GABRIEL FERIANCIC

**MODELAGEM MATEMÁTICA DO
PROBLEMA DE PROGRAMAÇÃO DE
ENTREGAS DE DERIVADOS DE PETRÓLEO**

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do Título de
Mestre em Engenharia.

SÃO PAULO

2005

GABRIEL FERIANCIC

**MODELAGEM MATEMÁTICA DO
PROBLEMA DE PROGRAMAÇÃO DE
ENTREGAS DE DERIVADOS DE PETRÓLEO**

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do Título de
Mestre em Engenharia.

Área de Concentração:
Engenharia de Sistemas Logísticos

Orientador:
Prof. Dr. Cláudio Barbieri da Cunha

SÃO PAULO

2005

Dedico essa conquista
a Marisa e Stan,
a Vivi,
a Carol,
em memória, aos meus avós.

*“O valor das coisas não está no tempo que elas duram,
mas na intensidade com que acontecem.
Por isso, existem momentos inesquecíveis,
coisas inexplicáveis e pessoas incomparáveis”.*
(Fernando Pessoa)

AGRADECIMENTOS

Ao meu orientador Cláudio Barbieri da Cunha, por guiar e incentivar a todo o momento, pela confiança desde o início, e pelo grande exemplo de seriedade e simplicidade.

Aos outros professores das disciplinas de Engenharia de Sistemas Logísticos, pelos valiosos ensinamentos passados dentro e fora da sala de aula: Hugo Yoshizaki, Marco Mesquita, Nicolau Gualda, Marco Brinati e Rui Botter.

Ao amigo Frederico Mourad, pelos trabalhos desenvolvidos, pelas discussões e pela ajuda mútua nas aulas e no desenvolvimento desse trabalho.

Aos amigos Fernando Fleury, José Henrique Verroni e Rodrigo Verroni, que me apoiaram, permitindo dedicar esse tempo tão precioso ao meu desenvolvimento e aprendizado.

Ao meu pai, Stanislav Feriatic, que sempre me inspirou, incentivou e apoiou pessoal e profissionalmente.

RESUMO

Esta dissertação trata do problema da distribuição de combustíveis com caminhões-tanque para realizar a entrega de derivados de petróleo para diversos postos de abastecimento a partir de uma base de distribuição.

O problema consiste da determinação de rotas para veículos de uma frota heterogênea, visando minimizar o custo total de distribuição dos veículos envolvidos sujeitos a restrições de capacidade dos compartimentos de cada veículos. O objetivo é garantir que cada entrega seja alocada a exatamente um veículo e que todos os veículos sejam adequadamente seqüenciados. Deve-se notar que cada caminhão pode ter até seis compartimentos com diferentes capacidades. Além disso, são consideradas restrições que impedem que um veículo atenda determinado cliente.

As restrições relacionadas a essa alocação de pedidos aos compartimentos dos veículos fazem esse problema tornar-se muito diferente de outros problemas de roteirização de veículos. Para ilustrar isso, uma entrega de 5.000 litros para um cliente apenas pode ser alocada em um compartimento de exatamente 5.000 litros, mas não a um compartimento maior preenchido parcialmente. Adicionalmente, caminhões do mesmo tamanho e capacidade (e.g. 30.000 litros) podem possuir diferentes números de compartimentos, inclusive de diferentes tamanhos (e.g. um caminhão de 30.000 litros pode ter 6 compartimentos de 5.000 litros ou 2 compartimentos de 10.000 litros e 2 compartimentos de 5.000 litros), tornando o problema ainda mais complexo.

Propõe-se inicialmente uma modelagem matemática inédita para o problema. Dada a dificuldade de resolver instâncias de tamanhos reais utilizando ferramentas comerciais de otimização como o ILOG CPLEX 9.0, foi também proposto um algoritmo heurístico que pode alcançar boas soluções em tempos curtos de processamento. Este algoritmo é inspirado em algumas idéias do GRASP. Ele se baseia em um método heurístico rápido de construção, que é repetidamente aplicado, baseado em um algoritmo de controle que, repedita e aleatoriamente, remove alguns pedidos da solução corrente, e então reconstrói uma nova solução a partir dos pedidos não-alocados restantes.

Também são relatados resultados computacionais com diversos problemas de teste que foram gerados, considerando diferentes tamanhos de problema, bem como diferentes níveis de dificuldade de alocação de pedidos aos caminhões.

ABSTRACT

This Master's dissertation deals with the problem of distributing fuels by petroleum tank trucks in the context of the delivery of petroleum products to gas stations originating at a single distribution base.

The problem comprises determining the vehicle delivery routes for a heterogeneous fleet, aiming to minimize the total distribution and fixed costs of the vehicles involved subject to capacity constraints for the tank compartments of each vehicle. The objective is to ensure that each delivery is assigned to exactly one truck and all trucks are properly sequenced. It should be noticed that each truck may have one to six tank compartments with different capacities eventually. In addition, there may be restrictions on which vehicles can service each client.

The constraints related to the assignment of deliveries to truck compartments makes this problem much different from other vehicle routing problems, thus preventing the traditional routing approaches and formulations to be applied in this case. To illustrate this, a delivery of 5,000 liters to a single client can only be assigned to a compartment of exactly 5,000 liters, but not to a larger compartment which is not entirely filled up. In addition, trucks of the same size and capacity (e.g. 30,000 liters) may have different numbers of compartments and even different sizes (e.g. a 30,000 liters truck may have 6 compartments of 5,000 liters or 2 compartments of 10,000 liters and 2 compartments of 5,000 liters), making the problem even more complicated.

We initially propose a novel mathematical IP formulation for this problem. Given the difficulty to solve instances of the same size as found in practice using off-the-shelf cutting-edge optimization tools like ILOG CPLEX 9.0, we also propose a heuristic algorithm that can reach good solutions in very short CPU times. This algorithm is inspired on some ideas of GRASP. It relies on a fast constructive heuristic, which is repeatedly applied, based on a control algorithm that repeatedly and randomly remove some deliveries from the current solution, and then rebuilds a new solution from the remaining unassigned and unrouted deliveries.

We also report the computational results with several test problems that we have generated, considering different problem sizes, as well as different levels of difficulty related to assignment of orders to trucks.

SUMÁRIO

1. INTRODUÇÃO	1
1.1. Apresentação	1
1.2. Relevância do Problema	2
1.3. Organização	3
2. REVISÃO BIBLIOGRÁFICA.....	4
2.1. O Problema de Roteirização de Veículos	4
2.2. O Problema de Distribuição de Combustíveis na Literatura.....	8
2.3. Considerações Finais.....	14
3. CARACTERIZAÇÃO DO PROBLEMA	15
3.1. A Cadeia Logística de Distribuição de Derivados de Petróleo.....	15
3.1.1. <i>Das Refinarias para as Distribuidoras</i>	15
3.1.2. <i>Comercialização</i>	16
3.1.3. <i>A Frota</i>	17
3.1.4. <i>Alocação da Distribuição</i>	18
3.1.5. <i>Tipos de Pedido</i>	19
3.1.6. <i>Alocação Manual Empírica</i>	20
3.2. Modelagem Matemática do Problema.....	21
3.2.1. <i>Modelagem dos Caminhões-Tanque</i>	22
3.2.2. <i>Modelagem dos Clientes e da Base</i>	23
3.2.3. <i>Modelagem dos Pedidos</i>	25
3.2.4. <i>Variáveis de Decisão</i>	27
3.2.5. <i>Função objetivo</i>	27
3.2.6. <i>Restrições</i>	28
3.2.7. <i>Sumário da Modelagem Matemática</i>	30
3.3. Considerações Finais.....	31
4. DETERMINAÇÃO DA SOLUÇÃO EXATA	32
4.1. Solução Utilizando o <i>Solver</i> para Planilha Eletrônica	32
4.1.1. <i>Dados de Entrada</i>	33
4.1.2. <i>Variáveis de Decisão</i>	34
4.1.3. <i>Restrições</i>	37
4.1.4. <i>Resultados Obtidos Através do Solver</i>	40

4.2. Solução Utilizando <i>Concert Technology</i> do <i>ILOG CPLEX</i> em C++	42
4.2.1. Implementação.....	42
4.2.2. Exemplo de Solução Obtida pelo <i>ILOG CPLEX</i>	42
4.3. Outros problemas resolvidos através do <i>ILOG CPLEX</i>	44
4.4. Considerações Finais.....	46
5. DESENVOLVIMENTO DA ESTRATÉGIA DE SOLUÇÃO HEURÍSTICA.....	47
5.1. Ferramentas de Desenvolvimento.....	47
5.2. Aspectos Gerais do Método Proposto.....	48
5.2.1. <i>GRASP</i>	49
5.2.2. Comparação entre a Estratégia Adotada e o <i>GRASP</i>	51
5.3. Método Heurístico de Solução Desenvolvido	52
5.3.1. Heurística Rápida de Alocação	54
5.3.2. Verificação da função objetivo.....	62
5.3.3. Definições de novas condições iniciais para busca de nova solução	63
5.3.4. Verificação do critério de parada.....	66
5.4. Considerações Finais.....	66
6. EXPERIMENTOS COMPUTACIONAIS	68
6.1. Geração de Instâncias de Problemas para Teste.....	68
6.1.1. Critérios de geração de Instâncias	69
6.2. Variação do Nível de Restrição	72
6.3. Exemplo de Resultado.....	72
6.4. Variação dos Parâmetros	75
6.5. Resultados Obtidos.....	75
6.6. Análise Detalhada dos Resultados Obtidos.....	78
6.7. Considerações Finais.....	80
7. CONCLUSÕES E RECOMENDAÇÕES.....	82
REFERÊNCIAS BIBLIOGRÁFICAS.....	85
APÊNDICE A – MODELAGEM PARA EVITAR <i>SUBTOURS</i>	89
APÊNDICE B – IMPLEMENTAÇÃO EM <i>ILOG CPLEX</i> PARA C++	91
APÊNDICE C – PLANILHA DE AUXÍLIO NO DESENVOLVIMENTO	96
APÊNDICE D – RESULTADOS DOS PROCESSAMENTOS	101

LISTA DE FIGURAS

Figura 4.1 – Dados dos Veículos	33
Figura 4.2 – Dados de Distância	33
Figura 4.3 – Dados de Restrição entre Clientes e Veículos	34
Figura 4.4 – Dados dos Pedidos.....	34
Figura 4.5 – Variáveis de Decisão X	35
Figura 4.6 – Variáveis de Decisão Y	36
Figura 4.7 – Variáveis de Decisão Z.....	36
Figura 4.8 – Restrição de Alocação de Pedidos a Veículos	37
Figura 4.9 – Restrição de Compartimentos de 5000 litros.....	37
Figura 4.10 – Restrição de Volume Máximo	38
Figura 4.11 – Restrição de Alocação de Veículos a Cliente	38
Figura 4.12 – Restrição entre Veículos e Clientes	38
Figura 4.13 – Restrição de Continuidade das Rotas	39
Figura 4.14 – Restrição de Máximo de Rotas.....	39
Figura 4.15 – Primeira Restrição para Evitar Sub-Rotas	39
Figura 4.16 – Segunda Restrição para Evitar Sub-Rotas	40
Figura 4.17 – Exemplo de Saída do Aplicativo em ILOG CPLEX	43
Figura 5.1 – Fluxograma genérico de método GRASP.....	50
Figura 5.2 – Fluxograma geral do algoritmo proposto	53
Figura 5.3 –Visão Geral da Heurística Rápida de Alocação.....	55
Figura 5.4 – Função de Verificação das Restrições Estáticas.....	57
Figura 5.5 – Parâmetro de Pedidos para um Mesmo Cliente.....	58
Figura 5.6 – Parâmetros de Serventia do Veículo e Facilidade de Alocação do Pedido.....	60
Figura 5.7 – Função de Busca para Determinação de Pedido a Alocar	61
Figura 5.8 – Visão Geral da Geração de Novas Condições Iniciais	64

Figura 5.9 – Sorteio e Desalocação de Pedidos	65
Figura 6.1 – Exemplo de Relatório de Comportamento do Algoritmo.....	73
Figura 6.2 – Exemplo de Rotas Geradas.....	74
Figura A.1 – Rede Ilustrativa sem <i>Subtours</i>	89
Figura A.2 – Rede Ilustrativa com Dois <i>Subtours</i>	90
Figura B.1 – Exemplo de Implementação em ILOG CPLEX para C++.....	95
Figura C.1 – Dados de clientes	96
Figura C.2 – Dados de pedidos	97
Figura C.3 – Dados de frota	98
Figura C.4 – Mapeamento dos clientes	98
Figura C.5 – Matriz de restrições.....	99
Figura C.6 – Matriz de alocação	100

LISTA DE QUADROS E TABELAS

Quadro 3.1 – Codificação dos Tipos de Caminhões	18
Quadro 3.2 – Exemplo de Tabelas para Alocação de Pedidos (Volumes em 1000 l).....	21
Tabela 2.1 – Exemplo de Matriz de Compartimentos Disponíveis.....	9
Tabela 4.1 – Resultado da Implementação no <i>Solver</i>	41
Tabela 4.2 – Instâncias e Resultados do Processamento no ILOG CPLEX	45
Tabela 5.1 – Exemplo de Cálculo de Serventia e Facilidade.....	59
Tabela 6.1 – Distribuição Arbitrada para Volume dos Pedidos.....	70
Tabela 6.2 – Distribuição Arbitrada para Tipo de Veículo.....	71
Tabela 6.3 – Resumo dos Resultados dos Testes Realizados	77
Tabela C.1 – Codificação dos Tipos de Caminhões	98
Tabela D.1 – Resultados para Classe A – Restrição 20%.....	102
Tabela D.2 – Resultados para Classe A – Restrição 40%.....	102
Tabela D.3 – Resultados para Classe A – Restrição 60%.....	103
Tabela D.4 – Resultados para Classe B – Restrição 20%	103
Tabela D.5 – Resultados para Classe B – Restrição 40%	104
Tabela D.6 – Resultados para Classe B – Restrição 60%	104
Tabela D.7 – Resultados para Classe C – Restrição 20%	105
Tabela D.8 – Resultados para Classe C – Restrição 40%	105
Tabela D.9 – Resultados para Classe C – Restrição 60%	106

LISTA DE ABREVIATURAS E SIGLAS

ANP	Agência Nacional de Petróleo
EPUSP	Escola Politécnica da Universidade de São Paulo
USP	Universidade de São Paulo
SA	Simulated Annealing
DA	Deterministic Annealing
TS	Tabu Search / Busca Tabu
GA	Genetic Algorithm / Algoritmo Genético
AS	Ant System
NN	Neural Network / Rede Neural
GRASP	Greedy Randomized Adaptative Search Procedure
INMETRO	Instituto Nacional de Metrologia, Normalização e Qualidade Industrial

1. INTRODUÇÃO

1.1. Apresentação

O presente trabalho trata da modelagem matemática e da elaboração de uma proposta de solução para o problema de programação de entregas de derivados de petróleo em postos de abastecimento.

As redes distribuidoras desses produtos concentram em pontos estratégicos as bases operacionais onde os combustíveis comprados das refinarias são armazenados. A partir desses locais, os diferentes produtos (gasolina, álcool e óleo diesel) são distribuídos aos postos de abastecimento utilizando veículos específicos. Tais veículos, chamados de caminhões-tanque, são apropriados para o transporte de líquidos inflamáveis e possuem compartimentação interna que permite o transporte simultâneo de um ou mais produtos para diferentes clientes.

Essa compartimentação constitui como um dos principais elementos complicadores do problema aqui proposto, pois os veículos que compõem a frota não são idênticos. Em outras palavras, os veículos-tanque apresentam diferentes tamanhos e também compartimentações diferentes em termos do número de tanques e das capacidades dos tanques, levando o problema a uma formulação de distribuição com frota heterogênea. Há ainda outras particularizações que podem gerar mais restrições e que serão apresentadas e discutidas.

A decisão de alocação da frota para realizar a distribuição passa pela análise de um número muito grande de possíveis combinações de pedidos e veículos. Quanto maior o problema a ser solucionado, maior é o número de possíveis combinações, que pode crescer exponencialmente, tornando mais difícil encontrar a solução ótima.

Assim, este trabalho tem como objetivos principais propor um modelo matemático que represente adequadamente o problema e conceber, implementar e testar uma solução heurística para este problema.

1.2. Relevância do Problema

A cadeia logística dos combustíveis automotores é parte da cadeia petroquímica. Os processos envolvidos nessa cadeia começam na extração do produto natural bruto, seguindo-se o transporte às refinarias, os processos industriais de refino e beneficiamento e de distribuição, até o consumo final na forma dos mais variados produtos como combustíveis, lubrificantes, parafinas e asfalto, entre outros.

O papel dos óleos combustíveis é fundamental no desenvolvimento econômico, e os preços finais desses produtos causam impacto em todas as cadeias produtivas e na própria situação econômica do país. Isso é ainda mais verdadeiro sob a ótica de um país como o Brasil que, segundo o GEIPOT (2001), tem em seu sistema rodoviário o principal meio de escoamento da produção nacional, o qual correspondeu a 60,5% da produção de transporte de carga em 2000 (em toneladas-quilômetro).

Conforme indicado pela ANP - Agência Nacional de Petróleo (2001), em 2000 os postos revendedores de combustíveis automotivos (postos de combustível) totalizavam 29.111 estabelecimentos em todo o território nacional, sendo 7.878 localizados apenas no Estado de São Paulo. A maior distribuidora do Brasil possuía, sozinha, 1.546 postos neste estado.

O desafio de distribuição de uma rede deste porte passa por questões comuns a qualquer problema dessa natureza, mas também por detalhes operacionais que dificultam sua otimização. O transporte da distribuição dos combustíveis para o varejo é apenas uma pequena etapa da cadeia produtiva, mas que gera uma parcela significativa no custo final dos produtos.

Assim, o presente projeto de pesquisa tem por objetivo contribuir para a racionalização da atividade de distribuição de derivados de petróleo. A redução de custos em combustíveis automotivos, em especial, impacta de forma benéfica todos os setores da economia.

Além da economia dos custos sociais em todos setores produtivos possibilitada pela otimização na distribuição de combustíveis automotivos, a importância da pesquisa proposta pode ser vista pela sua complexidade do ponto de vista de modelagem matemática e solução, pois o problema tem relevantes características combinatórias. Busca-se contribuir, então, através da elaboração de uma modelagem matemática

inédita e da experimentação no uso de uma heurística de solução. Adicionalmente, trata-se de um problema muito pouco estudado na literatura.

No caso particular do problema de roteirização e programação de entrega de combustíveis, além dos aspectos usuais de um problema de distribuição, as características operacionais, em especial a necessidade de compatibilização dos volumes pedidos com as capacidades dos compartimentos dos veículos, impede a utilização dos métodos de solução propostos na literatura, requerendo o desenvolvimento de uma estratégia de solução específica.

