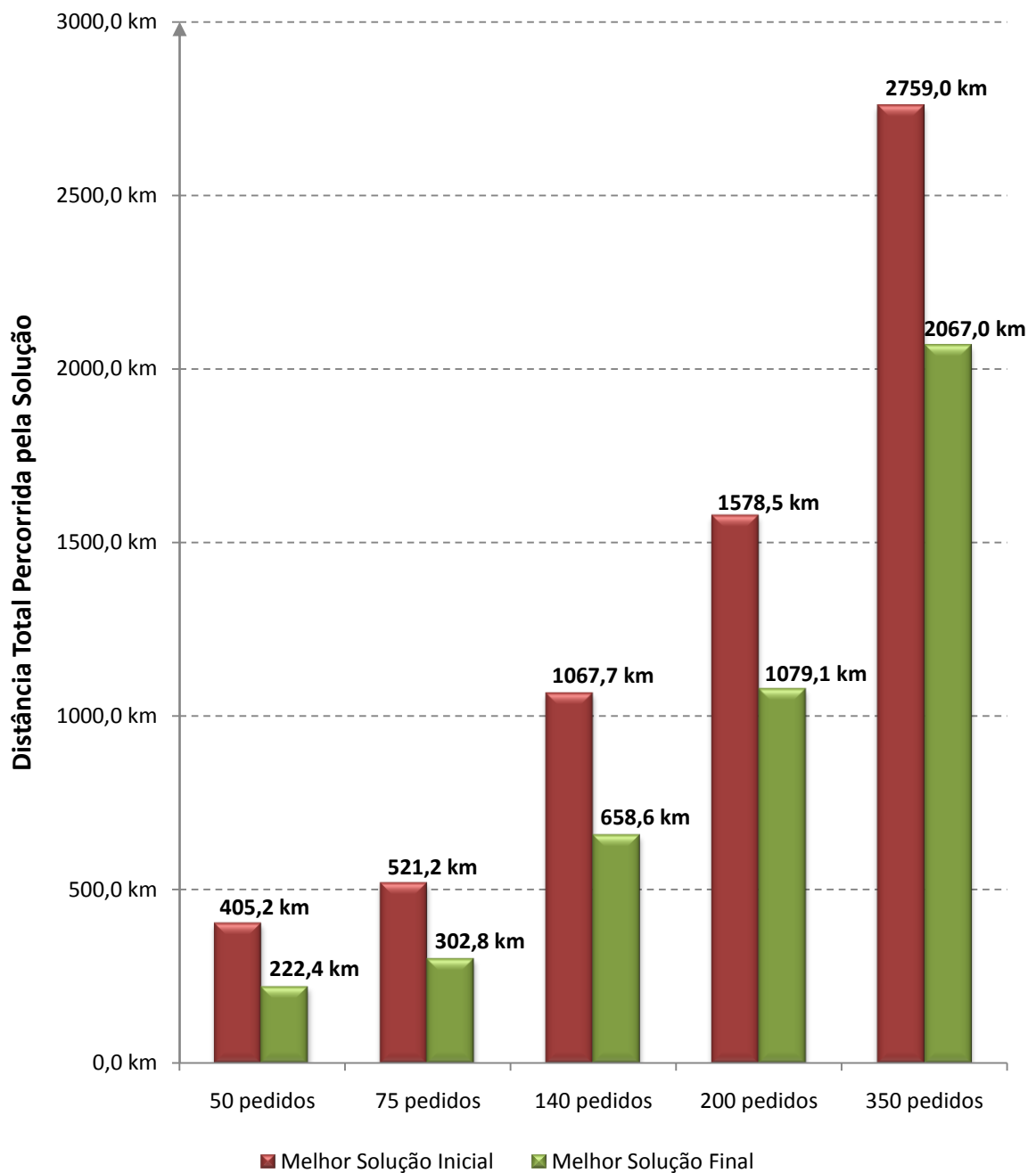


GRÁFICO 6.5 – COMPARAÇÃO ENTRE A MELHOR SOLUÇÃO INICIAL E A MELHOR SOLUÇÃO FINAL, EM MÉDIA, PARA CADA CENÁRIO ESCOLHIDO ENTRE 50 E 350 PEDIDOS.

Fonte: O Autor

Acompanhando a ideia do gráfico acima é possível construir um gráfico simples, mostrando a melhoria percentual média na solução encontrada para cada tamanho de amostra.

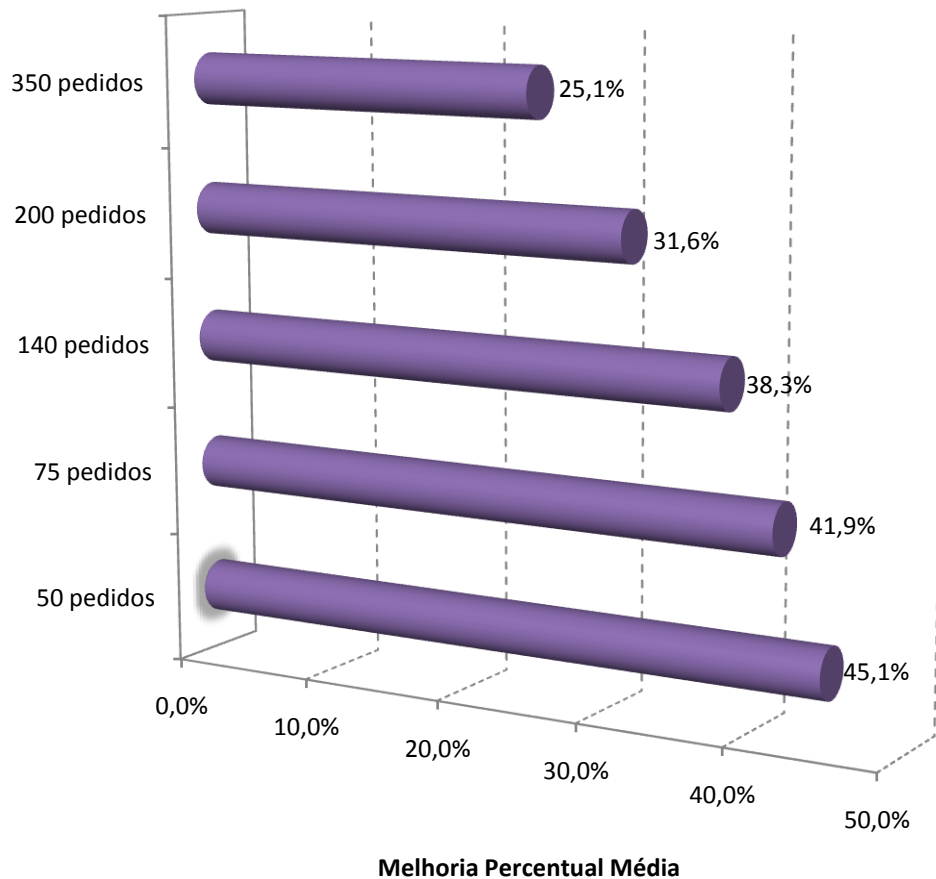


GRÁFICO 6.6 – MELHORIA PERCENTUAL MÉDIA PARA CADA CENÁRIO ESCOLHIDO ENTRE 50 E 350 PEDIDOS.
Fonte: O Autor

Além dos resultados apresentados acima, referentes às distâncias percorridas pelos veículos para realizar as entregas, foi construída uma análise sobre o número de veículos necessários para realização das entregas e também sobre a área média ocupada no piso dos veículos. Cabe ressaltar que não se faz necessário a avaliação do volume ocupado no container, já que as caixas estão dispostas em torres e a cada geração tende a ocorrer uma melhoria na disposição destas torres dentro do container, podendo assim aumentar a área ocupada pelas torres no piso do compartimento de carga do veículo e também podendo até diminuir o número de veículos utilizados para realizar as entregas, como se vê nos gráficos abaixo.