1.3. Organização

O presente texto está organizado da seguinte forma: o Capítulo 2 apresenta a revisão bibliográfica abordando duas vertentes: a questão de roteirização de veículos, incluindo trabalhos desenvolvidos na EPUSP, e estudos sobre o problema específico tema deste trabalho.

O problema proposto é caracterizado no Capítulo 3 e permite compreender os aspectos práticos da questão. Este capítulo compreende ainda a formulação da modelagem matemática do problema de otimização da distribuição de combustíveis.

O Capítulo 4 trata das tentativas de obter a solução exata para o problema. São apresentadas duas formulações: em planilha eletrônica *Excel* e no aplicativo *ILOG CPLEX*. O desempenho e os resultados dessas formulações também são apresentados no capítulo.

A estratégia de solução heurística para otimização do problema em estudo é apresentada no Capítulo 5.

No Capítulo 6 são apresentados os resultados alcançados a partir da resolução de várias instâncias de problema através do método proposto. No mesmo capítulo são discutidos os resultados alcançados, avaliado o desempenho do método e apresentadas análises consideradas relevantes.

O Capítulo 7 apresenta as conclusões e recomendações do presente trabalho de pesquisa.

2. REVISÃO BIBLIOGRÁFICA

2.1. O Problema de Roteirização de Veículos

A formulação original do problema de roteirização de veículos foi proposta por Dantzig e Ramser (1959) e, desde então, este problema vem sendo intensivamente estudado, abordando diferentes tipos de problemas e suas formulações, bem como diferentes estratégias de solução. Essa classe de problemas tem como formulação geral a determinação de rotas para os veículos de uma determinada frota, com o objetivo de minimizar os custos de transporte. Bodin *et al.* (1983) apresentam discussões e formulações para diversas categorias do problema de roteirização.

Problemas de que envolvem roteirização e programação de veículos em geral são problemas combinatórios, do tipo NP-difícil, o que torna impossível a obtenção de soluções ótimas para instâncias encontradas no mundo real, trazendo o desafio da busca de novas heurísticas mais eficientes que possibilitem a sua aplicação a problemas práticos do dia-a-dia das empresas (Cunha, 2000).

Ainda segundo Cunha (2000), o constante interesse em busca de novas estratégias de solução se deve ao fato de que, devido à complexidade computacional, as estratégias de solução que permitem resolver problemas reais são heurísticas, apóiam-se em abordagens intuitivas, na qual a estrutura particular do problema possa ser considerada e explorada de forma inteligente, para a obtenção de uma solução adequada (Cunha, 1997).

Estratégias de solução heurísticas são, por natureza, bastante específicas e, em geral, carecem de robustez, isto é, não conseguem proporcionar boas soluções para problemas com características, condicionantes ou restrições às vezes um pouco diferentes daquelas para as quais foram desenvolvidas (Teixeira e Cunha, 2002). Em outras palavras, roteirização de veículos é uma área onde uma solução para um determinado tipo de problema e dados e condicionantes pode não ser adequada para outro problema similar, conforme apontado por Hall e Partyka (1997).

Laporte *et al.* (2000) apresentam uma revisão de métodos heurísticos para solução de problemas de roteirização de veículos. Os autores descrevem inicialmente a modelagem da versão mais estudada do problema de roteirização de veículos:

Seja $G = (V, E)$ um grafo, onde:

- $V = \{v_0, v_1, \dots, v_n\}$ é um conjunto de nós; e
- $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ é um conjunto de arcos.

O vértice v_0 representa um depósito onde estão sediados m veículos idênticos de capacidade Q . A cada vértice v_i é associada uma demanda não-negativa q_i . Também é associado um custo não-negativo c_{ij} a cada arco (v_i, v_j) .

O problema de roteirização de veículos consiste de determinar um conjunto de m rotas, que busque um custo total mínimo, mas que sejam restritas a:

- Cada uma delas iniciar e terminar no depósito (v_0);
- Cada cliente (v_i , com $i > 0$) visitado uma única vez e por um único veículo;
- A demanda total da rota de cada veículo não ultrapasse sua capacidade Q .

Deve-se notar que essa formulação não considera frota heterogênea e os eventuais “trade-off” que podem ocorrer. Qual a alternativa de menor custo: um número menor de caminhões maiores percorrendo maior distância total ou uma frota maior de veículos menores percorrendo menor distância total?

As heurísticas de solução são classificadas por Laporte *et al.* em *heurísticas clássicas* e *heurísticas modernas*.

As *heurísticas clássicas* são, por sua vez, classificadas em três categorias. *Heurísticas de construção* geram ou constroem soluções viáveis buscando otimizar o custo da solução, porém sem uma fase de melhoria destacada. *Heurísticas de melhoria* buscam a melhoria de qualquer solução possível através de trocas de segmentos de solução de cada rota dentro da própria rota ou entre rotas diferentes. *Heurísticas de duas fases*, por fim, decompõem o problema em dois componentes: agrupamento de vértices em rotas possíveis e construção de rotas, com possível troca de informações entre essas fases.

Laporte *et al.* listam algumas heurísticas modernas, também conhecidas como *meta-heurísticas*, que já foram aplicadas a problemas dessa natureza: *Simulated Annealing* (SA), *Deterministic Annealing* (DA), *Busca Tabu* (TS), *Algoritmos Genéticos* (GA), *Ant Systems* (AS) e *Redes Neurais* (NN).

Os algoritmos *Simulated Annealing*, *Deterministic Annealing* e Busca Tabu iniciam por uma solução inicial e a cada iteração movem para uma solução vizinha até atingir um critério de parada.

Algoritmos da classe *Algoritmos Genéticos* examinam uma população de soluções, na qual cada geração é derivada de geração uma precedente através da combinação de seus melhores elementos e descartando os piores.

A estratégia chamada de *Ant Systems* baseia-se na construção de diversas soluções em cada iteração utilizando informações colhidas nas iterações anteriores. Esse método é inspirado no comportamento operativo das formigas para encontrar o melhor caminho, onde cada indivíduo compartilha a informação com os outros indivíduos acerca de seu caminho aleatório experimentado.

Algoritmo do tipo *Rede Neural* é um mecanismo que gradualmente ajusta parâmetros até alcançar a solução.

Segundo Pirlot (1996), esses métodos são inspirados em processos que não estão relacionados à otimização. Seus sucessos, no entanto, estão justamente relacionados à referência de processos encontrados na Natureza, aplicabilidade genérica, flexibilidade para aceitar restrições específicas de casos reais e excelente relação (*trade-off*) entre a qualidade da solução e a facilidade de implementação ou tempo computacional.

Diversos autores demonstraram aplicações desses métodos em problemas clássicos de roteirização de veículos, como a aplicação de uma estratégia de busca tabu para um problema com frota heterogênea descrito por Gendreau *et al.* (1999).

Segundo Cunha (2000), mais recentemente, em particular na última década, o esforço de pesquisa vem sendo direcionado à investigação das chamadas meta-heurísticas, como as propostas no presente trabalho.

Este trabalho integra um conjunto de pesquisas que vêm sendo realizadas nas áreas de Logística e de Engenharia de Transportes na Escola Politécnica da USP e que envolvem, genericamente, o desenvolvimento e a implementação computacional de novos algoritmos e heurísticas de solução para problemas de roteirização e programação de veículos, cujas principais contribuições mais recentes estão relacionadas a seguir:

- Cunha (1997) tratou do problema de roteirização de veículos com restrições operacionais de janelas de tempo e duração máxima da jornada. A estratégia de

solução proposta baseou-se na relaxação lagrangeana das restrições do modelo, a partir da qual foram desenvolvidas duas heurísticas para o problema com frota homogênea e uma terceira para frota heterogênea;

- Souza (1999) propôs uma estratégia de solução para o problema de transporte do tipo “Carga Única Coleta e Entrega” (*Full Truck Load Pickup and Delivery*) com janelas de tempo em duas etapas: na primeira são gerados todos os roteiros tecnicamente viáveis para atender as requisições de transporte dentro do período de programação, e na segunda é resolvido um problema de programação linear, do tipo partição de conjunto (*set partitioning*) para a seleção dos melhores roteiros que minimizem o custo total do transporte;
- Znamensky (2000) tratou do problema de roteirização e programação de veículos para o transporte porta-a-porta de idosos e portadores de necessidades especiais, através de veículos de pequena capacidade, tais como peruas ou vans. Nesse trabalho, buscou-se determinar roteiros econômicos que atendam a um conjunto de pessoas e, ainda, que respeitem as restrições operacionais e temporais. Destaca-se nesse problema sua formulação de múltiplas bases. A estratégia de solução para o problema de porte real foi baseada na heurística de inserção paralela que produziu soluções de boa qualidade (redução de custos operacionais e de frota) com agilidade computacional;
- Teixeira (2001) propôs heurísticas para tratar o problema de dimensionamento e roteirização de uma frota heterogênea baseadas na utilização do algoritmo Out-of-Kilter. Essas heurísticas baseiam-se nas combinações de rotas a partir da solução dos problemas de designação, pois o problema em questão consiste, resumidamente, em compor e roteirizar uma frota de veículos, considerando os seus custos fixos e variáveis e as restrições de capacidade, de tal forma que os custos de distribuição sejam minimizados. Assim, o problema de designação foi entendido como um problema de circulação com custo mínimo;
- Fonseca (2002) tratou da alocação de atendimentos (coleta e entrega de mercadorias) às rotas dos veículos das empresas de remessa expressa em grandes centros urbanos, e propôs uma estratégia de solução para minimizar os tempos e custos de coleta e entrega de mercadorias (“porta-a-porta”), com base nos conceitos da heurística de inserção concebida por Solomon (1987), porém

com alocação sequencial dos atendimentos e inserção paralela em cada uma das rotas do conjunto.

- Galvão (2004) considerou o problema de programação de veículos de coleta de resíduos de biomassa de madeira, visando à sua utilização como combustível para abastecimento de uma central produtora de energia e/ou de vapor. O problema consiste na escolha de conjunto de fornecedores, para minimizar o custo de abastecimento, e na determinação da frota mínima necessária para realizar a coleta durante um horizonte de programação correspondente a vários dias, considerando que cada cliente tem uma periodicidade de coleta diferente. Os veículos realizam viagens com carga completa entre a central e cada uma das geradoras.

2.2. O Problema de Distribuição de Combustíveis na Literatura

A distribuição de derivados de petróleo precisa considerar essa compartimentação dos veículos da frota. O caminhão-tanque é normalmente dividido em vários compartimentos e, para melhor aproveitamento de cada veículo, cada cliente deve apenas receber um volume do produto correspondente a um ou mais compartimentos preenchidos totalmente. A prática do setor de transporte de combustíveis impede que em um mesmo compartimento seja carregado combustível para mais de um cliente. Essa questão será discutida em maior profundidade no próximo capítulo.

Bechara e Galvão (1984) trataram do problema de roteirização de veículos para entrega de combustíveis ressaltando que um dos fatores complicadores ao problema é a compartimentação de veículos.

Bechara e Galvão (1984) sugerem, como alternativa para modelar essa restrição, a proibição de combinações de pedidos em veículos que sejam impossíveis de ocorrerem pelas configurações de compartimentos existentes em um mesmo veículo. Ou seja, toda ligação entre quaisquer dois clientes cujas demandas não correspondam a dois compartimentos existentes no mesmo veículo deve ser proibida.

Uma alternativa proposta pelos autores é a substituição da capacidade dos veículos por restrições associadas a uma matriz que indique os compartimentos disponíveis em cada

veículo. Para cada caminhão, então, considera-se capacidades discretas (ao invés de contínuas), conforme exemplo dado pelos autores, presente na Tabela 2.1 a seguir. Por exemplo, o *Veículo 2* possui 2 compartimentos de 8000 l, 1 de 5000 l e 1 de 3000 l.

Tabela 2.1 – Exemplo de Matriz de Compartimentos Disponíveis

Compartimento Veículo	11.000 litros	8.000 litros	5.000 litros	3.000 litros
1	0	2	1	1
2	0	2	1	1
3	1	2	1	0

Para cada ligação adicionada a um elemento da frota, a matriz de compartimentos disponíveis deve ser atualizada.

Bechara e Galvão não apresentam método eficiente de solução para esse caso específico, mas ressaltam claramente que essa restrição comprometia em muito a qualidade das soluções para o algoritmo heurístico de Carke & Wright, utilizado por eles par o desenvolvimento de solução para o problema de roteamento de veículos.

Brown e Graves (1981) descrevem a elaboração de um sistema de distribuição automatizado em tempo-real, com sub-rotinas de otimização para substituir o trabalho manual e otimizar os custos de distribuição.

O problema consiste na distribuição de até 16 diferentes produtos derivados de petróleo a partir de centros de distribuição através dos quais os caminhões-tanque são abastecidos. Os veículos variam muito e, em cada região, possuem configurações resultantes de legislações locais, padrões geográficos e temporais de demanda e práticas históricas de gestão. São também diferenciados caminhões próprios de terceirizados, indicando que podem ter composições diferentes de valor de frete.

Brown e Graves analisaram o processo na seguinte sequência de eventos:

- 1) Pré-análise do despacho: Verificação das informações dos pedidos e dos veículos, revisão de casos especiais e complementação de informações faltando.
- 2) Edição de Veículo Compatível: Determinação de quais veículos podem ser usados para atender cada pedido, considerando as restrições de equipamento e compartimentação.

- 3) Alocar Pedidos a Veículos: Alocação com objetivo de minimizar custos respeitando restrições do veículo e da operação.
- 4) Ajuste das Quantidades dos Pedidos: Ajuste nas quantidades dos pedidos, caso necessário, para adequá-los a compartimentação disponível e à sequência de carregamento possível.
- 5) Revisão Final: Identificação de condições excepcionais restantes e realização de pequenos ajustes.
- 6) Despacho: Impressão dos documentos de carregamento na base de distribuição.

Os autores descreveram que os passos 1 e 5 são de julgamento humano e os passos 2 e 4 pareciam ser feitos de forma simples com pequenas regras básicas. O foco do trabalho foi então a modelagem da automação do passo 3, correspondente à alocação dos pedidos.

Para desenvolvimento dessa alocação, Brown e Graves propuseram um modelo de programação inteira. A formulação considerava uma função objetivo em que se busca minimizar os custos e as penalidades por desrespeitar restrições de duração da rota. Ou seja, a formulação permite violação de restrições aplicando uma penalidade de custo. O modelo limitou-se a entregas de veículo completo (*full truck load*), pois os consumidores eram instruídos a pedir apenas dessa forma. As entregas de carga parcelada eram exceções e os autores informaram que a alocação para essas viagens de múltiplos pontos de atendimento não era facilmente automatizada, além de não haver informações disponíveis de tempo e distância de viagens entre clientes distintos.

A otimização por programação inteira foi implementada e testada. Comparado com o método manual, os resultados geraram uma melhor distribuição de carga de trabalho entre os veículos com significativas reduções de custo. Os resultados também revelaram que diversos critérios admitidos anteriormente pela alocação manual eram muito antieconômicos.

O sistema de alocação, porém, foi considerado não-ideal para funcionar em tempo real em conjunto com a operação. A grande quantidade de dados e o volume de transações do banco de dados já consumiam a maior parte da capacidade de processamento do sistema, restando pouca capacidade para a execução da otimização por esse método.

Brown e Graves, então, propuseram a adoção de uma heurística eficiente com otimização inclusa (*efficient heuristic with embedded optimization*). A heurística foi desenvolvida com um seqüenciamento de geração e resolução de problemas de rede. A cada geração, alguns pedidos são alocados por critério do custo projetado e penalizado, respeitando restrições dos veículos.

Segundo os autores, após diversos ajustes e testes a heurística permitiu o módulo de automação da distribuição produzir excelentes resultados com boa performance e eficiência.

A partir do sistema de programação de viagens desenvolvido por Brown e Graves (1981) para uma única base de distribuição, foi desenvolvida uma extensão por Brown *et al.* (1987). Neste trabalho seguinte, a heurística de alocação foi utilizada com uma parte de um mecanismo de processamento maior de forma a permitir a alocação da distribuição a partir de múltiplas bases e com roteirização de atendimentos a diversos clientes por cada veículo.

Brown *et al.* (1987) relataram a implantação de um sistema de auxílio computacional (chamado de *computer assisted dispatch*) no controle centralizado do comércio e da distribuição de derivados de petróleo em todo o território continental dos Estados Unidos da América. Segundo os autores, esse sistema representou importantes reduções de custos e equipes e melhorias do serviço aos clientes. Eles ressaltam que os custos operacionais são muito sensíveis à decisão da alocação que a decisão no momento do despacho segue regras complexas que envolvem segurança e eficiência da operação. Assim, programadores de despacho “humanos” buscam apenas soluções possíveis aceitáveis e a otimização do problema é muito complexa para ser realizada dessa forma.

O sistema considerado por Brown *et al.* (1987) é composto por muitos terminais e cada terminal podendo conter diferentes produtos disponíveis entre 20 diferentes tipos. A frota considerada era mista (veículos próprios e terceirizados), alocada em alguns terminais. Outros terminais não dispõem de frota no local. Os caminhões possuíam diferentes capacidades e formatos de compartimentação. Os objetivos do processo de despacho são minimizar os custos, balancear o trabalho entre as companhias de caminhões e buscar o maior aproveitamento possível da frota.

As decisões tomadas para atingir esses objetivos são: escolher o terminal que atende cada pedido, alocar os pedidos a cada veículo, verificar a adequação dos volumes aos compartimentos do veículo e roteirizar os caminhões e sequenciar as entregas.

Uma das partes de todo o sistema desenvolvido por eles é um módulo de processamento automatizado que busca a otimização dessa alocação, pela aplicação de, segundo os autores, “um conjunto de métodos de otimização altamente especializados”. Esses métodos são divididos em seis etapas:

- 1) Edição e Análise: Construção de tabelas de custos, restrições e outras informações com relação à ligação das bases com os clientes, entre os clientes e dos caminhões com os clientes.
- 2) Alocação Inicial: Alocar pedidos como carregamentos, sem verificar ainda a sequência de entrega, sendo apenas o ponto de início da otimização.
- 3) Alocação Linear: A alocação de pedidos aos caminhões é realizada objetivando a redução de custos esperados. Baseia-se na troca entre pedidos simples, duplas de pedidos e assim por diante, em geral permitindo até a segunda troca por veículo (também chamado de melhorias k-opt).
- 4) Alocação Quadrática: A alocação determina, então, qual a melhor rota, ou sequência de carregamentos e entregas para os veículos, através de uma formulação do Problema do Caixeiro Viajante.
- 5) Carregamento Final: Após a alocação e sequenciamento de todos os pedidos, uma rotina específica e verifica quanto de cada produto será carregado em cada compartimento de cada veículo.
- 6) Trocas: A solução final é testada, então, para possíveis melhorias a partir dos custos conhecidos (e não estimados) através da tentativa de trocar um pedido de veículo ou trocar dois pedidos entre veículos.

Avella, Boccia e Sforza (2004) abordaram um problema de distribuição de combustível com uma formulação próxima à desenvolvida neste trabalho. Eles estudaram a distribuição para postos, a partir de um único local, com uso de frota heterogênea de caminhões-tanque, isto é, diferentes capacidades e perfil de compartimentos.

O objetivo do problema descrito pelos autores é otimizar os custos de transporte com os recursos já disponíveis. A solução do problema é um conjunto de rotas de cada veículo que devem iniciar e terminar em um depósito (base).

As restrições que restringem a distribuição são: cada cliente deve ser visitado uma única vez por dia; o custo total de viagem diária precisa ocorrer dentro do tempo de um dia de trabalho de cada veículo; e cada veículo não pode ser carregado além de sua capacidade para cada rota simples.

O enfoque dado pelos autores buscou resolver o problema por um método exato a partir de uma modelagem baseada em *Set Partitioning*, sendo utilizado um algoritmo *Branch-and-Price* para a solução do problema. A estratégia de solução proposta pelos autores compreendeu ainda uma heurística rápida para a busca de uma boa solução viável e para realizar o pré-processamento do algoritmo exato a fim de limitar o número de combinações avaliadas.

Para essa solução, os autores elaboraram uma matriz na qual cada linha representa um cliente e cada coluna uma rota de um caminhão. Um custo é calculado através dessas rotas e associado a cada coluna. Para reduzir o número de colunas e identificar eliminar algumas soluções inviáveis, realiza-se um processamento que elimina colunas quando elas não satisfazem as restrições do problema (como capacidade do caminhão e jornada máxima) ou quando não satisfazem a economia mínima dada pela heurística.

Em seguida, a otimização é dada pela minimização do custo total sujeito às restrições de cada cliente ser visitado apenas uma vez por dia e o custo total por caminhão não exceder a jornada de trabalho do dia.

As soluções encontradas foram comparadas quanto ao tempo computacional necessário, custo total e número de caminhões utilizados.

A primeira diferença entre o problema estudado por Avella, Boccia e Sforza e o problema apresentado neste trabalho é na formulação matemática, voltada para o método de solução de *Set Partitioning*. As restrições principais estão consideradas nas colunas geradas e eliminadas, sobrando um problema de otimização com formulação mais simples.

A segunda diferença é que os caminhões podem fazer mais de uma viagem e o horizonte de análise é uma jornada de trabalho diária. A formulação é focada para

diversos clientes a serem atendidos diariamente e, por consequência, na decisão da frota a ser adotada. No problema que será aqui apresentado, no entanto, a decisão é na forma de atender um grupo de pedidos a serem entregues com cada caminhão realizando apenas uma viagem.

Além disso, outros detalhes também diferem como tamanhos e compartimentação dos veículos e o número de clientes que podem ser atendidos em cada viagem: quatro, apesar dos veículos possuírem de sete a nove compartimentos.

2.3. Considerações Finais

O problema proposto nesse trabalho é o de distribuição com frota heterogênea com compartimentação com veículos atendendo múltiplos clientes a partir de uma única base. Outras especificidades são consideradas e descritas no próximo capítulo. Propõe-se também a busca por um método heurístico de solução.

Assim, o trabalho aqui proposto se difere dos demais, seja pela formulação do problema (distribuição atendendo múltiplos clientes a partir de uma única base), seja pela estratégia de solução adotada (solução através de um método heurístico).

Por fim, é possível sintetizar os principais aspectos da literatura existente para distribuição de combustíveis para postos de abastecimento:

- Bechara e Galvão (1984) apenas apontam a existência da questão da compartimentação dos veículos como fator complicador do problema de distribuição.
- Brown e Graves (1981) resolvem o problema para atendimentos de um único cliente por veículo a partir de uma única base através de método exato e heurístico.
- Brown *et al.* (1987) estendem a solução heurística anterior para múltiplos clientes para veículos partindo de múltiplas bases.
- Avella, Boccia e Sforza (2004) resolvem um problema semelhante ao aqui proposto, a partir de uma base, com veículos atendendo a múltiplos clientes, através de um método exato, utilizando uma heurística para realizar o pré-processamento do mesmo. Esta formulação baseia-se, no entanto, numa modelagem *Set Partitioning*.

3. CARACTERIZAÇÃO DO PROBLEMA

Este capítulo tem por objetivo apresentar a caracterização completa do problema proposto da programação de entregas de derivados de petróleo em postos de abastecimentos a partir de uma base de distribuição. Inicialmente, é apresentado na seção 3.1 a caracterização geral, sua inserção na cadeia logística petroquímica e as principais características e condicionantes do problema. Na seção 3.2, em seguida, é apresentado o detalhamento do modelo matemático proposto.

3.1. A Cadeia Logística de Distribuição de Derivados de Petróleo

A cadeia petroquímica é composta por diversas etapas, que vão desde a extração do produto natural bruto até o consumo final na forma de derivados. Segundo Silveira (2002): “Na cadeia petrolífera, compreendem o *upstream* as atividades de exploração e produção de petróleo e gás natural. O segmento *midstream* inclui as atividades de refino, transporte, e a importação e exportação de petróleo, gás natural e derivados. Define-se como *downstream* o segmento da cadeia petrolífera que inclui a distribuição e a revenda de derivados a partir das refinarias de petróleo”.

O ciclo *downstream* é caracterizado pelas etapas de transporte dos produtos da refinaria às bases das distribuidoras, armazenagem e distribuição o ponto de atendimento ao consumidor final: os postos de abastecimento, também chamados de “postos de combustível”.

A distribuição de derivados consiste, então, no transporte do produto final no ciclo *downstream*, a partir dos insumos recebidos por refinarias de petroquímicas e refinarias de álcool.

3.1.1. Das Refinarias para as Distribuidoras

Os reservatórios das distribuidoras são os pontos onde são armazenados os combustíveis visando a sua distribuição para os postos de combustível.

Em geral, gasolina e óleo diesel são originários de uma refinaria petroquímica onde são produzidos a partir do refino de petróleo bruto, através do transporte usualmente feito

por dutos. Por isso, ao redor das refinarias é comum o estabelecimento de diversas bases de redes de companhias distribuidoras (por exemplo: Shell, Ipiranga, BR etc.).

O Álcool é recebido, geralmente, por meio rodoviário, através de caminhões-tanque que trazem o produto originário de regiões de plantio e beneficiamento de cana-de-açúcar.

Os produtos são comprados por volume e existem quatro tipos de produtos recebidos: Óleo Diesel, Gasolina Tipo A, Álcool Anidro e Álcool Hidratado.

O Óleo Diesel e o Álcool Hidratado são os mesmos produtos vendidos posteriormente nos postos. A Gasolina Tipo A e o Álcool Anidro são misturados pelas distribuidoras na proporção 80% e 20%, respectivamente, para formar a Gasolina Tipo C, que é a vendida nos postos de combustível.

3.1.2. Comercialização

A comercialização dos produtos aos postos de combustível e demais clientes (empresas de transporte, indústrias etc.) pelas distribuidoras pode ser realizada pelo escritório localizado na base de distribuição ou em escritórios comerciais avançados. Caso a distribuidora possua mais de uma base, a eficiência logística pode estar atrelada à decisão de que base deve atender cada pedido, antes de proceder a programação de distribuição e a roteirização dos pedidos.

Algumas distribuidoras, no entanto, visualizam cada base de distribuição como uma unidade de comercialização e, assim, pedidos vendidos por cada base são atendidos por ela, mesmo que a eficiência do resultado global possa ser comprometida.

A distribuidora, como o próprio nome define, tem como produto principal o transporte dos derivados para os pontos de venda e sua eficiência logística atinge diretamente o preço (valor do frete) e a qualidade (tempo e confiabilidade de entrega).

Entre práticas percebidas durante a venda que visam economia no transporte, percebeu-se que os vendedores são instruídos a tentar realizar *vendas casadas*, isto é, tentar completar volumes de 15.000 l ou 30.000 l para uma mesma região. Esses volumes estão relacionados à capacidade máxima de um caminhão típico e permite uma viagem com carga completa. Porém, isso acaba sendo apenas intuitivo e extremamente dependente de condições de comercialização e da demanda dos clientes.

3.1.3. A Frota

A compartimentação dos veículos é uma importante restrição do problema. Ela é padronizada e os pedidos apenas podem ser alocados em compartimentos completos, isto é, um veículo apenas utiliza um compartimento totalmente vazio ou totalmente cheio.

A verificação da quantidade de combustível em um compartimento durante o recebimento é feita pela sua capacidade total, não sendo usualmente praticada a medição de um volume inferior ao do compartimento cheio. Por isso, não se utiliza um compartimento para atender mais de um posto de combustível.

Por essas razões, os veículos-tanque devem portar o documento de calibragem válido para realizar qualquer transporte de líquidos a granel, chamado Certificado de Verificação. A capacidade nominal de cada compartimento é indicada por um dispositivo de referência, que é uma semi-esfera na ponta de uma haste metálica. Conforme o INMETRO (1998), o volume nominal é correspondente ao líquido no nível desta esfera, com uma tolerância de 30mm. Esse dispositivo é popularmente conhecido como “seta”.

Quando o volume está abaixo ou acima da seta, não existe nenhum tipo de escala que indique qual o volume no tanque, apenas sabe-se que é superior ou inferior ao volume nominal.

O padrão estabelecido pelo mercado para os veículos-tanque e, conforme classificação do INMETRO (1993) são:

- Caminhão-tanque, com tanque montado sobre o próprio chassi. Chamado de *truck*, padronizado em três compartimentos de 5.000 litros, totalizando 15.000 litros.
- Tanque-reboque ou semi-reboque, sem propulsão própria, transportados com auxílio de caminhão trator (cavalo). Esses veículos são chamados de *carreta*. Os compartimentados, possuem dois padrões: seis compartimentos de 5.000 litros; ou dois compartimentos de 5.000 litros e dois de 10.000. Ambos padrões totalizando 30.000 litros.

Para o problema aqui proposto, então, foram considerados os três tipos comuns de veículo, conforme a sua compartimentação. Isso também permite identificar o tamanho do veículo. A codificação do tipo de veículo no exemplo criado segue a apresentada no Quadro 3.1:

Quadro 3.1 – Codificação dos Tipos de Caminhões

Tipo	Tamanho	Volume Máximo (litros)	Compartimentação (10 ³ litros)
1	Truck	15.000	5 - 5 - 5
2	Carreta	30.000	5 - 5 - 5 - 5 - 5 - 5
3	Carreta	30.000	10 - 5 - 10 - 5

3.1.4. Alocação da Distribuição

O objetivo da distribuição é realizar as entregas aos clientes, geralmente postos de abastecimento, com o menor custo total. Existem dois componentes tradicionais na determinação do custo total: custos variáveis por veículo e custos fixos por veículo. O custo variável por veículo, neste caso, é diretamente proporcional à quilometragem percorrida e representa a parcela referente ao gasto com combustível, lubrificantes, pneus, desgastes com peças, etc. O custo fixo por veículo é a parcela referente à mão-de-obra alocada disponível, depreciação e remuneração de capital sobre o valor da frota e custos administrativos como seguros, licenciamentos, entre outros.

O objetivo da alocação, por tanto, é realizar as entregas com o menor custo total. Esse custo é representado por uma parcela de custos variáveis, proporcional à soma de distâncias percorridas, e uma parcela de custos fixos, proporcional ao número de veículos alocados. Portanto, o objetivo é minimizar as distâncias percorridas e a quantidade de veículos utilizados.

A frota pode ser própria ou terceirizada. A frota terceirizada é composta por empresas transportadoras especializadas nesse tipo de serviço ou de caminhoneiros autônomos cadastrados pela distribuidora. No caso de frota terceirizada, é usual o custo fixo ser diluído em um valor de frete quilométrico.

Para análise de frota própria, é importante ressaltar que existe um problema anterior ao problema aqui exposto, que é a decisão do tamanho e composição da frota. Não será abordado, portanto, a questão de decisão de tamanho e composição da frota, mas apenas

a otimização de utilização de uma frota pré-determinada. Esse trabalho, no entanto, pode permitir que as alocações mais econômicas revelem, em médio prazo, a real necessidade de frota de um determinado distribuidor.

Em caso de frotas mistas (própria e terceirizada), considera-se a preferência pelo uso da frota própria, pois ela é teoricamente mais barata. Além dela já estar disponível, com parte dos seus custos já consumados e que incidem independentemente do uso, a frota própria não inclui a parcela de lucro do frotista terceiro. A frota utilizada é, então, complementada por veículos terceirizados apenas quando necessário. É prática de algumas empresas do setor, inclusive, a utilização de terceiros para a demanda que excede a capacidade da frota regular própria nos momentos de picos de comercialização.

3.1.5. Tipos de Pedido

Podem existir diversas restrições operacionais no pedido, como, por exemplo:

- Horário de entrega;
- Padrão do veículo;
- Tamanho do veículo.

Com relação ao horário de entrega, pode existir a necessidade de definição de uma janela de tempo para os clientes por razões de horário de funcionamento, disponibilidade de funcionário capacitado ao recebimento e descarregamento do veículo ou restrições de circulação em vias urbanas. Como, na prática, um número muito reduzido de pedidos tem esse tipo de restrição, essa restrição foi desconsiderada com o objetivo de simplificar o problema e sua resolução.

O padrão do veículo é a pintura do veículo pela qual é identificada a distribuidora. As distribuidoras, em geral, exigem da sua rede de postos que o combustível seja exclusivo da própria distribuidora. Porém, na prática, existem postos que compram combustível de outras distribuidoras, exigindo que o veículo esteja identificado como sendo da distribuidora correspondente ao posto.

A consideração ou não dessa restrição deve ser avaliada caso a caso, pois existem distribuidoras que atendem a postos de uma única rede, dispensando, portanto, esse tipo de restrição.

O tamanho do veículo pode ser restrito em postos menores, que não suportam o recebimento por carretas. Neste caso, a entrega deve ser exclusivamente realizada por veículos do tipo *truck*.

3.1.6. *Alocação Manual Empírica*

As distribuidoras geralmente realizam a programação diária das entregas manualmente, de maneira empírica. Preferencialmente, os pedidos devem ser entregues até o dia seguinte. Na prática, no entanto, os prazos dependem de relações comerciais específicas entre a distribuidora e cada posto de combustível. As solicitações são feitas direto para a distribuidora por telefone.

Essa alocação é feita através de pessoal treinado que trabalham utilizando as seguintes informações:

- Localização dos clientes, agrupados por regiões;
- Pedidos já aprovados para entrega, com características de cada pedido;
- Frota disponível, com características de cada veículo.

Durante a alocação, os pedidos mais difíceis de atender, chamados de *especiais*, são resolvidos antes. Isto é, existe prioridade para alocação dos pedidos de acordo com:

- Exigência de padrão de caminhão;
- Horário de entrega programado;
- Distância muito grande.

Em seguida, tenta-se completar a carga desses veículos com outros pedidos da mesma área. As áreas de agrupamento de pedidos são correspondentes às rotas de tráfego usuais. Isto é, são, na verdade, grupos de clientes localizados em cidades posicionadas ao longo de um mesmo eixo rodoviário.

Pontos de atendimento mais distantes tendem a ser atendidos pelos veículos terceirizados, dando prioridade a atendimentos mais próximos através dos veículos próprios. O tempo de ciclo dos veículos próprios torna-se menor, permitindo que realizem um maior número de viagens.

De forma geral, tem-se uma lista de frota, com tipo do veículo, compartimentação e padrão de pintura. A outra lista possui os pedidos, indicados com o volume, o local e as

observações de restrição. O Quadro 3.2 apresenta exemplos de dados disponíveis para a realização da alocação.

Quadro 3.2 – Exemplo de Tabelas para Alocação de Pedidos (Volumes em 1000 l)

Frota				Pedidos		
Tipo	Compart.	Tipo	Pintura	Local	Volume	Observações
Truck	5-5-5	Próprio	A	São Paulo - SP	5.000 l	nenhuma
Carreta	5-10-5-10	Próprio	B	Sorocaba - SP	20.000 l	padrão: A ou B
Truck	5-5-5	Próprio	C	Diadema - SP	15.000 l	horário: 7-12h / só truck
Carreta	5-5-5-5-5-5	Terceiro	A	São Paulo - SP	5.000 l	padrão: só C
...	...	...	...	Itu - SP	5.000 l	só truck
...	...	...	...	...	...	...

Pedidos de uma mesma região, ou rota, são agrupados até completar um veículo completo.

3.2. Modelagem Matemática do Problema

Descreve-se nesta seção a modelagem matemática proposta para o problema da distribuição de combustível a partir de uma base de abastecimento (em geral, um terminal) de uma distribuidora para postos de combustível. Para a realização dessa distribuição, dispõe-se de uma frota heterogênea de caminhões-tanque e é necessário determinar a forma como essa frota disponível será alocada para o atendimento de um conjunto de pedidos.

Mais especificamente, pretende-se definir a alocação ótima dos pedidos aos veículos e a sequência de entrega de cada veículo. Trata-se de um problema de roteirização de veículos com frota heterogênea a partir de uma única base e com restrições especiais de capacidade devidas à compartimentação dos veículos. Os roteiros envolvem um número limitado de paradas, correspondente ao máximo de seis para as carretas, embora na prática um veículo não faça mais que duas a três paradas.

O objetivo da distribuição é realizar as entregas com o menor custo total possível. Como a frota é heterogênea essa solução deve levar em consideração que cada veículo tem um custo diferente associado, além da questão de utilizar frota própria ou de terceiros.

3.2.1. Modelagem dos Caminhões-Tanque

Além do padrão de pintura e do tipo (tamanho e compartimentação) dos veículos, é importante considerar que parte da frota é própria da distribuidora e parte corresponde a veículos terceirizados. Os veículos que compõem a frota própria possuem custos fixos maiores e custos variáveis menores. Veículos terceirizados, no entanto, possuem custo variável elevado, uma vez que o frete dos serviços prestados deve cobrir toda remuneração, inclusive a referente aos custos fixos nos momentos de ociosidade. Desse modo, busca-se a maior utilização possível de veículos próprios, uma vez que a ociosidade destes é indesejável para a distribuidora, pois incidem custos, ao contrario dos veículos de terceiros. Por outro lado, caso a frota própria não consiga atender toda a demanda é preciso determinar qual a melhor alocação e utilização das frotas própria e terceirizada, de tal modo a minimizar o custo total de distribuição.

O conjunto de veículos disponíveis é dado por $V = \{1, 2, \dots, |V|\}$.

Os veículos $v \in V$ são, então, diferenciados através de três características principais:

- Padrão de Pintura;
- Propriedade (próprio ou terceirizado);
- Tipo e tamanho de veículo.

O padrão de pintura interfere diretamente na possibilidade ou não de um determinado veículo atender um determinado cliente. Conforme mencionado anteriormente, pode ser uma restrição não ativa em alguns problemas.