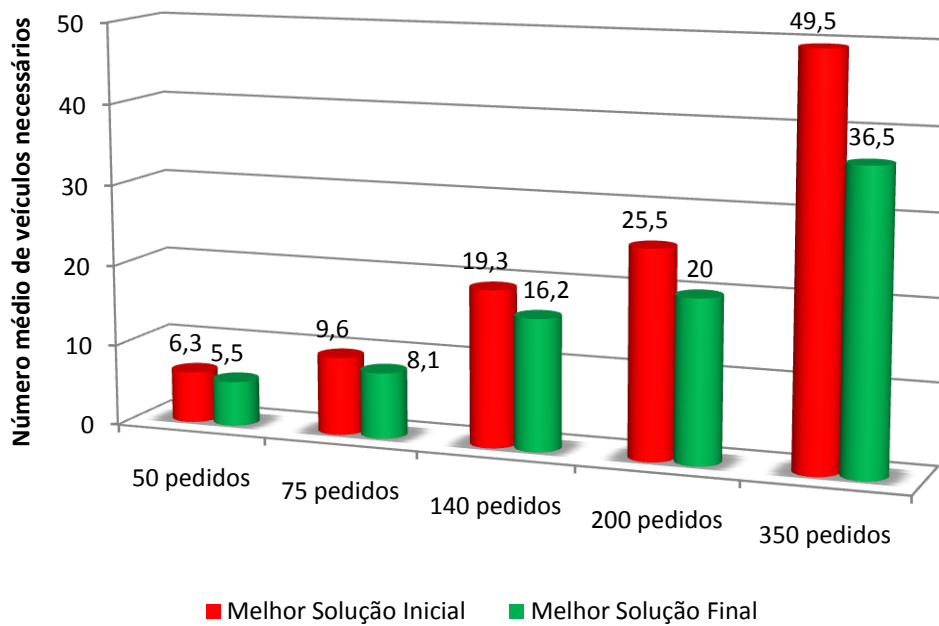


GRÁFICO 6.7 – COMPARAÇÃO ENTRE A QUANTIDADE DE VEÍCULOS NECESSÁRIA PARA A MELHOR SOLUÇÃO INICIAL E FINAL, EM MÉDIA, PARA CADA CENÁRIO ESCOLHIDO ENTRE 50 E 350 PEDIDOS.