A identificação se o veículo é próprio ou terceirizado e o tipo e tamanho de veículo têm implicações diretas nos custos fixos e variáveis que são atribuídos à rodagem de cada veículo.

Assim, como este caso configura-se com frota heterogênea, essas características (propriedade e tipo de veículo) levam à necessidade de atribuir um custo fixo para cada veículo $v \in V$, dado por cf_v .

De modo semelhante, define-se o custo variável de rodagem para cada veículo $v \in V$ como cv_v .

A atribuição de custo fixo aos veículos permite maior flexibilidade no uso do modelo, permitindo também que ele auxilie a decisão sobre o melhor uso de frota. No caso da

decisão ser operacional, isto é, programação diária dos roteiros de entregas a partir de veículos já alocados à operação, os custos fixos da frota própria podem ser desconsiderados, uma vez que os veículos já estão disponíveis para uso, não incorrendo nenhum custo adicional para tanto. Neste caso, o modelo matemático proposto permite ainda definir em que rotas é melhor usar a frota própria e quais devem ser alocadas a terceiros, caso a frota própria não consiga atender todas as entregas programadas.

Conforme afirmado anteriormente, os compartimentos presentes nos veículos podem ser de 5.000 ou de 10.000 litros. Uma vez que o volume de cada pedido é sempre múltiplo de 5.000 litros, a identificação da quantidade de compartimentos de 5.000 litros dos veículos é importante na modelagem para garantir que uma quantidade de 5.000 litros não seja alocada a um compartimento de 10.000 litros.

Por exemplo, para transportar um pedido de 5.000, 15.000 ou 25.000 litros é necessário assegurar que o veículo disponha, além da capacidade total, de pelo menos um compartimento de 5.000 litros. Por outro lado, um pedido de 10.000, 20.000 ou 30.000 litros, no entanto, apenas necessita da disponibilidade de seu volume total para que seja possível sua alocação, sem qualquer restrição quanto ao tamanho dos compartimentos. Essa percepção evita que seja necessária a modelagem da alocação de cada pedido a compartimentos específicos. Assim, apenas o monitoramento da necessidade de uso dos compartimentos de 5.000 litros garante que a alocação de pedidos a um caminhão esteja adequada e seja viável.

Por isso, define-se $V5_v$, indicando a quantidade de compartimentos de 5.000 litros de cada veículo $v \in V$.

Outra característica importante na modelagem é o volume máximo, ou capacidade, admitido (em litros) em cada caminhão $v \in V$, dada por $Vmax_v$.

3.2.2. Modelagem dos Clientes e da Base

Cada cliente corresponde a um posto de combustível capaz de realizar um pedido de um tipo de combustível para a distribuidora. As seguintes características dos clientes foram consideradas para solução do problema:

- Localização geográfica;
- Tipos de caminhões que podem atender.

O conjunto de clientes é dado por $\{1, 2, \dots, |I|\}$. Adiciona-se ao conjunto o ponto $\{0\}$, correspondendo à base de distribuição de onde partem os veículos. Assim, define-se $I = \{1, 2, \dots, |I|\} \cup \{0\}$, onde $\{0\}$ corresponde à base de distribuição.

Define-se um grafo $G = (N, A)$ onde N é o conjunto de nós do problema I , enquanto A é o conjunto de arcos ou ligações entre os nós.

A localização real do cliente corresponde a um ponto numa malha viária, que define um nó $i \in N$ no grafo G . Como se considera que o caminhão sempre utilizará o traçado de caminho mínimo ou de menor impedância, pode-se determinar os arcos $(i, j) \in A$ da rede e a matriz de distâncias entre os clientes, obtida pela aplicação de um algoritmo de caminho mínimo entre todos os pares de nós do grafo G . Um código-fonte para esse algoritmo em C++ é apresentado por Sedgewick (2002).

Assim, assume-se serem conhecidas todas as distâncias d_{ij} entre clientes $i, j \in I$ que correspondem a arcos do grafo.

A outra informação importante para caracterização de cada cliente é o tipo de veículo que pode atendê-lo. Conforme mencionado anteriormente, existe a limitação do padrão de pintura e do tamanho do veículo.

O padrão do veículo é definido pela pintura do mesmo veículo a qual identifica a distribuidora. Algumas distribuidoras exigem da sua rede de postos que o combustível seja exclusivo da própria distribuidora. Muitos postos comprem combustível de outras distribuidoras, mas exigem que o veículo esteja identificado como sendo da distribuidora correspondente ao posto. Conseqüentemente, decidiu-se pela incorporação dessa restrição, pois é fator determinante na viabilidade prática da solução do problema para muitas distribuidoras, particularmente as menores. Essa característica, no entanto, não é estendida a todas as operações dessa natureza. A maior parte das distribuidoras operam apenas com veículos de um único padrão.

O tamanho do veículo pode ser restrito em postos menores, com área reduzida, que não suportam o recebimento por carretas. Neste caso, a entrega deve ser exclusivamente realizada por veículos menores.

Assim, a fim de considerar as condicionantes de padrão de veículo e tamanho, para cada veículo é indicado se é possível ou não o atendimento de um determinado pedido

através de uma matriz de coeficientes binários, que representa o conjunto dessas restrições. Mais especificamente, para cada cliente $i \in I - \{0\}$, temos a possibilidade ou impedimento do mesmo ser atendido por determinado veículo $v \in V$ da frota, dada por:

$$R_{iv} = \begin{cases} 1, & \text{se o veículo } v \text{ pode atender o ponto } i \\ 0, & \text{caso contrário} \end{cases}$$

3.2.3. Modelagem dos Pedidos

A modelagem proposta considera apenas os pedidos que a distribuidora deve entregar, desconsiderando, por motivos óbvios, eventuais pedidos retirados pelos clientes na própria distribuidora utilizando veículo próprio ou de terceiros.

Cada solicitação de um cliente indica uma combinação de quantidades de derivados de petróleo a serem entregues. Por exemplo: 15.000 litros de gasolina, 5.000 litros de álcool e 10.000 litros de óleo diesel. Os pedidos são sempre padronizados, como múltiplos de 5.000 litros e a quantidade máxima de 30.000 litros.

Dada a grande dificuldade de modelagem dos pedidos compostos de quantidades de diferentes combustíveis, decidiu-se pela representação dos pedidos de forma simplificada. Por exemplo, uma solicitação que envolva gasolina, álcool e óleo diesel, cuja soma seja maior que a capacidade de um caminhão, não é possível, a priori, saber qual desses combustíveis deveria ser agrupado dos demais que resultasse na solução ótima. Então, pedidos formados por grupos de pedidos teriam que possuir uma estrutura de dados que permitisse seu agrupamento e desagrupamento. O algoritmo, por sua vez, também deveria ser capaz de testar as várias hipóteses. Deve-se destacar que, na prática, pedidos contendo múltiplos itens e quantidades elevadas são raros e podem ser tratados separadamente, dada a sua complexidade e a necessidade de um ou mais veículos exclusivos para atendê-los.

Assim, a representação dos pedidos foi, então, determinada através de pedidos desmembrados que nunca podem ser alocados em um mesmo compartimento. Em outras palavras, cada pedido corresponde a um único tipo de combustível. Por exemplo, um pedido de um cliente correspondente a 10.000 litros de gasolina e 5.000 litros de álcool é decomposto e representado por dois pedidos separados, um pedido de 5.000

litros e outro de 10.000 litros, ambos destinados para um mesmo local ou cliente. Isso facilita a modelagem da alocação dos pedidos aos veículos com diferentes compartimentos. Ressalta-se que, então, pode eventualmente ocorrer de um cliente receber pedidos divididos em vários caminhões.

Assim, a representação dos pedidos foi simplificada a duas características:

- Identificação do cliente;
- Volume do pedido, ou seja, a quantidade de um tipo de combustível a ser entregue, que deve ser transportada em um ou mais compartimentos cheios de um veículo.

O conjunto de pedidos é dado por $P = \{1, 2, \dots, |P|\}$.

Para cada pedido $p \in P$ é conhecido o cliente $Cliente_p$, ao qual o pedido deve ser entregue, tal que $Cliente_p \in I$.

Para evitar que algum pedido não possa ser atendido, um pré-processamento deve evitar que existam pedidos maiores que a máxima capacidade do maior caminhão (30.000 litros, no caso estudado). Caso exista um pedido maior, deve-se separar em pedidos maiores possíveis e um único pedido menor. Por exemplo, um pedido de 35.000 litros decomposto em dois pedidos para um mesmo cliente: um de 30.000 litros e outro de 5.000 litros.

Adotou-se, como premissa para o modelo aqui proposto, não considerar a otimização da divisão de pedidos maiores que 30.000 litros. A decisão da forma de dividir esses pedidos pode afetar o resultado da alocação final e é um problema adicional bastante complexo por si só, que se sugere contemplar em uma abordagem futura.

Assim, determinamos o volume de cada pedido $p \in P$ como Vol_p .

Conforme explicado anteriormente, é necessário observar se, por seu volume, o pedido necessita do uso de um compartimento de 5.000 litros ou se ele pode igualmente ser atendido por compartimentos de 5.000 e 10.000 litros. Assim, pedidos de 5.000, 15.000 e 25.000 litros necessitam de, ao menos, um compartimento de 5.000 litros para serem alocados. Essa identificação é feita para cada pedido $p \in P$ através de:

$$\blacksquare \quad R5_p = \begin{cases} 1, & \text{se necessita o uso de compartimento de 5.000 l} \\ 0, & \text{caso contrário} \end{cases}$$

3.2.4. Variáveis de Decisão

Para cada arco que liga a base e os clientes, define-se uma variável binária que indica se o caminhão $v \in V$ passa pelo arco $(i, j) \in A$, com $i, j \in I$, da seguinte forma:

$$\blacksquare \quad X_{ij}^v = \begin{cases} 1, & \text{se } j \text{ é visitado imediatamente após } i \text{ pelo veículo } v \\ 0, & \text{caso contrário} \end{cases}$$

Para cada pedido e veículo existe uma variável binária que indica se o pedido $p \in P$ está alocado no veículo $v \in V$:

$$\blacksquare \quad Y_{pv} = \begin{cases} 1, & \text{se o pedido } p \text{ é alocado ao veículo } v \\ 0, & \text{caso contrário} \end{cases}$$

Para cada arco $(i, j) \in A$ e cada veículo $v \in V$, define-se também uma variável de fluxo para evitar a criação de sub-rotas, como será visto adiante (restrições (9) e (10)).

$$\blacksquare \quad Z_{ij}^v \geq 0$$

3.2.5. Função objetivo

A função objetivo corresponde à minimização do custo total, composto pelas parcelas de custo fixo dos veículos e custo variável proporcional à distância total percorrida.

$$\blacksquare \quad \min \sum_{v \in V} \sum_{j \in J} \left(cf_v \times X_{0j}^v + \sum_{i \in I} (cv_v \times d_{ij} \times X_{ij}^v) \right) \quad (1)$$

3.2.6. Restrições

Cada um dos pedidos $p \in P$ deve ser alocado a um e apenas um único veículo:

$$\sum_{v \in V} Y_{pv} = 1 \quad \forall j = \text{Cliente}_p, \forall p \in P \quad (2)$$

Nenhum veículo pode ter pedidos alocados que necessitem de mais compartimentos de 5.000 litros maior que o número desses compartimentos disponíveis em cada veículo $v \in V$:

$$\sum_{p \in P} (R5_p \times Y_{pv}) \leq V5_v \quad \forall v \in V \quad (3)$$

Nenhum veículo $v \in V$ pode ter sua capacidade máxima superada pela quantidade de volume total alocado:

$$\sum_{p \in P} (Vol_p \times Y_{pv}) \leq Vmax_v \quad \forall v \in V \quad (4)$$

Para todo pedido $p \in P$, o cliente $j = \text{Cliente}_p$, quando alocado a um caminhão $v \in V$, precisa necessariamente ser visitado por este caminhão.

$$\sum_{i \in I} X_{ij}^v \geq Y_{pv} \quad \forall p \in P, \forall v \in V, \forall j \in I \quad (5)$$

Cada um dos veículos $v \in V$ deve respeitar as restrições de tamanho e de padrão de veículo permitido para cada cliente $j \in I$:

$$\sum_{i \in I} X_{ij}^v \leq R_{jv} \quad \forall j \in I - \{0\}, \forall v \in V \quad (6)$$

Cada ponto $k \in I$ da malha no qual chega um determinado veículos $v \in V$, deve sair o mesmo veículo, garantindo a continuidade de fluxo das rotas determinadas:

$$\sum_{i \in I} X_{ik}^v - \sum_{j \in J} X_{kj}^v = 0 \quad \forall k \in I, \forall v \in V \quad (7)$$

Cada veículo $v \in V$ pode fazer apenas uma rota, isto é, ele pode deixar a base $\{0\} \in I$ apenas uma única vez.

$$\sum_{j \in I - \{0\}} X_{0j}^v \leq 1 \quad \forall v \in V \quad (8)$$

As condições de continuidade e de realização de uma única rota não garantem por si só que o conjunto de soluções representem rotas válidas. Soluções inválidas que aqui se refere são aquelas que foram as chamadas sub-rotas (*subtours*). Entre as formulações existentes na literatura para proibir esse fenômeno, foi escolhida a formulação proposta por Gavish e Graves (1978) *apud* Bodin *et al.* (1983), conforme ilustrado no Apêndice A, que introduz uma variável de decisão Z_{ij}^v que, atrelada pela equação (10) aos arcos que compõem a solução, cria um sequência de valores válida para todos os pontos exceto a base de veículos $\{0\} \in I$.

Porém, no caso estudado faz-se necessária uma modificação em relação à formulação originalmente proposta. Os autores determinaram equações para solução do Problema do Caixeiro Viajante, no qual um determinado veículo percorre todos os pontos da rede. No caso aqui estudado, o veículo apenas percorre alguns pontos e estes pontos são indicados pela variável de decisão X_{ij}^v .

Ou seja, para pontos não selecionados como parte da rota de um veículo, as variáveis que sequenciam os arcos serão nulas em torno desse ponto, assim como a diferença em torno.

Assim, em cada conjunto de arcos que são atendidos por cada veículo $v \in V$, tem-se:

$$\sum_{j \in I} Z_{kj}^v - \sum_{i \in I} Z_{ik}^v = -1 \times \sum_{j \in I} X_{kj}^v \quad \forall k \in I - \{0\}, \forall v \in V \quad (9)$$

$$Z_{ij}^v \leq M \times X_{ij}^v \quad \forall v \in V, \forall i, \forall j \in I, M \geq |I| \quad (10)$$

Define-se M como uma constante suficientemente grande. Na prática, basta assegurar que $M \geq |I|$.

Por fim, definem-se as variáveis de decisão $X_{i,j}^v$ e $Y_{p,v}$ como binárias e $Z_{i,j}^v$ como reais não-nulos.

$$\begin{array}{ll} \blacksquare & X_{i,j}^v \in \{0,1\} \quad \forall v \in V, \forall i, \forall j \in I \end{array} \quad (11)$$

$$\begin{array}{ll} \blacksquare & Y_{p,v} \in \{0,1\} \quad \forall p \in P, \forall v \in V \end{array} \quad (12)$$

$$\begin{array}{ll} \blacksquare & Z_{i,j}^v \geq 0 \quad \forall v \in V, \forall i, \forall j \in I \end{array} \quad (13)$$

3.2.7. Sumário da Modelagem Matemática

A ordem de grandeza de variáveis da formulação matemática exata é:

$$\blacksquare \quad 2 \times (|I|^2 - |I|) \times |V| + |P| \times |V|$$

Sendo:

- $|P|$ a quantidade de pedidos;
- $|V|$ a quantidade de veículos;
- $|I|$ a quantidade de clientes mais a base.

A seguir, apresenta-se um resumo da modelagem matemática apresentada anteriormente.

Função Objetivo:

$$\blacksquare \quad \min \sum_{i \in V} \sum_{j \in J} \left(cf_v \times X_{0j}^v + \sum_{i \in I} (cv_v \times d_{ij} \times X_{ij}^v) \right) \quad (1)$$

Restrições:

$$\blacksquare \quad \sum_{i \in V} Y_{p,v} = 1 \quad \forall p \in P \quad (2)$$

$$\blacksquare \quad \sum_{p \in P} (R5_p \times Y_{p,v}) \leq V5_v \quad \forall v \in V \quad (3)$$

$$\sum_{p \in P} (Vol_p \times Y_{pv}) \leq Vmax_v \quad \forall v \in V \quad (4)$$

$$\sum_{i \in I} X_{ij}^v \geq Y_{pv} \quad j = Cliente_p, \forall p \in P, \forall v \in V \quad (5)$$

$$\sum_{i \in I} X_{ij}^v \leq R_{jv} \quad \forall j \in I - \{0\}, \forall v \in V \quad (6)$$

$$\sum_{i \in I} X_{ik}^v - \sum_{j \in I} X_{kj}^v = 0 \quad \forall k \in I, \forall v \in V \quad (7)$$

$$\sum_{j \in I - \{0\}} X_{0j}^v \leq 1 \quad \forall v \in V \quad (8)$$

$$\sum_{j \in I} Z_{kj}^v - \sum_{i \in I} Z_{ik}^v = -1 \times \sum_{j \in I} X_{kj}^v \quad \forall k \in I - \{0\}, \forall v \in V \quad (9)$$

$$Z_{ij}^v \leq M \times X_{ij}^v \quad \forall v \in V, \forall i, \forall j \in I, M \geq |I| \quad (10)$$

3.3. Considerações Finais

Buscou-se aqui demonstrar a estrutura geral do problema de roteirização de entregas de petróleo, suas condicionantes e ainda contribuir através da elaboração de um modelo matemático para o problema. Esta modelagem corresponde a uma instância de um problema mais geral de roteirização de veículos: a roteirização de entregas a partir de uma base única, com o uso de uma frota heterogênea, veículos compartimentados e com restrições adicionais para determinado veículo atender determinado cliente.

Como descrito anteriormente, problemas de roteirização de veículos são de classe NP-difícil (Cunha, 2000), o que significa que o esforço computacional cresce exponencialmente com o tamanho do problema. Portanto, torna-se impossível resolver através de métodos exatos utilizando pacotes comerciais. A implementação da modelagem matemática em pacotes comerciais de programação linear inteira mista e a resolução de pequenas instâncias de problema são apresentados no Capítulo 4 a seguir.

Coloca-se, então, a necessidade de desenvolver uma heurística que permita a solução de instâncias encontradas na prática. No Capítulo 5 é apresentada a estratégia de solução proposta.

4. DETERMINAÇÃO DA SOLUÇÃO EXATA

Como foi visto anteriormente no Capítulo 3, o problema de roteirização de veículos apresenta complexidade exponencial, o que impede a solução de instâncias reais utilizando pacotes comerciais de programação linear inteira, tais como o *ILOG CPLEX*, *GAMS*, etc, requerendo o uso de métodos heurísticos.