Fonte: O Autor

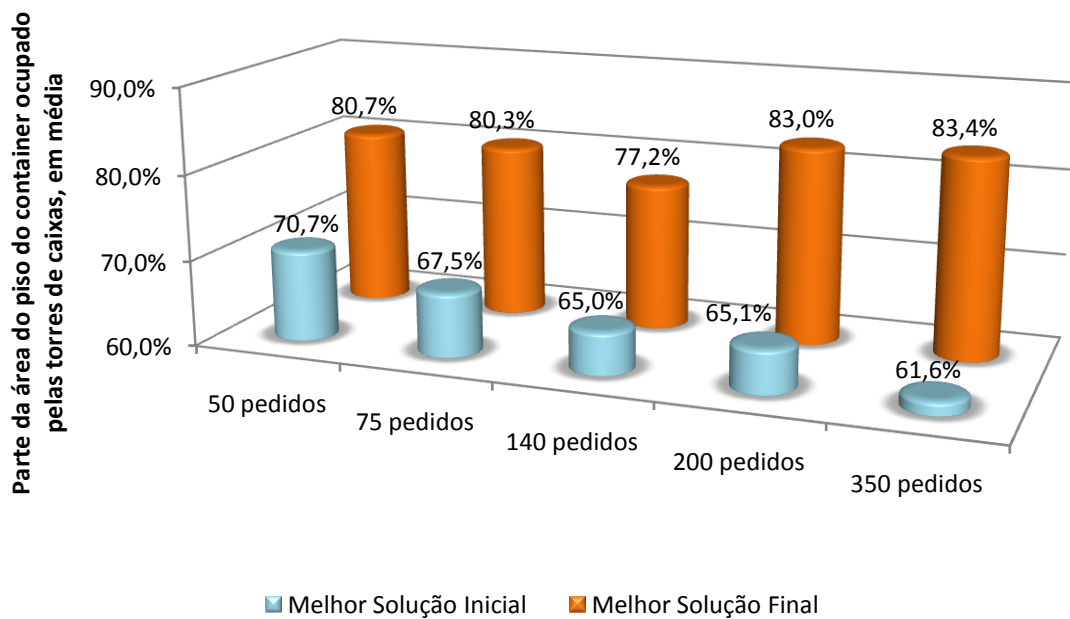


GRÁFICO 6.8 – COMPARAÇÃO ENTRE A ÁREA OCUPADA PELAS TORRES DE CAIXAS NO CONTAINER PARA A MELHOR SOLUÇÃO INICIAL E FINAL, EM MÉDIA, PARA CADA CENÁRIO ESCOLHIDO ENTRE 50 E 350 PEDIDOS.

Fonte: O Autor

De posse das respostas mostradas neste capítulo, o próximo capítulo da dissertação apresenta as conclusões obtidas pelo autor.

7 CONCLUSÕES

O problema de entrega de mercadorias envolve a integração dos problemas de carregamento de container (PCC) e de roteamento de veículos (PRV). É um problema muito comum na área da logística, tanto em empresas que prestam o serviço de entrega de encomendas quanto em grandes empresas varejistas que optam por não terceirizar este serviço. Estas grandes empresas geralmente possuem um Centro de Distribuição de Produtos (CD) e uma frota de veículos de carga para realizar estas entregas. Na presente dissertação foi considerada a segunda proposta de aplicação com um Centro de Distribuição.

O objetivo da dissertação foi desenvolver um método de resolução desta integração dos dois problemas, buscando uma solução com a melhor configuração de carregamento possível e, ao mesmo tempo, a menor distância percorrida pelos veículos que farão a entrega. Esta proposta foi construída montando torres com os produtos de cada um dos pedidos recebidos pela empresa e buscando a melhor sequência de carregamento destes diversos pedidos em vários caminhões. Isto com o objetivo de diminuir a distância total percorrida pelos veículos, consequentemente diminuindo o número de veículos utilizados e otimizando o acondicionamento destas torres de pedidos dentro dos veículos. Para carregar as torres nos veículos foi proposto um Algoritmo *Bottom-Left* (ABL) e para otimizar as soluções de maneira evolutiva foi utilizado um Algoritmo Genético (AG).

Os resultados apresentados pelo problema mostram melhoria média, em relação às soluções iniciais aleatórias, em torno de 25% a 45% na diminuição da distância total percorrida pelos veículos de carga, dependendo do tamanho da amostra e, chegando, em algumas amostras, até a 60% de diminuição. Em valores absolutos, isto representa uma economia média de mais de cem quilômetros para cenários com 50 pedidos até em torno de setecentos quilômetros para cenários com 350 pedidos. Esta economia pode ser medida financeiramente através do consumo de combustível ou ainda através da diminuição do tempo para realizar todas as entregas, podendo evitar que motoristas e entregadores necessitem receber pagamentos de horas extras.

Outro fato que se pode ressaltar é o software desenvolvido através da implementação computacional desta proposta de resolução do problema de entregas. Este apresentou diferentes respostas para os diferentes setores que

podem trabalhar no problema de entregas. A implementação mostrou como construir respostas de simples entendimento para:

i) o setor de Estoque e Organização de Mercadorias, facilitando assim o trabalho dos funcionários que devem direcionar cada produto que será carregado para o devido veículo que o transportará;

ii) o setor responsável pelo carregamento dos veículos, de forma que o funcionário que carregará o veículo i saiba quais os pedidos serão carregados neste veículo e em que ordem devem ser carregados, quais os produtos de cada pedido e como formar as torres de produtos para cada pedido;

iii) o setor responsável por fazer a entrega dos produtos (motorista de cada veículo) de modo que o mesmo saiba o menor percurso a percorrer seguindo a sequência de entregas e quais as caixas devem ser descarregadas em cada entrega.

Esta implementação ainda mostrou uma resposta sintética do problema que pode ser utilizado pelo setor de Controle para que possa conferir a distância total percorrida, as distâncias percorridas por cada veículo e quais pedidos foram transportados por cada veículo.

Os tempos de processamento do programa são plenamente aceitáveis, pois, para fins acadêmicos, foi escolhido um limite de tempo baixo. Para situações reais a empresa de entrega pode despende até de algumas horas de processamento durante parte da noite, de forma que até a manhã do próximo dia o programa tenha encontrado a solução e todos os pedidos já tenham sido carregados, já que raramente se faz entrega de produtos à pessoa física fora do 'horário comercial'.

Outra conclusão importante a ser ressaltada é que, pelos gráficos 6.7 e 6.8 apresentados nos resultados, aparentemente não há uma diminuição significativa na quantidade de veículos utilizados no decorrer das gerações, porém, deve-se ressaltar que os veículos possuem um compartimento de carga com dimensões significativamente grandes (240cm de largura por 600cm de profundidade). Se fossem utilizados veículos menores esta variação na quantidade de veículos seria mais representativa. Além do que, os testes mostraram que quanto maior a quantidade de pedidos a serem carregados, maior será a variação na quantidade de veículos necessários para realizar as entregas.

Com isso, pode-se dizer que a presente dissertação produziu uma contribuição de valor na discussão sobre a integração dos problemas de

carregamento de container e de roteamento de veículos, que embora seja recente no meio acadêmico possui aplicações práticas que necessitam cada dia mais de soluções eficientes.

7.1 SUGESTÕES PARA TRABALHOS FUTUROS

O campo de pesquisa que envolve a integração dos problemas PCC e PRV ainda necessita ser amplamente estudado em virtude da vastidão que possui. Da presente dissertação, pode-se extrair algumas ideias de pesquisas futuras que contribuiriam da seguinte forma:

a) optar por metodologias de carregamento diferentes utilizando paredes virtuais ou ainda cortes guilhotinados tridimensionais, de forma a otimizar o espaço ocupado dentro do container.

b) mudar o método de resolução do problema de corte bidimensional utilizando Algoritmos *Best-Fit* ou *Next-Fit*, buscando melhorar o encaixe das torres de produtos dentro do container.

c) substituir o método de otimização utilizado, AG, por outros métodos heurísticos como *Simulated Annealing* ou ainda Busca Tabu, a fim de comparar as soluções encontradas e os tempos computacionais necessários a cada método.

d) aprimorar o software desenvolvido de forma a apresentar soluções com maiores requintes visuais como visão tridimensional do carregamento, ou ainda sincronizar as soluções com aparelhos de GPS (*Global Position System*) acoplados a cada um dos veículos.

REFERÊNCIAS

- ARAUJO, O. B., **Problema de corte e empacotamento tridimensional e integração com roteamento de veículos**. Tese (Doutorado em Engenharia Elétrica) Universidade Estadual de Campinas. Campinas, p. 185. 2006.
- ARAUJO, O. B.; ARMENTANO, V. A., **A multi start random constructive heuristic for the container loading problem**. Pesquisa Operacional, Rio de Janeiro, 27, n. 2, 2007. 311-331.
- BAZGAN, C., MONNOT, J., HASSIN, R., **Approximation algorithms for some vehicle routing problems**. Discrete Applied Mathematics 146, 2005. 27-42.
- BEASLEY, J.E., **An exact two-dimensional non-guillotine cutting tree search procedure**. Operations Research, 33 (1), 1985. 49-64.
- BELFIORI, P., **Sactter Search para problemas de roteirização de veículos com frota heterogênea, janelas de tempo e entregas fracionadas**. Tese (Doutorado em Engenharia de Produção), Universidade de São Paulo. São Paulo, p. 202. 2006.
- BISCHOFF, E. E., **Three-dimensional packing of items with limited loading bearing strength**. European Journal of Operational Research 168, 2006. 952-966
- BISCHOFF, E. E., RATCLIFF, M. S. W., **Issues in the development of approaches to container loading**. Omega 4, 1995. 377-390.
- BORTFELDT, A.; GEHRING, H., **A hybrid genetic algorithm for the container loading problem**. European Journal of Operational Research, 131, 2001. 143-161.
- BORTFELDT, A.; GEHRING, H.; MACK, D., **A parallel tabu search algorithm for solving the container loading problem**. Parallel Computing, 29 (5), 2003. 641-662.
- BRAYSY, O.; GENDRAU, M., **Vehicle Routing Problem with nTime Windows, Part I: Route Construction and Local Search Algorithms**. Transportation Science, 39, n. 1, 2005a, 104-118.
- BRAYSY, O.; GENDRAU, M., **Vehicle Routing Problem with nTime Windows, Part II: Metaheuristics**. Transportation Science, 39, n. 1, 2005b, 119-139.
- CAMPOS, D. S., **Integração de problemas de carregamento e roteamento de veículos com janela de tempo e frota heterogênea**. Tese (Doutorado em Engenharia de Produção) Universidade de São Paulo, São Paulo, p. 121. 2008.
- CHRISTENSEN, S.G.; ROUSØE, D.M., **Container loading with multi-drop constraints**. Technical Report, Technical University of Denmark, Kgs. Lyngby, p.18. 2007.