A fim de validar a modelagem matemática proposta no capítulo anterior (seção 3.2) e permitir a verificação dos resultados da heurística de otimização proposta no próximo capítulo, foram elaboradas duas implementações do modelo matemático utilizando dois pacotes comerciais de programação linear inteira mista.

O primeiro modelo, em planilha eletrônica, teve como objetivo auxiliar na elaboração e validar a formulação matemática proposta, aproveitando a característica visual desses modelos com a facilidade de estruturação, e está descrito na seção 4.1.

Para esta implementação foi criada uma instância de pequeno porte para auxiliar na validação do modelo matemático e permitindo a conferência manual da solução obtida em termos das restrições do modelo e da solução ótima da função objetivo. Essa instância possui 10 pedidos, 4 clientes e 4 veículos, como será detalhado adiante.

O segundo modelo, apresentado na seção 4.2, foi formulado para servir de teste na solução do problema aqui apresentado e, caso fosse possível, serviria para comparar com os resultados do algoritmo heurístico proposto no Capítulo 5.

Nesta segunda implementação, foi inicialmente buscada a solução da mesma instância utilizado no teste da planilha eletrônica. Em seguida, foram criadas e testadas as soluções de outras instâncias, progressivamente maiores, apresentadas na seção 4.3.

Ambas as implementações utilizam as mesmas fórmulas e nomenclaturas das variáveis indicadas anteriormente na seção 3.2.

4.1. Solução Utilizando o *Solver* para Planilha Eletrônica

O modelo matemático desenvolvido foi implementado em planilha eletrônica para ser resolvido através do aplicativo de otimização *Solver* da *Frontline Systems* e distribuído com o *Microsoft Excel*. O *Solver*, entre outras funções, permite obter a solução ótima para problemas de programação linear e linear inteira.

A versão utilizada permite apenas a solução de problemas de até 200 variáveis de decisão. Dessa forma, foi proposto o maior problema que pudesse ser resolvido com o *Solver*, limitado a 200 variáveis de decisão. Foi, então, gerado um problema de distribuição de 10 pedidos para 4 possíveis clientes, utilizando 4 veículos.

4.1.1. Dados de Entrada

Apenas com função ilustrativa adotou-se, nas figuras, as células em amarelo correspondem aos dados de entrada, as azuis aos valores calculados, em vermelho os valores sujeitos a restrições e em verde as variáveis de decisão do problema.

Os dados de entrada foram implementados na planilha eletrônica conforme as figuras apresentadas nesta seção. Na Figura 4.1 são apresentados os dados de cada veículo v : custo fixo, custo variável, quantidade de compartimentos de 5.000 litros e capacidade total (em milhares de litro).

Veículos

$V=\{1, 2, \dots, |V|\}$

v	cf [v]	cv [v]	V5 [v]	Vmax [v]
1	10	1	2	30
2	10	1, 1	6	30
3	10	0, 75	3	15
4	10	1	2	30

Figura 4.1 – Dados dos Veículos

A Figura 4.2 indica a implementação dos dados referentes à matriz de distância entre a base (ponto 0) e os clientes (i e j) e entre os próprios clientes.

Clientes e Base

$I=\{0\}+\{1, 2, \dots, |I|\}$

$[k] = 0 = \text{Base}$

$d[i][j] = (\text{distância de } i, j)$

$i \setminus j \rightarrow$	0	1	2	3	4
0	0	1	1, 4	2	2, 8
1	1	0	2	3	3, 6
2	1, 4	2	0	3, 6	3
3	2	3	3, 6	0	2
4	2, 8	3, 6	3	2	0

Figura 4.2 – Dados de Distância

Na Figura 4.3 apresentam-se os dados referentes às restrições de permissão de cada veículo v atender cada cliente i .

Restrições

$R[i][v] = (1, \text{ se } i \text{ aceita } v), (0, \text{ c.c.})$

$i \backslash v \rightarrow$	1	2	3	4
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1

Figura 4.3 – Dados de Restrição entre Clientes e Veículos

Na Figura 4.4 estão apresentados os dados de cada pedido p : para qual cliente é o pedido e o volume pedido. A partir do volume, é calculado se ele necessita ou não de um compartimento de 5.000 litros.

Pedidos

$P=\{1, 2, \dots, |P|\}$

p	Cliente [p]	Vol [p]	R5 [p]
1	1	10	0
2	1	25	1
3	1	5	1
4	2	5	1
5	2	10	0
6	2	10	0
7	3	5	1
8	3	5	1
9	4	5	1
10	4	10	0

Figura 4.4 – Dados dos Pedidos

4.1.2. Variáveis de Decisão

Nesta seção, as figuras a seguir apresentam a implementação das variáveis de decisão na planilha eletrônica.

A Figura 4.5 apresenta as variáveis de decisão X_{ij}^v , enquanto as variáveis Y_{pv} se encontram na Figura 4.6. Por fim, a Figura 4.7 demonstra a implementação das variáveis Z_{ij}^v em planilha eletrônica.

Variáveis de Decisão

$X[v][i][j] = (1, \text{ se } v \text{ passa por } i,j) \text{ ou } (0, \text{ c.c.})$

Binárias						
$v=1 \mid i \setminus j \rightarrow$	0	1	2	3	4	
0	0	1	0	0	0	
1	1	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

$v=2 \mid i \setminus j \rightarrow$	0	1	2	3	4	
0	0	0	0	1	0	
1	0	0	0	0	0	
2	1	0	0	0	0	
3	0	0	0	0	1	
4	0	0	1	0	0	

$v=3 \mid i \setminus j \rightarrow$	0	1	2	3	4	
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

$v=4 \mid i \setminus j \rightarrow$	0	1	2	3	4	
0	0	1	0	0	0	
1	0	0	1	0	0	
2	1	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

Figura 4.5 – Variáveis de Decisão X

$Y[p][v] = (1, \text{ se } p \text{ está em } v) \text{ ou } (0, \text{ c.c.})$

Binárias				
$p \backslash v \rightarrow$	1	2	3	4
1	0	0	0	1
2	1	0	0	0
3	1	0	0	0
4	0	1	0	0
5	0	0	0	1
6	0	0	0	1
7	0	1	0	0
8	0	1	0	0
9	0	1	0	0
10	0	1	0	0

Figura 4.6 – Variáveis de Decisão Y

$Z[v][i][j] = (0) \text{ ou } (1) \text{ para evitar sub-rotas}$

Não-Negativo					
$v=1 \mid i \backslash j \rightarrow$	0	1	2	3	4
0	0,0	1,0	0,0	0,0	0,0
1	0,0	0,0	0,0	0,0	0,0
2	0,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	0,0
4	0,0	0,0	0,0	0,0	0,0

$v=2 \mid i \backslash j \rightarrow$	0	1	2	3	4
0	0,0	0,0	0,0	3,0	0,0
1	0,0	0,0	0,0	0,0	0,0
2	0,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	2,0
4	0,0	0,0	1,0	0,0	0,0

$v=3 \mid i \backslash j \rightarrow$	0	1	2	3	4
0	0,0	0,0	0,0	0,0	0,0
1	0,0	0,0	0,0	0,0	0,0
2	0,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	0,0
4	0,0	0,0	0,0	0,0	0,0

$v=4 \mid i \backslash j \rightarrow$	0	1	2	3	4
0	0,0	2,0	0,0	0,0	0,0
1	0,0	0,0	1,0	0,0	0,0
2	0,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	0,0
4	0,0	0,0	0,0	0,0	0,0

Figura 4.7 – Variáveis de Decisão Z

4.1.3. Restrições

As restrições do modelo matemático proposto foram implementadas na planilha eletrônica conforme as figuras que ilustram esta seção.

A Figura 4.8 ilustra a restrição de cada pedido estar alocado a um único veículo.

Pedido x Veículo

p	f [p]	=
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1

Figura 4.8 – Restrição de Alocação de Pedidos a Veículos

A restrição que verifica a quantidade de pedidos que necessitam de compartimentos de 5.000 litros não excede o disponível aparece representada na Figura 4.9.

Compartimentos de 5

v	f [v]	<=
1	2	2
2	4	6
3	0	3
4	0	2

Figura 4.9 – Restrição de Compartimentos de 5000 litros

A Figura 4.10 a seguir indica a restrição de nenhum veículo pode ter sua capacidade total excedida.

Volume Máximo

v	f[v]	<=
1	30	30
2	30	30
3	0	15
4	30	30

Figura 4.10 – Restrição de Volume Máximo

A restrição que impõe que cada veículo deve passar pelos clientes que lhe estão alocados encontra-se na Figura 4.11.

Caminhão atende Cliente do Pedido

p \ v →	1	2	3	4	>=	>=	>=	>=
1	1	0	0	1	0	0	0	1
2	1	0	0	1	1	0	0	0
3	1	0	0	1	1	0	0	0
4	0	1	0	1	0	1	0	0
5	0	1	0	1	0	0	0	1
6	0	1	0	1	0	0	0	1
7	0	1	0	0	0	1	0	0
8	0	1	0	0	0	1	0	0
9	0	1	0	0	0	1	0	0
10	0	1	0	0	0	1	0	0

Figura 4.11 – Restrição de Alocação de Veículos a Cliente

A Figura 4.12 ilustra a restrição que cada veículo deve respeitar a restrições de atender cada cliente.

Veículo respeita Cliente

v \ i →	1	2	3	4	<=	<=	<=	<=
2	1	0	0	1	1	1	1	1
3	0	1	0	1	1	1	1	1
4	0	1	0	0	1	1	1	1
5	0	1	0	0	1	1	1	1

Figura 4.12 – Restrição entre Veículos e Clientes

A continuidade das rotas é garantida pela restrição ilustrada na Figura 4.13.

Continuidade das Rotas

$\downarrow k \backslash v \rightarrow$	1	2	3	4	=	=	=	=
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0

Figura 4.13 – Restrição de Continuidade das Rotas

A Figura 4.14 apresenta a restrição de cada veículo realizar apenas uma rota.

Veículo faz no máximo uma Rota

v	f [v]	<=
1	1	1
2	1	1
3	0	1
4	1	1

Figura 4.14 – Restrição de Máximo de Rotas

Para a eliminação de sub-rotas foram elaboradas na planilha eletrônica as restrições que se encontram na Figura 4.15 e na Figura 4.16 a seguir.

Evitar sub-rotas - Primeira restrição

$\downarrow k \backslash v \rightarrow$	1	2	3	4	=	=	=	=
1	-1,0	0,0	0,0	-1,0	-1,0	0,0	0,0	-1,0
2	0,0	-1,0	0,0	-1,0	0,0	-1,0	0,0	-1,0
3	0,0	-1,0	0,0	0,0	0,0	-1,0	0,0	0,0
4	0,0	-1,0	0,0	0,0	0,0	-1,0	0,0	0,0

Figura 4.15 – Primeira Restrição para Evitar Sub-Rotas

Evitar sub-rotas - Segunda restrição

$v=1 \mid i \setminus j \rightarrow$	0	1	2	3	4	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
0	0,0	1,0	0,0	0,0	0,0	0,0	4,0	0,0	0,0	0,0
1	0,0	0,0	0,0	0,0	0,0	4,0	0,0	0,0	0,0	0,0
2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

$v=2 \mid i \setminus j \rightarrow$	0	1	2	3	4	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
0	0,0	0,0	0,0	3,0	0,0	0,0	0,0	0,0	4,0	0,0
1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	0,0	0,0	0,0	0,0	0,0	4,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	2,0	0,0	0,0	0,0	0,0	4,0
4	0,0	0,0	1,0	0,0	0,0	0,0	0,0	4,0	0,0	0,0

$v=3 \mid i \setminus j \rightarrow$	0	1	2	3	4	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

$v=4 \mid i \setminus j \rightarrow$	0	1	2	3	4	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
0	0,0	2,0	0,0	0,0	0,0	0,0	4,0	0,0	0,0	0,0
1	0,0	0,0	1,0	0,0	0,0	0,0	0,0	4,0	0,0	0,0
2	0,0	0,0	0,0	0,0	0,0	4,0	0,0	0,0	0,0	0,0
3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

Figura 4.16 – Segunda Restrição para Evitar Sub-Rotas

4.1.4. Resultados Obtidos Através do Solver

Conforme visto anteriormente, o problema de distribuição testado possui 10 pedidos a serem entregues a partir de uma base para 4 possíveis clientes, com a disponibilidade de 4 diferentes veículos.

O *Solver* do *Excel* levou 8 horas para encontrar a solução ótima do problema ilustrado em um computador Pentium IV / 3.06 GHz. Computacionalmente, podemos perceber que é gasto um tempo muito grande para resolver um problema muito simples. O valor da função objetivo atingido para essa instância corresponde a um custo total de \$45,64.

Foi atingido o limite de variáveis possíveis para resolução na versão do aplicativo utilizado, mesmo para um problema que não representa a ordem de grandeza de problemas reais.

A solução ótima do problema é indicada na Tabela 4.1. Esta tabela indica quanto combustível foi alocado a cada caminhão para entregar a cada cliente.

Tabela 4.1 – Resultado da Implementação no *Solver*

Cliente	Volume Alocado (1.000 l)				
	Veículo 1	Veículo 2	Veículo 3	Veículo 4	Total
1	30	0	0	10	40
2	0	5	0	20	25
3	0	10	0	0	10
4	0	15	0	0	15
Total	30	30	0	30	

Esta solução indica que cada cliente recebe exatamente os pedidos que lhe foram solicitados, que pode ser observado também nas restrições indicadas na seção anterior na Figura 4.11 e na Figura 4.12.

Os caminhões passam pelos clientes que foram alocados para atender, como pode ser facilmente percebido nos resultados das variáveis de decisão X_{ij}^v apresentadas na Figura 4.5. O veículo 1 sai da base (ponto 0) até o cliente 1 e retorna até a base. O veículo 2 faz o roteiro saindo da base passando, na ordem, pelos clientes 3, 4 e 2, e volta para a base. O veículo 3 não é utilizado. Por fim, o veículo 4 parte da base para o cliente 1, cliente 2 e retorna à base.

Não foi possível encontrar nenhuma rota mais eficientes que as encontradas pelo aplicativo para atender o problema proposto.

Com base nos resultados da solução ótima obtida, indicados na Figura 4.5, também pode-se verificar que não houve formação de sub-rotas.

A Tabela 4.1 também demonstra, que os volumes alocados respeitam a capacidade total dos caminhões, como apresentado antes na Figura 4.10. Pela Figura 4.9 pode-se perceber que a quantidade de compartimentos de 5.000 litros de cada veículo são suficientes para atender os pedidos alocados.

Percebe-se que foi possível a utilização de apenas três caminhões para atender os pedidos, sendo que esse era o número mínimo de caminhões, sendo que os pedidos totalizavam 90.000 litros. Esse volume foi atendido por três caminhões de 30.000 litros, não sendo utilizado um veículo com capacidade de 15.000 litros. Se este caminhão menor fosse incorporado à solução, não seria possível o uso de apenas três veículos. Como o custo fixo é alto, percebe-se que não existe solução com quatro caminhões melhor que a solução com três.

Pode-se afirmar, então, que os objetivos de auxiliar na formulação da modelagem matemática e verificar sua consistência foram atingidos.

4.2. Solução Utilizando *Concert Technology* do *ILOG CPLEX* em C++

Conforme discutido anteriormente, o problema proposto evidencia ser de natureza combinatória, uma vez que é um caso particular de problemas de roteirização, considerando como sendo pertencentes à classe NP-difícil, que requer o uso de métodos aproximados ou heurísticos para a sua solução.

A fim de avaliar os resultados da heurística para instâncias maiores que a possível de ser modelada em planilha eletrônica usando o *Solver*, o modelo matemático foi implementado utilizando *Concert Technology* para *ILOG CPLEX* na linguagem C++.

4.2.1. Implementação

O código-fonte em C++ corresponde à implementação do modelo matemático proposto para o problema de distribuição de derivados de petróleo no pacote *ILOG CPLEX* apresentado no Apêndice B, na Figura B.1.

O modelo implementado segue a mesma estrutura de linguagem e introduz no modelo as mesmas equações determinadas na Modelagem Matemática (seção 3.2) deste trabalho.

4.2.2. Exemplo de Solução Obtida pelo *ILOG CPLEX*

A mesma instância do problema implementado em planilha eletrônica foi colocado para permitir a avaliação do correto funcionamento do algoritmo formulado com o uso do

ILOG CPLEX. O seu resultado será exposto aqui para ilustrar as informações obtidas através do aplicativo.

O problema é o mesmo problema de distribuição de 10 pedidos para 4 clientes, com disponibilidade de 4 veículos, apresentado na seção 4.1. O resultado pode ser visto na Figura 4.17 a seguir.

```

Selected objective sense: MINIMIZE
Selected objective name: obj
Selected RHS name: rhs
Selected bound name: bnd

Problem 'distcomb100.mps' read.
Read time = 0.06 sec.
Tried aggregator 1 time.
MIP Presolve eliminated 22 rows and 25 columns.
Reduced MIP has 192 rows, 215 columns, and 953 nonzeros.
Presolve time = 0.04 sec.
Clique table members: 43
MIP emphasis: balance optimality and feasibility
Root relaxation solution time = 0.02 sec.

Flow cuts applied: 17
Gomory fractional cuts applied: 12

Integer optimal solution: Objective = 4.5640000000e+001
Solution time = 6.16 sec. Iterations = 63519 Nodes = 4544

Variable Name          Solution Value
y[0,0]                  1.000000
y[1,3]                  1.000000
y[2,3]                  1.000000
y[3,1]                  1.000000
y[4,0]                  1.000000
y[5,0]                  1.000000
y[6,1]                  1.000000
y[7,1]                  1.000000
y[8,1]                  1.000000
y[9,1]                  1.000000
x[0,2,1]                1.000000
x[3,0,1]                1.000000
x[0,0,2]                1.000000
x[1,4,2]                1.000000
x[1,0,3]                1.000000
x[1,3,4]                1.000000
x[0,1,0]                1.000000
x[1,2,0]                1.000000
x[3,1,0]                1.000000
z[0,2,1]                1.000000
z[0,0,2]                2.000000
z[1,2,0]                1.000000
z[1,4,2]                2.000000
z[1,0,3]                4.000000
z[1,3,4]                3.000000
z[3,0,1]                1.000000
All other variables in the range 1-240 are zero.

```

Figura 4.17 – Exemplo de Saída do Aplicativo em ILOG CPLEX

Percebe-se pela figura que o resultado da função objetivo (*Objective*) atingido foi um custo total de \$45,64. Esse valor, como esperado, é exatamente o mesmo que aquele resultante da otimização através da planilha eletrônica, como indicado na seção 4.1.4.