CECÍLIO, F. O., MORABITO, R., **Refinamentos na heurística de George e Robinson para o problema do carregamento de caixas dentro do contêineres.** Transportes 12, 2004. 32-45.

CIURLIZZA, A. G., CHÁVEZ, B. C., **Uso Del *Simulated Annealing* para minimizar El desperdicio de tela.** Congreso Latino-Ibero-Americano de Investigación Operativa 9, Buenos Aires, 1998. 1-10.

CORRÊA, E. S., **Algoritmos Genéticos e Busca Tabu aplicados ao Problema das P-medianas.** Dissertação (Mestrado em Métodos Numéricos em Engenharia) – Setor de Tecnologia, Universidade Federal do Paraná, Curitiba, 2000.

DAVIES, A. P.; BISCHOFF, E. E., **Weigh distribution considerations in container loading.** European Journal of Operational Research, 114, 1999. 509-527

ELEY, M., **Solving Container Loading Problems by block arrangement.** European Journal of Operational Research 141, 2002. 393-409.

ELEY, M., **A bottleneck assignment approach to the multiple container loading problem.** Operations Research Spectrum, 45, 2003. 45-60.

GEHRING, H., BORTFELDT, A. A., **A genetic algorithm for solving the container loading problem.** International Transactions In Operations Research 4, 1997. 401-418.

GEHRING, H., BORTFELDT, A. A., **A parallel genetic algorithm for solving the container loading problem.** International Transactions in Operational Research, 9 (4), 2002. 497-511.

GENDREAU, M.; IORI, M.; LAPORTE, G.; MARTELLO, S., **A Tabu Search Algorithm for a Routing and Container Loading Problem.** Transportation Science, 40, n. 3, Agosto 2006. 342–350.

GEORGE, J. A., ROBINSON, D. F., **A heuristic for packing boxes into a container.** Computers & Operations Research 7, 1980. 147-156.

GILMORE, P. C., GOMORY, R. E., **A linear programming approach to the cutting-stock problem.** Operations Research 9, 1961. 848-859.

GOLDBERG, D. E., **Genetic Algorithms in search, optimization and machine learning.** Menlo Park, Addison-Wesley, CA: 1986.

GUIMARÃES, T. A., **Uma nova abordagem heurística para a resolução do Problema do Roteamento de Veículos Capacitados com restrições tridimensionais de carregamento.** Dissertação (Mestrado em Métodos Numéricos em Engenharia) – Setor de Tecnologia, Universidade Federal do Paraná, Curitiba, p. 91. 2011.

HOLLAND, J. H., **Adaptation in natural and artificial systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.** Ann Arbor: The University of Michigan Press, 1975.

HOPPER, E., TURTON, B., **A Genetic Algorithm for a 2D industrial packing problem.** Computers and Industrial Engineering 37, 1999. 375-378.

IORI, M.; SALAZAR-GONZÁLES, J. J.; VIGO, D., **An exact approach for the vehicle routing problem with two dimensional loading constraints.** Transportation Science, 40, 2006. 342-350.