As variáveis de decisão obtidas estão indicadas abaixo de *Variable Name*, possuem o valor indicado ao lado de cada uma (*Solution Value*). Todas as outras variáveis são nulas. As variáveis podem ser conferidas com aquelas presentes. Os valores são os mesmos indicados nas figuras que indicam as variáveis de decisão na seção 4.1.2. A indexação dos pedidos no *ILOG CPLEX* inicia em 0, ao invés de, como no *Solver*, começar em 1. Assim, comparando a Figura 4.17 com os resultados da Figura 4.6, é possível verificar que as respostas são iguais, sendo igualmente utilizados apenas três veículos: um veículo atende os pedidos 0, 4 e 5 (no *Solver* são os pedidos 1, 5 e 6), outro veículo atende os pedidos 3, 6, 7, 8 e 9 (4, 7, 8, 9 e 10 no *Solver*) e, por fim, o último veículo atende os pedidos 1 e 2 (2 e 3 no *Solver*).

O tempo total de processamento foi cerca de seis segundos. A planilha, como visto, levou oito horas através do aplicativo de otimização *Solver*. Isso se deve ao fato do *ILOG CPLEX* possuir bibliotecas mais avançadas e específicas para a resolução de problemas de programação linear inteira mista, enquanto o *Solver* utilizado nesta pesquisa é a versão “*standard*” disponibilizada gratuitamente com o *Microsoft Excel* e, portanto, sem os recursos.

4.3. Outros problemas resolvidos através do ILOG CPLEX

Para permitir a comparação de resultado e desempenho do método heurístico de solução proposto mais adiante neste trabalho no Capítulo 5, foram propostas outras instâncias de problemas de teste para resolução através do *ILOG CPLEX*.

Foi possível perceber, no entanto, que ao aumentar o número de pedidos, de cliente e de caminhos, o tempo de processamento aumentou dramaticamente, o que comprova o problema ser de complexidade exponencial, conforme pode ser observado na Tabela 4.2.

O nível de restrição do problema pode ser visualizado através da matriz $R_{i,v}$ (vide seção 3.2.2), que indica se determinado veículo pode atender determinado cliente ou não. O percentual de restrição pode ser dado pela relação entre as variáveis restritas e o total de

variáveis. As variáveis restritas são aquelas quais $R_{i,v} = 0$ e o total de variáveis corresponde ao produto $I \times V$.

Entretanto, como seria de se esperar, o tamanho do problema, no entanto, dados pelo número de pedidos, clientes e veículos, exerce maior influência no tempo de processamento. Algumas das instâncias geradas e sua tentativa de resolução são listadas na Tabela 4.2 a seguir. A coluna “Tempo” indica o tempo de processamento até o término da otimização ou até ser interrompido. O *Gap* é a diferença entre a melhor solução viável inteira encontrada e a solução relaxada de referência (*lower-bound*). Um *Gap* final de valor 0% indica que a solução ótima foi encontrada. Todas as instâncias testadas possuem ao menos uma solução viável, obtida manualmente ou através da heurística proposta no capítulo seguinte.

Tabela 4.2 – Instâncias e Resultados do Processamento no ILOG CPLEX

Veículos	Pedidos	Clientes + Base	Variáveis de Decisão	Restrição	Tempo	Gap
4	10	5	200	0%	6 seg	0%
10	13	8	1.250	28,6%	153 seg	0%
16	18	13	5.280	57,3%	10.978 seg	11%
16	18	13	5.280	20,8%	3.062 seg	n/a*
31	36	27	44.640	62,2	2.718 seg	n/a*

* Após o tempo de processamento indicado, nenhuma solução viável ainda havia sido encontrada e o processamento foi interrompido.

Note-se que para os dois últimos problemas não foi possível sequer encontrar uma solução viável após um tempo de processamento bastante elevado.

O algoritmo criado em *ILOG CPLEX* seria utilizado para realizar os testes de resultado e desempenho do método heurístico, porém foi demonstrada a inviabilidade do método exato de solução para o problema aqui proposto. O tempo de processamento é muito grande para problemas bem inferiores aos problemas reais existentes e que serão testados para a heurística proposta adiante.

4.4. Considerações Finais

O modelo matemático para o problema de programação de entregas de derivados de petróleo proposto na seção 3.2 foi validado a partir dos problemas resolvidos. Através de sua implementação em duas ferramentas diferentes, foram testadas instâncias pequenas de forma a permitir uma análise manual pudesse concluir que a solução encontrada era ótima e que todas as restrições haviam sido respeitadas.

Os resultados não puderam, contudo, serem obtidos para instâncias com maior número de variáveis. O tempo de processamento tornou-se muito elevado e mesmo o pacote comercial mais eficiente testado não conseguiu obter solução para instâncias de tamanho compatível com problemas reais encontrados na prática.

A natureza combinatória do problema, então, impede a obtenção de soluções ótimas mesmo para instâncias de pequeno porte. Estas instâncias, caso solucionadas, poderiam ser utilizadas para avaliar a qualidade das soluções obtidas pelo método heurístico proposto e detalhado no próximo capítulo.

5. DESENVOLVIMENTO DA ESTRATÉGIA DE SOLUÇÃO HEURÍSTICA

Este capítulo descreve a estratégia de solução proposta para o problema de roteirização da distribuição de combustível por caminhões-tanque. Conforme apresentado no Capítulo 4, o problema evidencia ser de natureza combinatória e complexidade exponencial, o que impede a obtenção de solução ótima através de métodos exatos, mesmo com pacotes comerciais computacionais mais sofisticados. Assim, propõe-se uma estratégia de solução heurística que permite a solução de instâncias encontradas na prática.

Inicialmente, na seção 5.1, são apresentadas as ferramentas de desenvolvimento utilizadas no trabalho. Em seguida, serão explicados e justificados na seção 5.2 os aspectos do método escolhido. Por fim, na seção 5.3, apresenta-se uma visão da concepção geral do algoritmo desenvolvido e o detalhamento de cada uma de suas partes, procurando explicar os conceitos adotados.

5.1. Ferramentas de Desenvolvimento

Para desenvolvimento da solução heurística, foram utilizadas como ferramentas uma linguagem de programação e uma planilha eletrônica que auxiliou a concepção inicial da estratégia.

A linguagem de programação escolhida para o desenvolvimento do algoritmo é a linguagem C++. Conforme descrito pela “The C++ Resources Network” (2004), esta é uma linguagem que permite programação orientada a objeto, pode ser compilada em quase qualquer tipo de computador ou sistema operacional e tem um desempenho muito eficiente.

Para auxiliar na concepção e no desenvolvimento da estratégia de solução heurística proposta, foi proposto um problema típico de roteirização da distribuição de derivados de petróleo, contendo a distribuição de 36 pedidos em 26 clientes, distribuídos aleatoriamente em torno de uma base, que podem ser atendidos através de 55 veículos. O exemplo foi todo modelado em uma planilha eletrônica para testar e verificar a consistência dos resultados de cada etapa de desenvolvimento do algoritmo. Essa planilha encontra-se detalhada no Apêndice C.

A avaliação que se fez durante o desenvolvimento do algoritmo é que a planilha eletrônica, apesar de possuir limitação na resolução do problema, permite uma excelente visualização do andamento da solução. Uma importante utilidade que deve ser destacada é a o apoio à depuração do código, ou seja, a busca e identificação de erros de programação que ocorreram e foram corrigidos durante a fase de desenvolvimento.

5.2. Aspectos Gerais do Método Proposto

A estratégia de solução deve ser adequada a um problema que apresenta natureza combinatória e, ao mesmo tempo, é muito restrito em decorrência das restrições anteriormente apresentadas na formulação matemática.

Não existe na literatura nenhum algoritmo que possa ser diretamente aplicado ao problema descrito, conforme observado na revisão bibliográfica (vide seção 2). A complexidade combinatória do problema impõe a adoção de métodos heurísticos para a resolução de instâncias reais, uma vez que a quantidade de variáveis eleva exponencialmente o esforço computacional para a determinação da solução exata.

Inicialmente, foi considerado o desenvolvimento de uma estratégia de solução baseada em alguma meta-heurística consagrada, tal como Algoritmo Genético ou Busca Tabu. O que se percebeu, já de início, é que algoritmos baseados nessas meta-heurísticas não seriam eficientes, uma vez que requereriam um esforço computacional elevado para encontrar soluções viáveis, dado o grande número de restrições existentes.

Mais especificamente, em relação aos Algoritmos Genéticos, a própria estrutura do problema dificulta a definição de uma representação cromossômica adequada, além da dificuldade de implementar o operador de cruzamento de forma a permitir obter soluções viáveis em termos das restrições do problema.

Da mesma forma, uma estratégia eficiente baseada em Busca Tabu fica dificultada pelas restrições do problema que tornam a exploração de uma vizinhança uma tarefa complexa. Mais especificamente, os movimentos que permitem a evolução de uma solução corrente para uma solução vizinha ficam dificultados e são pouco eficientes, tendo em vista as restrições do problema que precisam ser verificadas. Em outras palavras, o esforço computacional para explorar a vizinhança de uma solução em busca

de uma solução melhor é bastante elevado e que requereria um mecanismo de busca bastante complexo.

Assim, a estratégia de solução proposta apóia-se em uma heurística rápida que permite a determinação de uma boa solução. A heurística consiste, primeiramente, em classificar os pedidos e os caminhões conforme a facilidade de alocação e a utilidade para alocação, respectivamente. Em seguida, busca-se agrupar pedidos próximos, iniciando pelos de mais difícil alocação.

O algoritmo proposto define uma ordem de alocação dos pedidos aos veículos e vai executando as alocações segundo essa ordem, sem considerar o efeito posterior de uma alocação nos demais pedidos ainda não atendidos. Assim, o algoritmo não analisa o impacto de cada alocação nas iterações seguintes, por isso ele pode ser classificado como “*greedy algorithm*” (ou “algoritmo guloso”). O encaminhamento da solução adequada fica por conta da classificação prévia que direciona a ordem de alocação dos pedidos.

Esse algoritmo foi então aprimorado para uma estratégia mais complexa inspirada na heurística originalmente proposta por Feo e Resende (1989). A estratégia é baseada numa heurística gulosa no qual é introduzido um fator probabilístico, sendo a mesma reiniciada por diversas vezes até atingir um critério de parada. Inicialmente proposto para solução de problemas *Set Covering*, a estratégia tem sido aplicada a diversos outros tipos de problemas de natureza combinatória.

Esta estratégia concebida por esses autores foi posteriormente denominada *Greedy Randomized Adaptive Search Procedure* – GRASP – e suas diversas aplicações são listadas por Feo e Resende (1995), incluindo a roteirização de veículos.

5.2.1. GRASP

A estratégia GRASP consiste, basicamente, em um método de busca de soluções através da construção de soluções por um método guloso. Nele é inserida uma variabilidade aleatória para permitir diversificar as soluções encontradas. Como o esforço computacional para geração de cada solução é bem reduzido, procede-se diversas vezes com a construção de novas soluções até que se alcance um critério de parada e obtenha-se a solução mais próxima possível do ótimo.

Feo e Resende (1995) apresentam o conceito da estratégia GRASP, discutem a metodologia dessa técnica, demonstram como desenvolver uma solução desse tipo e apresentam uma revisão da literatura de implementações de GRASP.

Nesse trabalho, os autores apresentam o GRASP como uma heurística dividida em duas fases: (A) uma construção de uma solução rápida com fator aleatório e (B) uma busca local. Essas fases estão dispostas conforme fluxograma apresentado na Figura 5.1 a seguir.

Figura 5.1 – Fluxograma genérico de método GRASP

A fase de construção da solução (A) baseia-se em uma heurística rápida com as seguintes características:

- *Greedy*: Gulosa. É uma heurística míope, isto é, a cada iteração insere um novo elemento na solução baseado apenas no benefício isolado de selecionar cada elemento;

- *Randomized*: Aleatória. A seleção de cada elemento é feita com um sorteio entre elementos candidatos, composta por um conjunto de melhores elementos com base no benefício medido;
- *Adaptative*: Adaptativa. Pois o benefício associado a cada elemento é atualizado a cada inserção de um elemento na solução.

Após cada fase de construção determinar uma solução completa, aplica-se uma busca local (B), pois conforme explicado por Feo e Resende (1995), as soluções geradas por uma construção GRASP não são localmente ótimas.

Um exemplo da utilização da estratégia GRASP pode ser encontrado no trabalho de Delorme, Gandibleux e Rodriguez (2004), em um problema do tipo *Set Packing*.

De forma geral, a estratégia GRASP é interessante para o problema de roteirização de entregas de derivados de petróleo aqui proposto por duas características:

- A estratégia GRASP permite buscar soluções ótimas através da inserção de variabilidade nas soluções geradas através de construção heurística gulosa, respeitando a estrutura de construção de soluções de cada problema.
- O problema abordado é muito restrito, não sendo eficazes estratégias de troca de valores entre algumas poucas variáveis (como ocorre, por exemplo, em outras estratégias de melhoria como *2-opt* ou *3-opt*).

Assim, para um problema com elevado nível de restrição e com características muito particulares como o aqui proposto, parece ser indicada uma estratégia com características semelhantes às do GRASP.

Em resumo, a partir da análise da estratégia GRASP foram identificadas duas importantes características desejadas ao método proposto: heurística rápida e introdução de um fator aleatório. Essas características foram consideradas no desenvolvimento de uma heurística que permitisse a solução do problema de alocação.

5.2.2. Comparação entre a Estratégia Adotada e o GRASP

A estratégia GRASP possui um módulo de busca local bem definido. Porém, para o problema aqui contemplado, devido ao alto número de restrições, um módulo deste tipo geraria um esforço computacional muito elevado em face de um benefício relativamente restrito.

Assim, conforme será detalhado adiante, não foi criada uma rotina para a busca local para gerar uma busca na vizinhança de uma determinada solução. Foi preferido, em seu lugar, aplicar o mesmo método de construção, aplicado a sub-conjuntos dos elementos anteriormente alocados, permitindo que a busca se limite a uma vizinhança de uma solução. Essa justificativa é a mesma que Feo e Resende utilizam na concepção da fase de busca local do GRASP.

O fator aleatório, na estratégia GRASP, está contido na fase de construção (alocação). Este fator, no algoritmo aqui proposto, é introduzido na fase de geração de novas condições, na qual ocorre a definição dos elementos que permanecem na solução, ao redor dos quais será pesquisada a vizinhança.

Em resumo, foram adotadas, então, duas alterações em relação ao método GRASP:

- Não utilização de sorteio entre elementos candidatos na fase de construção, realizando-se a iteração de construção da solução através da inserção do elemento de melhor benefício.
- Supressão de uma heurística específica de busca local, utilizando-se o próprio algoritmo de construção para gerar soluções na vizinhança da solução anterior.

5.3. Método Heurístico de Solução Desenvolvido

A estratégia de solução adotada foi estabelecer uma heurística rápida que permitisse encontrar facilmente soluções possíveis, isto é, que respeitassem as restrições do problema.

Estabeleceu-se, então, um algoritmo que se utiliza dessa heurística rápida para iniciá-la com diferentes condições iniciais, verificar a qualidade da solução gerada, controlar um critério de parada e reiniciá-la com novas condições.

O algoritmo, então, pode ser dividido em:

- (a) Leitura dos dados iniciais;
- (b) Definição de solução através de heurística rápida de alocação;
- (c) Cálculo da função objetivo;
- (d) Comparação com a melhor solução;
- (e) Verificação do critério de parada;

- (f) Definição de novas condições iniciais para busca de nova solução.

O fluxograma do processo está indicado na Figura 5.2.

Figura 5.2 – Fluxograma geral do algoritmo proposto

Cada um desses passos será detalhado adiante, porém a questão que se faz imediata na compreensão da estratégia escolhida é como foi tratada a relação entre a heurística de alocação (b) e a geração de novas condições para reinício (f).

O processo de alocação tem como objetivo principal definir quais pedidos serão atendidos por quais caminhões. Como será demonstrado na seção 5.3.1, cada passo dessa alocação depende da alocação estabelecida até aquele momento.

Percebeu-se, então que uma possível forma de gerar novas soluções sem descartar parte da qualidade de uma solução inicial é realizar um sorteio de alguns veículos de uma solução para terem os respectivos pedidos desalocados. Isso permite uma busca em uma vizinhança da solução encontrada anteriormente.

Com alguns pedidos mantidos e outros removidos aleatoriamente, uma nova alocação parcial tende a gerar uma solução diferente da anterior. Portanto, a geração de novas condições configurou-se na desalocação de uma parte dos pedidos da solução anterior.

Deve-se destacar que a heurística proposta tem por finalidade principal a resolução do problema operacional diário de programação das entregas e não o problema tático de dimensionamento da frota. Assim, pressupõe-se que a frota própria é utilizada prioritariamente, sendo os veículos de terceiros alocados apenas à medida da necessidade em decorrência da falta de veículos próprios para realizar as entregas.

5.3.1. Heurística Rápida de Alocação

A heurística rápida de alocação integra a estratégia adotada para criação do algoritmo que resolva o problema proposto. Dado o grande número de restrições envolvidas, uma alocação rápida bem estruturada permite gerar boas soluções, mesmo que sub-ótimas.

5.3.1.1. Visão Geral

A heurística rápida de alocação foi formulada com objetivos bem definidos:

- Encontrar rapidamente uma solução possível;
- Gerar soluções adequadas ao objetivo da solução do problema.

Para a geração de soluções, a heurística de alocação procura, da mesma forma que a função objetivo do problema, minimizar as rotas a serem percorridas pelos veículos. Ou seja, a soma das distâncias totais percorridas deve ser a menor possível.

A heurística para ser rápida, porém, admite-se que a mesma não deve ficar avaliando diversas combinações possíveis. Ela deve se limitar, a cada iteração, no benefício isolado que a adição de cada um dos elementos promove na solução em construção.

A forma “inteligente” de fazer isso é estabelecer regras claras para que a alocação ocorra. Para que isso fosse possível, buscou-se definir características importantes que deveriam ser consideradas. Entre elas:

- Pedidos agrupados em um mesmo veículo devem estar mais próximos possível;
- Pedidos para um mesmo cliente preferencialmente estão agrupados.

- Melhor alocar antes pedidos os mais difíceis de serem alocados, ou seja, mais restritos;
- Um pedido muito difícil só consegue ser combinado com outros pedidos em caminhões mais úteis, ou seja, de maior serventia, uma medida de quantos pedidos ainda podem ser alocados a um caminhão, conforme explicado mais adiante.

Em outras palavras, a cada passo o algoritmo, apesar de míope, deve escolher pedidos e caminhões de forma a gerar uma boa solução. Para poder realizar isso sem o esforço computacional de comparar diversas combinações, utilizaram-se parâmetros que são atualizados a cada iteração.