JIN, Z.; OHNO, K.; DU, J., **An efficient approach for the three-dimensional container packing problem with practical constraints.** Asia-Pacific Journal of Operational Research, 21 (3), 2004. 279-295.

JUNQUEIRA, L., MORABITO, R., YAMASHITA, D. S., **Modelos de otimização para problemas de carregamento de contêineres com considerações de estabilidade e de empilhamento.** Pesquisa Operacional 30, 2010. 73-98

LIANG, S. C., LEE, C. Y., HUANG, S. W., **A Hybrid Meta-heuristic for the Container Loading Problem.** Communications of IIMA 7, 2007. 73-84.

LIN, J.L.; CHANG, C.H.; YANG, J.Y., **A study of optimal system for multiple-constraint multiple-container packing problems.** In: Proceedings of the 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, 4031, 2006. 1200-1210, Annecy.

MACK, D.; BORTFELDT, A.; GEHRING, H., **A parallel hybrid local search algorithm for the container loading problem.** International Transactions in Operational Research, Vol. 11, 2004, pp. 511–534.

MORABITO, R., ARENALES, M., **An And/Or-graph approach to the container loading problem.** International Transactions in Operational Research 1, 1994. 59-73.

MORABITO, R., ARENALES, M., **Abordagens para o problema de carregamento de contêineres.** Pesquisa Operacional 17, 1997. 29-56.

MOURA, A.; OLIVEIRA, J. F., **A GRASP approach to the container loading problem.** IEEE Intelligent Systems, 20, n. 4, Julho 2005. 50-57.

OLIVEIRA, E. V., **Resolução do Problema de Carregamento de Container através de uma heurística.** Omnia Exatas 2, 2009. 18-26.

PISINGER, D., **Heuristics for the Container Loading Problem.** European Journal of Operational Research 141, 2002. 143-153.

RATICLIFF, M. S. W., BISCHOFF, E. E., **Allowing for weight considerations in container loading.** OR Spektrum 20, 1998. 65-71.

ROCHA, M. N., **Otimização de cortes de barras**. Dissertação (Mestrado em Ciência da Computação) – Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 1997.

SCHEITHAUER, G.; TERNO, J.; RIEHME, J.; SOMMERWEISS, U., **A new heuristic approach for solving the multi-pallet packing problem**. Technical Report, MATH-NM-03-1996, Technische Universität Dresden, Dresden. 1996.

SILVA, J. L. D. C.; SOMA, N. Y., **Um algoritmo polinomial para o problema do empacotamento de contêineres com estabilidade estática de carga**. Pesquisa Operacional, Rio de Janeiro, 23, n. 1, Janeiro 2003. 79-98.

SILVA, J.L.C.; SOMA, N.Y.; MACULAN, N., **A greedy search for the three-dimensional bin packing problem: the packing static stability case**. International Transactions in Operational Research, 10, 2003. 1-13.

TANOMARU, J., **Motivação, fundamentação e aplicações de Algoritmos Genéticos**. In: II Congresso Brasileiro de Redes Neurais, III Escola de Redes Neurais – Curitiba, 1995.

TERNO, J.; SCHEITHAUER, G.; SOMMERWEISS, U.; RIEHME, J., **An efficient approach for the multi-pallet loading problem**. European Journal of Operational Research, 123, 2000. 372-381.

TOTH, P.; VIGO, D., **The vehicle routing problem**. 1. ed. Filadélfia: SIAM - Monographs on Discrete Mathematics and Applications, 2002.

VENDRAMINI, E., **Otimização do Problema de Carregamento de Container utilizando uma metaheurística eficiente**. Dissertação (Mestrado em Engenharia Elétrica), Universidade Estadual Paulista, Faculdade de Engenharia de Ilha Solteira, Ilha Solteira, p. 99. 2007.

YEUNG, L.H.W.; TANG, W.K.S., **A hybrid genetic approach for container loading in logistics industry**. IEEE Transactions on Industrial Electronics, 52 (2), 2005. 617-627.