O fluxograma geral da heurística rápida de alocação pode ser visto na Figura 5.3 a seguir:

Figura 5.3 –Visão Geral da Heurística Rápida de Alocação

De forma geral, a sequência de procedimentos realizados pela heurística de alocação é a seguinte:

- (1) Verifica se existem pedidos em um mesmo cliente:
 - Caso sim: (2) Busca um pedido no maior grupo de pedidos a um cliente;
 - Caso não: (3) Busca pedido sozinho mais difícil de alocar.
- (4) Busca caminhão próprio de maior serventia:
 - Caso não exista: (5) Busca caminhão terceirizado de maior serventia.
- (6) Aloca pedido no caminhão;
- (7) Atualiza restrições e parâmetros;
- (8) Verifica se cabem mais pedidos no caminhão:
 - Caso sim:
 - (9) Busca pedido mais próximo e aloca no caminhão;
 - (10) Atualiza restrições e parâmetros;
 - Continua até não caber outro pedido.
 - Caso não: (11) Verifica se há mais pedidos a alocar:
 - Caso sim: Reinicia;
 - Caso não: Encerra.

5.3.1.2. Restrições

As restrições às quais estão sujeitas a alocações dos pedidos nos veículos foram separadas em duas categorias: restrições estáticas e restrições dinâmicas.

Restrições estáticas são aquelas que são válidas a qualquer momento, independentemente da alocação de outros pedidos. Um exemplo de restrição estática é a limitação quanto ao padrão de pintura do veículo: independente de outros pedidos estarem alocados ao veículo, um determinado veículo pode ou não atender o cliente daquele pedido.

Restrições dinâmicas são aquelas que, ao contrário, dependem da alocação de outros pedidos para verificar se um pedido pode ou não ser atendido pelo veículo. Um exemplo é a soma do volume dos pedidos em relação à capacidade máxima do caminhão-tanque.

O primeiro passo da heurística de alocação é criar a primeira matriz de restrições estáticas. Essa matriz é uma matriz binária de dimensão $(n \times m)$, onde n é o número de caminhões e m o número de pedidos. As restrições estáticas são compostas pelas seguintes comparações:

- Padrão de pintura do veículo x permissão do cliente do pedido quanto o padrão;
- Tamanho do caminhão x restrição de tamanho pelo cliente;
- Se o pedido é maior que 15.000 litros: Pedido x capacidade do caminhão.

A seguir, apresenta-se a função responsável pela verificação das restrições estáticas:

```

1  Função Compara(ID_Caminhao, ID_Cliente_do_Pedido ){
2
3      permite = verdadeiro;
4      se(Cliente_do_Pedido não permitir Padrão do Caminhão)
5      ou (Cliente_do_Pedido não permitir Tipo do Caminhão)
6      ou ((Volume do pedido > 15) e (Tipo do Caminhão == 1)){
7          permite = falso;
8      };
9      retorna permite;
10 };

```

Figura 5.4 – Função de Verificação das Restrições Estáticas

Na linha 3 do código é inicializada a variável como verdadeira, isto é, indicando que a alocação do pedido no caminhão poderia ser feita. Caso o padrão do caminhão, seu tipo ou o volume do pedido seja maior que o volume máximo comportado pelo caminhão (linhas 4, 5 e 6), é atribuído falso à variável.

Conforme a alocação é realizada, são verificadas as restrições dinâmicas que buscam garantir que não sejam alocados mais pedidos que o possível no veículo.

Como pedidos de 5.000 litros ocupam necessariamente um compartimento desse tamanho, existe um limite de pedidos desse tamanho possíveis em cada caminhão, mesmo que a capacidade total esteja respeitada. Isto evita que o algoritmo aloque dois pedidos de 5.000 litros em um compartimento de 10.000 litros. Assim, as duas comparações que norteiam as restrições dinâmicas são:

- Soma dos pedidos alocados x capacidade total do veículo;
- Pedidos de 5.000 litros alocados x número de compartimentos de 5.000 litros.

5.3.1.3. Parâmetros de Alocação

Além das restrições, para que o algoritmo tenha o desempenho adequado, foram elaborados alguns parâmetros que permitem nortear a ordem de alocação. A alocação apesar de míope, busca ser orientada por parâmetros que indicam uma maior chance de realizar uma boa alocação a cada iteração. Esses parâmetros, portanto, são atualizados a cada nova alocação que a heurística realiza. Os três principais parâmetros de alocação definidos para a heurística são:

- Pedidos no mesmo cliente;
- Serventia do veículo;
- Facilidade de alocação do pedido.

O primeiro critério de alocação é a quantidade de pedidos a serem alocados que coincidem a um mesmo cliente. Esse valor indica que pedidos para um mesmo local tendem a serem alocados em um mesmo caminhão, sendo preferível que sejam alocados antes. Para isso, então, foi criado um vetor que, para cada pedido, indica quantos pedidos existe para aquele cliente. A seguir um trecho de código indica essa atribuição:

```

1  para (Cada pedido j){
2      soma = 0;
3      para (Cada pedido k){
4          se(pedidosHoje[k].IdCliente == pedidosHoje[j].IdCliente)
5              e {facilidadePedidos[k] != 0 }
6              e {facilidadePedidos[j] != 0} { soma++; };
7      };
8      pedidosMesmoCliente[j] = soma;
9  };

```

Figura 5.5 – Parâmetro de Pedidos para um Mesmo Cliente

O código acima indica que, para cada pedido (linha 1), serão verificado quantos pedidos existem para aquele cliente (linha 4), excluindo os pedidos já alocados (linhas 5 e 6).

Os conceitos de serventia de um veículo e facilidade de alocação de um pedido são semelhantes e se baseiam na matriz binária de restrições. Essa matriz binária de restrições indica para cada veículo i e pedido j , se é possível realizar a alocação de j em

i. Ela é atualizada a cada passo do algoritmo, pois é responsável por nortear cada passo da alocação.

A serventia de um caminhão indica durante o processo de alocação quantos pedidos ainda podem ser alocados a ele, isto é, a serventia de um veículo *i* é a contagem de quantos entre os pedidos *j* podem ser atendidos por ele.

A facilidade de um pedido ser alocado indica durante o processo de alocação, por sua vez, quantos veículos podem atendê-lo, isto é, a facilidade de um pedido *j* é a contagem de quantos entre os veículos *i* podem atendê-lo naquele momento.

Na matriz binária de restrições, um valor “1” indica que é possível a alocação do pedido *j* no caminhão *i*, enquanto um valor “0” indica o contrário. A Tabela 5.1 ilustra o conceito do cálculo dos parâmetros de serventia e facilidade.

Tabela 5.1 – Exemplo de Cálculo de Serventia e Facilidade

Ped. \ Veic.	0	1	2	3	4	5	6	7	8	9	10	Facilidade
0	1	1	0	1	1	0	1	0	0	1	1	7
1	1	1	1	1	1	1	1	1	1	1	1	11
2	1	1	1	1	1	1	1	1	1	1	1	11
3	1	1	1	1	1	1	1	1	1	1	1	11
4	1	1	1	1	1	1	1	1	1	1	1	11
5	1	1	1	1	1	1	1	1	1	1	1	11
6	1	1	1	1	1	1	1	1	1	1	1	11
7	1	1	0	1	1	0	1	0	0	1	1	7
8	0	0	1	0	0	1	0	1	1	0	0	4
9	1	1	0	0	0	0	0	0	0	0	0	2
10	1	1	1	0	0	0	0	0	0	0	0	3
Serventia	10	10	8	8	8	7	8	7	7	8	8	

Por exemplo, o pedido 7 pode ser alocado em sete diferentes veículos (0, 1, 3, 4, 6, 9 e 10), tendo como facilidade o valor 7. O pedido 8 pode ser alocado em quatro veículos (2, 5, 7 e 8), definindo sua facilidade como 4. Analogamente, o veículo 1 pode atender dez pedidos (0, 1, 2, 3, 4, 5, 6, 7, 9 e 10), tendo serventia 10, enquanto o veículo 2 pode atender a oito pedidos (1, 2, 3, 4, 5, 6, 8 e 10), tendo como serventia o valor 8.

Assim, para cada veículo é definida a serventia, ou seja, o número de pedidos que o mesmo pode alocar. E, para cada pedido, é definida a facilidade, ou seja, o número de veículos que podem lhe alocar.

O trecho de código a seguir na Figura 5.6 mostra o cálculo inicial de serventia de veículos e facilidade de alocação de pedidos:

```

1  para (Cada caminhão i){
2      soma = 0;
3      para (Cada pedido j){
4          soma = soma + matrizRestricoes[i][j];
5      };
6      serventiaCaminhoes[i] = soma;
7  };
8
9  para (Cada pedido j){
10     soma = 0;
11     para (Cada caminhão i){
12         soma = soma + matrizRestricoes[i][j];
13     };
14     facilidadePedidos[j] = soma;
15 };

```

Figura 5.6 – Parâmetros de Serventia do Veículo e Facilidade de Alocação do Pedido

Nas linhas 1 a 7 a matriz de restrições é somada para cada caminhão-tanque, gerando os valores de serventia. Nas linhas 8 a 15 a matriz é somada para cada pedido, gerando os valores de facilidade.

Um valor nulo no vetor facilidade indica que aquele pedido já foi alocado. Um valor nulo no vetor serventia indica que não cabem mais pedidos naquele caminhão.

5.3.1.4. Seleção de Pedido para Alocação

A alocação sempre é iniciada pela escolha de um pedido para ser alocado. O critério de escolha é o seguinte:

- Caso haja mais de um pedido de um mesmo cliente, procura-se o cliente com maior número de pedidos a serem entregue;
- Caso não haja clientes com mais de um pedido, procura-se o pedido mais difícil de alocar (menor facilidade).

No primeiro caso, basta identificar o índice do maior valor do vetor de contagem de pedidos para o mesmo cliente. No segundo caso, o índice necessário é o do menor valor

não-nulo (conforme código a seguir). Uma vez determinado um pedido para alocação, segue-se para a definição do veículo, conforme ilustrado no código da Figura 5.7.

```

1  Função ProcuraPedido(matrizValores[]){
2      i = 0;
3      menorIndice = 0;
4
5      para (Cada pedido i com valor nulo) {menorIndice = i};
6
7      se (Todos pedidos tem valores nulos) {Alerta};
8      senão {
9          para (Cada pedido i)      {
10             se (Valor do Pedido i < Valor do menorIndice)
11             e (Valor do Pedido i não é nulo) {menorIndice = i};
12             i++;
13         };
14     };
15     retorna menorIndice;
16 };

```

Figura 5.7 – Função de Busca para Determinação de Pedido a Alocar

Para buscar o menor valor não-nulo, o primeiro passo é encontrar um primeiro valor não-nulo para iniciar a comparação com os outros valores (linhas 2 a 5). Caso todos sejam nulos, é necessário a função retornar um valor de controle para alertar o código principal do erro (linha 7). Se ele encontrar um valor não-nulo, deve percorrer todo o restante buscando o índice do menor valor (linhas 8 a 14).

5.3.1.5. Busca de Veículo

O caminhão em que se aloca o pedido é, entre os caminhões possíveis de atender o pedido, o de maior serventia. Essa decisão é justificada, pois se pretende maximizar as opções de combinação de pedidos.

A busca é feita inicialmente apenas entre os veículos próprios. Caso não haja nenhum veículo possível, a busca é realizada entre os veículos terceirizados.

Uma vez definido o veículo, aloca-se o pedido nele, marcando a matriz alocação com essa informação e atualizando as restrições dinâmicas e os parâmetros de alocação.

5.3.1.6. Completando o Caminhão

Evidentemente que a solução final, via de regra, não terá todos os veículos ocupados em sua máxima capacidade.

Porém, a formulação da alocação foi definida para buscar o máximo aproveitamento possível dos caminhões. Como o principal objetivo é otimizar a distribuição, reduzindo as distâncias percorridas, enquanto houverem pedidos possíveis de serem alocados ao caminhão, isso será feito.

O critério para escolha do novo pedido, caso haja mais de um pedido possível, é a proximidade ao último pedido alocado. A cada novo pedido alocado ao caminhão, são atualizados as restrições e os parâmetros de alocação.

O processo segue sucessivamente até que não seja mais possível alocar nenhum pedido ao caminhão. Verifica-se, então, se existem outros pedidos a serem alocados. Em caso afirmativo, procede-se novamente com a escolha inicial de pedidos conforme descrito na seção 5.3.1.4.

Se todos os pedidos já estiverem alocados, encerra-se a alocação e passa-se à avaliação de seu resultado.

Se ao final da alocação algum pedido ainda não estiver alocado, mas não existir veículo possível de atendê-lo, a solução é descartada. No lugar dela é re-inserida a atual melhor solução.

5.3.2. Verificação da função objetivo

A função objetivo considerada para este problema é o custo total, considerado em duas parcelas: custos fixos e custos variáveis.

Custos fixos e variáveis são atribuídos individualmente a cada veículo, pois a frota é heterogênea.

De forma geral, os custos atribuídos a veículos terceirizados são considerados mais elevados, pois devem cobrir adicionalmente no mínimo o lucro do operador do veículo.

A parcela de custo fixo é um valor considerado para um determinado veículo caso este faça parte da solução, ou seja, independente do número de pedidos alocados a ele.

O custo variável é um valor de custo por unidade de distância percorrida, ou seja, o custo variável é multiplicado pela distância percorrida na rota a qual o veículo está alocado.

O custo total por veículo, então, é nulo caso ele não tenha pedidos a ele alocados. Se houver, é o seu custo fixo somado a seu custo variável multiplicado pela distância programada a percorrer.

Após o cálculo do valor da função objetivo, é verificada se a solução encontrada é a melhor solução até aquele momento. Em caso afirmativo, ela é copiada para uma matriz e guardada tanto para fornecer a resposta final, quanto para poder ser re-inserida no processo, conforme será explicado adiante.

5.3.3. Definições de novas condições iniciais para busca de nova solução

Além da heurística rápida de alocação, o segundo ponto principal do algoritmo desenvolvido está relacionado aos critérios de geração de novas soluções híbridas a partir de uma solução corrente.

De forma simplificada, o processo adotado pode ser explicado como a desalocação aleatória de alguns pedidos para obter novos resultados com a re-alocação dos mesmos.

Esse procedimento, no entanto, deve ser compreendido mais profundamente para que seja possível entender os fatores que podem levá-lo ou não a ser uma boa solução.

A estratégia, inspirada em conceitos do GRASP, torna-se interessante neste caso pois a eficiência da heurística de alocação permite que ela seja repetida por dezenas de milhares de vezes em menos de um minuto em um computador pessoal, conforme será apresentado mais adiante na seção 6.5 deste trabalho.

O desafio que se apresentou nesse momento foi equilibrar a variabilidade de soluções com a convergência em busca da solução ótima. As etapas que compõem este processo são:

- (1) Verifica repetição se atende critério para re-inserção da melhor solução;
 - Caso sim: (2) Re-insere melhor solução;
- (3) Sorteia uma quantidade pré-determinada de caminhões para desalocar;
- (4) Desaloca todos os pedidos dos caminhões sorteados;
- (5) Atualiza restrições e parâmetros.

O fluxograma dessas etapas está ilustrado na Figura 5.8.

Figura 5.8 – Visão Geral da Geração de Novas Condições Iniciais

5.3.3.1. Re-inserção da melhor solução

A re-inserção da melhor solução permite que após alguns ciclos de desalocação e re-alocação, seja possível retornar a um ponto melhor pelo qual já se passou e, a partir dele, explorar outras soluções próximas.

Quanto mais freqüentemente a melhor solução corrente for repetida, menor será a variabilidade em relação a solução e maior a chance do problema de prender a soluções de mínimo local.

Por outro lado, uma dispersão muito alta na busca, permite buscar patamares de solução menores, mas com dificuldade de encontrar os mínimos locais desses novos patamares. A convergência do processo, obviamente, é mais lenta.

O critério escolhido foi definir um intervalo de ciclos entre cada re-inserção da melhor solução no processo. Esse valor foi testado e definido como uma re-inserção a cada 1.000 ciclos.

5.3.3.2. Desalocação

O objetivo da desalocação é promover os ciclos de desalocação e re-alocação que permitam obter novas soluções. A desalocação compõe-se de duas etapas: o sorteio dos caminhões e a desalocação propriamente dita.

Para o sorteio de caminhões, define-se anteriormente uma porcentagem dos caminhões da solução atual a serem desalocados. Essa porcentagem, semelhante ao que foi exposto na seção anterior, também deve ser avaliada e testada.

O processo de desalocação é muito similar ao processo inverso da heurística de alocação. Conforme vão sendo sorteados os pedidos a desalocar, processa-se as restrições estáticas. A reprogramação das restrições dinâmicas de toda a matriz, no entanto, só pode ser feita após todos os veículos terem sido sorteados e desalocados.

No código de exemplo da Figura 5.9 são desalocados $\frac{1}{4}$ dos caminhões alocados.

```

1  camUsados = 0;
2  para(Cada caminhão k){
3      se(Caminhão k foi utilizado) camUsados++;
4  };
5
6  para(Repetir até 1/4 dos caminhões usados){
7      caminhaoDesalocando = Caminhão rand();
8      enquanto(caminhão sorteado não foi usado){
9          caminhaoDesalocando = Caminhão rand();
10     };
11
12     volumeOcupado[caminhaoDesalocando] = 0;
13     compDe5Ocupados[caminhaoDesalocando] = 0;
14
15     para(Cada pedido j){
16         Atualiza matrizRestricoes[][];
17         Atualiza matrizAlocacao[][];
18     };
19
20 };
21
22 para(Cada pedido j){
23     para(Cada caminhão i){
24         se(Volume de j for maior que o espaço restante em i)
25         ou(Volume de j = 5 e não há compartimentos de 5 disponíveis em i)
26             matrizRestricoes[i][j] = falso;
27     };
28 };

```

Figura 5.9 – Sorteio e Desalocação de Pedidos

O primeiro passo para a desalocação é a contagem dos caminhões utilizados, realizado nas linhas 1 a 4. Depois, devem ser sorteados $\frac{1}{4}$ dos caminhões utilizados, apenas entre

eles (linhas 6 a 12). Para cada caminhão utilizado sorteado, procede-se com a desalocação (linhas 14 a 21). Após todos os caminhões sorteados terem sido desalocados, é feita a atualização das restrições conforme as alocações que permaneceram na solução. Por fim, ainda é realizado o re-processamento dos parâmetros de alocação. Após a atualização de restrições e parâmetros, é reiniciado o processo de alocação.

Como o problema aqui proposto é muito restrito, um número muito pequeno de desalocações realizadas tende a gerar posteriormente pouca alternativa para as realocações e, conseqüentemente, não obterá grande variabilidade de soluções. Caso a quantidade de desalocações seja muito elevada, a solução também tende a ter pouca variabilidade, mas por outra razão: quanto menos alocações feitas, mais próxima a alocação tende a ser da mesma alocação inicial (gerada a partir de nenhum pedido alocado).

Como a heurística de alocação é míope, para um mesmo conjunto de pedidos e veículos, a alocação tende a ser a mesma. A variabilidade é gerada, pois os parâmetros de alocação e restrições mudam de acordo com a alocação.

5.3.4. Verificação do critério de parada

Para verificação do comportamento da evolução do algoritmo em relação a solução do problema, o critério de parada foi expandido um pouco além do que parece ser necessário, como forma de garantir que existe um mínimo de estabilidade na solução gerada, após muitas iterações sem melhoria da mesma.

O critério de parada é um número fixo de repetições que foi estabelecido em 100.000 ciclos.

5.4. Considerações Finais

Após a demonstração que a natureza combinatória e complexidade exponencial do problema de distribuição de derivados de petróleo impedir a obtenção de solução ótima através de métodos exatos, foi proposta uma estratégia de solução heurística.

A partir da análise dos conceitos da estratégia GRASP, foram identificadas as características importantes que foram consideradas no desenvolvimento de uma heurística que permitisse a solução heurística do problema.

Foi, então, elaborado um algoritmo de solução heurística que contempla uma heurística rápida de alocação iniciada múltiplas vezes com diferentes condições iniciais através de ciclos de desalocações sorteadas aleatoriamente.

No próximo capítulo são apresentados experimentos computacionais que permitem avaliar e analisar o método aqui proposto.

6. EXPERIMENTOS COMPUTACIONAIS

Este capítulo tem por objetivo descrever os experimentos computacionais realizados a fim de avaliar a estratégia de solução proposta, assim como medir os impactos de utilização de diferentes parâmetros na eficiência e eficácia da mesma.

Não foi possível aplicar a heurística proposta a uma instância real de problema de distribuição de derivados de petróleo, em decorrência da dificuldade de obter dados reais adequados. Assim, não foi possível para comparar uma solução obtida pela alocação manual praticada nas distribuidoras, com uma resultante do algoritmo computacional proposto, uma medida do potencial de economia proporcionada pela heurística proposta.

A heurística não é também comparada com o problema de dimensões reduzidas proposto na seção 4.1, uma vez que o mesmo é muito restrito e impede a troca de pedidos, base da heurística proposta.

Conseqüentemente, a fim de testar a heurística proposta, foram gerados, então, algumas instâncias de problemas de teste para realização de experimentos de forma estruturada, permitindo realizar a análise e avaliação do método proposto.

Na seção 6.1 são detalhadas as diferentes classes de problemas gerados para os testes do método heurístico e os critérios para geração das instâncias com bases nessas classes. Em seguida, na seção 6.2, será descrita como essas classes ainda são variadas em função do grau de restrição das instâncias. Para exemplificar a forma de resposta que a heurística fornece, na seção 6.3 é apresentado um exemplo de relatório do algoritmo e um exemplo gráfico de solução gerada. A variação dos parâmetros da heurística utilizada para realizar os testes é descrita na seção 6.4.

Os resultados dos testes são apresentados na seção 6.5, e a análise mais detalhada destes resultados encontra-se na seção 6.6.

6.1. Geração de Instâncias de Problemas para Teste

Conforme discutido anteriormente no Capítulo 4, a evidência de complexidade exponencial do problema tratado impede a obtenção da solução ótima através de

algoritmos exatos, mesmo para problemas de menor porte, o que impede a sua utilização para a avaliação da heurística proposta.

Por outro lado, não existe, também, uma base de dados de problemas teste difundida para o problema específico tratado aqui.

Não sendo possível comparar o desempenho da heurística formulada com outro método para o mesmo problema ou com problemas de referência, propõe-se aqui a realização de outros testes, com diferentes instâncias, a fim de aferir a aplicabilidade do método e que tornem possíveis futuras comparações com outras estratégias de solução que venham a ser desenvolvidas para problemas de mesma natureza.

As instâncias elaboradas devem ter tamanhos compatíveis com a estimativa que se faz a respeito da grandeza de problemas reais. Conforme dados da ANP (2001), a maior rede de distribuição no Estado de São Paulo possuía em 2001 1.546 postos de combustíveis. Esses postos não solicitam pedidos todos os dias, nem são atendidos por uma única base. Propõe-se então a verificação do desempenho do algoritmo com problemas de ordem de grandeza entre 50 e 100 pedidos, atendendo entre 35 e 75 clientes em um único dia, o que parece razoável para um problema real. A quantidade de veículos disponível foi considerada metade em caminhões próprios e outra igual quantidade em veículos terceirizados.

Assim, serão testadas três classes de problema, com as seguintes características:

- Classe A: 50 pedidos, 50 caminhões e 35 clientes;
- Classe B: 75 pedidos, 75 caminhões e 53 clientes;
- Classe C: 100 pedidos, 100 caminhões e 75 clientes.

Conforme será detalhado adiante na seção 6.2, para cada classe serão geradas três instâncias de, respectivamente, 20%, 40% e 60% de restrição, totalizando nove instâncias para teste.

6.1.1. Critérios de geração de Instâncias

Os problemas propostos correspondem a uma região com geometria retangular e distribuição uniforme e homogênea dos clientes, com a base, de onde partem os veículos, posicionada no centro dessa área.

Para a geração das instâncias, foi arbitrada uma área de 120 por 80 quilômetros e, conseqüentemente, a base de distribuição localiza-se no ponto de coordenadas (60, 40). Para cada cliente foi definida a sua localização com base em coordenadas sorteadas aleatoriamente.

A dimensão da área foi escolhida em função de uma hipótese de uma base atendendo postos de combustível em municípios vizinhos, em uma distância de influência de até 72 km, que é a máxima distância do centro a um canto da área. Essa distância representa uma base que atende uma região de municípios próximos.

Os pedidos foram gerados também de forma aleatória. Sorteou-se, para cada cliente, um número de pedidos variando entre um e três, de modo que nenhum cliente fique sem pedido. O volume do pedido foi gerado de acordo com uma distribuição amostral arbitrada conforme Tabela 6.1.

Tabela 6.1 – Distribuição Arbitrada para Volume dos Pedidos

Volume	5.000 l	10.000 l	15.000 l	20.000 l	25.000 l	30.000 l
Quantidade	10%	27%	32%	23%	7%	1%

A distribuição proposta foi arbitrada de forma a representar um padrão comercial no qual pedidos médios de um mesmo combustível são muito comuns, pedidos pequenos também são freqüentes, mas que raramente é feito um pedido muito grande de um mesmo combustível. Ressalta-se que, como explicado anteriormente, o maior caminhão é de 30.000 litros e um pedido com produtos diferentes para um mesmo cliente são representados como pedidos diferentes para esse cliente.

Por fim, os veículos foram gerados considerando 50% veículos próprios e 50% terceirizados. O dimensionamento de frota própria tende a ser abaixo do usual necessário, de forma que sempre uma pequena parcela seja feita através de terceiros, porém garantindo que em boa parte do tempo toda frota própria seja utilizada. A oferta de frota terceirizada deve atender situações de pico; assim se imagina que esta não seja muito reduzida. Considera-se, então, adequada a proporção de meio a meio entre veículos próprios e terceirizados, de forma a não haver risco de desabastecimento por falta de veículos para entrega.

Conforme visto anteriormente, os custos fixo e variável podem ser definidos veículo a veículo. Porém, para geração das instâncias, os custos fixos de cada veículo próprio foram admitidos como R\$ 100,00 (cem reais) e dos veículos terceirizados como R\$ 200,00 (duzentos reais). Os custos variáveis foram atribuídos como R\$ 1,00 (um real por quilômetro) para veículos próprios e R\$ 1,50 (um real e meio por quilômetro) para veículos terceirizados.

Na prática, às vezes o frete de veículos terceirizados pode ser configurado apenas como um custo variável, mas, quando isso ocorre, usualmente esses operadores exigem uma rota de distância mínima a ser percorrida, o que pode ser interpretado como uma forma de considerar implicitamente um valor de custo fixo a ser coberto pelo valor total do frete a ser pago. Optou-se aqui por considerar, então, esses componentes separadamente mesmo para os veículos terceirizados.

Os custos fixos estão associados a um custo mínimo diário para cobrir despesas principalmente de mão de obra. O custo variável é principalmente associado ao custo de rodagem (combustível, óleos lubrificantes e pneus). Os veículos terceirizados devem ainda cobrir o lucro do transportador. Esses valores representam ordens de grandezas próximas a valores reais, mas não advém de nenhuma pesquisa de mercado.

Como detalhado anteriormente, existem três tipos de veículo, conforme sua capacidade e compartimentação:

- Tipo 1: Truck com três compartimentos de 5.000 litros;
- Tipo 2: Carreta com seis compartimentos de 5.000 litros;
- Tipo 3: Carreta com dois compartimentos de 10.000 litros e dois de 5.000 litros.

A distribuição da frota por tipo de veículo foi determinada conforme a seguinte distribuição arbitrada:

Tabela 6.2 – Distribuição Arbitrada para Tipo de Veículo

Tipo de Veículo	Tipo 1 (<i>Truck</i>)	Tipo 2 (Carreta)	Tipo 3 (Carreta)
Volume Total	15.000 litros	30.000 litros	30.000 litros
Compartimentos	5 - 5 - 5	5 - 5 - 5 - 5 - 5 - 5	10 - 5 - 10 - 5
Quantidade	40%	30%	30%

6.2. Variação do Nível de Restrição

Um aspecto importante a ser verificado é o impacto das restrições de atendimento aos resultados e desempenho do algoritmo heurístico elaborado.

Além das restrições do problema descritas anteriormente, que impedem um determinado veículo atender determinado cliente por questões de padrão de pintura e tamanho de veículo, pode-se imaginar quaisquer outras questões, como por exemplo, a segmentação de frota para grupos de clientes. Isto é, para cada veículo determina-se se existe ou não restrição de atender determinado cliente. O conjunto de restrições dessa natureza pode ser simplificada a uma representação de matriz de restrições análoga à elaborada na formulação matemática descrita na seção 3.2.2.

O nível de restrição influenciou muito no desempenho do método exato. Para verificar isso, foi instituída uma medida de restrição na seção 4.3, que representa o percentual de atendimentos restritos, ou seja, a parcela de pedidos que apresentam restrições dos clientes quanto ao tamanho e ao tipo de pintura.

Para cada uma das instâncias foram elaboradas aleatoriamente matrizes de restrição com três graus de restrição: 20%, 40% e 60%. O grau de restrição, como dito anteriormente, é o percentual de combinações de atendimentos de cada veículo para o conjunto de clientes que está impedidos. Ou seja, nos problemas de teste gerados, cada veículo é impedido de atender, em média, respectivamente 20%, 40% e 60% dos clientes.

6.3. Exemplo de Resultado

Para ilustrar o comportamento do algoritmo e facilitar a compreensão de sua heurística, ilustra-se aqui o relatório do processamento. Um exemplo de saída do relatório do algoritmo é, então, apresentado na Figura 6.1 a seguir.

O exemplo apresentado aqui possui as seguintes características:

- 35 Clientes distribuídos aleatoriamente;
- 50 Pedidos;
- Frota composta por 50 veículos, sendo 25 próprios e 25 terceirizados;
- Nível de restrição: 60%.

```

Solucao inicial na iteracao 0: 8230.63.
Nova melhor solucao encontrada na iteracao 15: 8182.74.
Nova melhor solucao encontrada na iteracao 16: 8141.24.
Nova melhor solucao encontrada na iteracao 21: 7972.89.
Nova melhor solucao encontrada na iteracao 22: 7849.57.
Nova melhor solucao encontrada na iteracao 55: 7809.44.
Nova melhor solucao encontrada na iteracao 87: 7800.1.
Nova melhor solucao encontrada na iteracao 91: 7749.55.
Nova melhor solucao encontrada na iteracao 99: 7655.2.
Nova melhor solucao encontrada na iteracao 137: 7607.03.
Nova melhor solucao encontrada na iteracao 158: 7408.
Nova melhor solucao encontrada na iteracao 736: 7359.4.
Nova melhor solucao encontrada na iteracao 1273: 7209.45.
Nova melhor solucao encontrada na iteracao 2001: 7207.86.
Nova melhor solucao encontrada na iteracao 2002: 7089.69.
Falta Pedido. Reinserindo melhor solucao na iteracao 4509.
Falta Pedido. Reinserindo melhor solucao na iteracao 9486.
Nova melhor solucao encontrada na iteracao 13001: 7073.08.
Nova melhor solucao encontrada na iteracao 15002: 7062.56.
Falta Pedido. Reinserindo melhor solucao na iteracao 15189.
Nova melhor solucao encontrada na iteracao 16914: 6776.66.
Nova melhor solucao encontrada na iteracao 18001: 6759.49.
Falta Pedido. Reinserindo melhor solucao na iteracao 19820.
Falta Pedido. Reinserindo melhor solucao na iteracao 20625.
Nova melhor solucao encontrada na iteracao 20627: 6678.93.
Falta Pedido. Reinserindo melhor solucao na iteracao 21260.
Nova melhor solucao encontrada na iteracao 24001: 6650.03.
Falta Pedido. Reinserindo melhor solucao na iteracao 24343.
Nova melhor solucao encontrada na iteracao 25001: 6649.18.
Nova melhor solucao encontrada na iteracao 28001: 6469.87.
Falta Pedido. Reinserindo melhor solucao na iteracao 36743.
Falta Pedido. Reinserindo melhor solucao na iteracao 39821.
Falta Pedido. Reinserindo melhor solucao na iteracao 40876.
Falta Pedido. Reinserindo melhor solucao na iteracao 42747.
Nova melhor solucao encontrada na iteracao 51001: 6419.26.
Falta Pedido. Reinserindo melhor solucao na iteracao 53341.
Falta Pedido. Reinserindo melhor solucao na iteracao 61133.
Falta Pedido. Reinserindo melhor solucao na iteracao 68596.
Falta Pedido. Reinserindo melhor solucao na iteracao 73704.
Falta Pedido. Reinserindo melhor solucao na iteracao 87176.
Falta Pedido. Reinserindo melhor solucao na iteracao 88703.
Falta Pedido. Reinserindo melhor solucao na iteracao 93589.
Nova melhor solucao encontrada na iteracao 94001: 6404.56.
FIM. Melhor solucao encontrada: 6404.56

```

Figura 6.1 – Exemplo de Relatório de Comportamento do Algoritmo

O critério de parada imposto foi um número máximo de iterações fixado em 100.000. Antes de cada nova iteração foi estipulada a desalocação de $\frac{1}{4}$ da frota utilizada. A cada 1.000 iterações, a melhor solução obtida até o momento é re-inserida, ou quando é gerada uma solução incompleta (falta pedido a ser alocado).

Este é apenas um exemplo, sendo que este e outros resultados são apresentados mais claramente mais adiante na seção 6.5 deste relatório. O exemplo, porém, ilustra bem o comportamento típico do algoritmo criado. Percebe-se que no início diversas novas melhores soluções são encontradas e, quanto melhor a solução vigente, mais difícil se torna de encontrar uma solução ainda melhor.

No desenho estão sobrepostas as 29 rotas, uma para cada veículo. Elas estão diferenciadas por cor e pelos números indicados nos vértices, que representam o código do veículo (de 0 a 49).

Não é possível visualizar todas as linhas, uma vez que alguns clientes são atendidos por mais de um caminhão e as rotas se sobrepõem, não sendo possível representar todas na figura.

A ilustração, no entanto, permite visualizar a consistência das rotas traçadas e a qualidade da solução obtida.

6.4. Variação dos Parâmetros

O principal aspecto a ser avaliado na heurística proposta é a calibração com relação o parâmetro que influencia na variabilidade e na convergência do método proposto. A diferenciação das soluções produzidas em cada iteração depende do número de veículos desalocados em relação à solução anterior. O ciclo de desalocação e re-alocação é melhor explicado na seção 5.3.3.2.

Como a heurística é fixa, para um determinado conjunto de pedidos e veículos disponíveis, o resultado é sempre o mesmo. Então o número de veículos desalocado não pode nem ser muito grande, nem ser muito pequeno. Foram testados os valores de $\frac{1}{2}$ e $\frac{1}{4}$ de veículos da frota utilizada para cada instância considerada.

O critério de parada não sofreu variação, permanecendo fixado em 100.000 iterações (ciclos de desalocação e re-alocação). Para cada instância de teste foram feitas cinco rodadas para cada valor de parâmetro de taxa de desalocação ($\frac{1}{2}$ e $\frac{1}{4}$), totalizando dez rodadas, conforme detalhado na seção a seguir.

6.5. Resultados Obtidos

Como cada uma das três classes foi gerada com três diferentes graus de restrição, temos na verdade, nove diferentes instâncias. Para cada um dos níveis de restrição, no entanto, foram mantidos os mesmos conjuntos de pedidos, clientes e caminhões. Cada uma dessas nove instâncias foi solucionada utilizando dois diferentes valores para o parâmetro de desalocação de $\frac{1}{2}$ e $\frac{1}{4}$.

Como o resultado de cada rodada é influenciado por fatores aleatórios, inerentes à estratégia de solução proposta, a heurística elaborada foi rodada cinco vezes para cada instância.

Os resultados detalhados de todas as rodadas de processamento encontram-se descritos no Apêndice D.

A análise dos resultados será realizada na próxima seção deste relatório e a Tabela 6.3 apresenta o resumo dos resultados alcançados.

Para cada instância são apresentados os seguintes resultados:

- Resultado Inicial – Custo do resultado obtido no primeiro resultado da heurística, sem os ciclos de desalocação e re-alocação aleatórios.
- Melhor Solução – Custo da melhor solução em relação às dez rodadas (cinco rodadas com cada parâmetro de desalocação), que resulta da melhor solução encontrada.
- Taxa – Taxa de desalocação que indica a fração dos veículos aleatórios da solução corrente que tem seus pedidos desalocados, conforme descrito anteriormente.
- Melhor Resultados – Custo da melhor solução em relação às cinco rodadas com a taxa de desalocação indicada. Entre parênteses é indicado o desvio em relação à melhor solução.
- Resultado Médio – com a taxa de desalocação indicada. Entre parênteses é indicado o desvio em relação à melhor solução.
- Desvio Padrão – Desvio padrão (em %) entre os custos obtidos nas cinco rodadas da instância com a taxa de desalocação indicada;
- Melhoria Média – Percentual ou valor absoluto médio de melhoria do resultado inicial para os resultados no fim dos ciclos de sorteio com a taxa de desalocação indicada.
- Tempo (s) – Tempo total de processamento de cada rodada, em segundos, em um computador Pentium IV, 3GHz. Ressalta-se que a variação do tempo de processamento em um mesmo computador é imperceptível nas diferentes rodadas de uma mesma classe e mesmo grau de restrição. Isso ocorre, pois o número de iterações é fixo em 100.000, como critério de parada.

Tabela 6.3 – Resumo dos Resultados dos Testes Realizados

Instância	Resultado Inicial	Melhor Solução	Taxa	Melhor Resultado	Resultado Médio	Desvio Padrão %	Melhoria Média%	Tempo (s)
A - 20%	7.998,23	6.343,40	½	6.432,78 (+1,4%)	6.580,06 (+3,7%)	1,5%	17,70%	57
			¼	6.343,40 (+0,0%)	6.385,28 (+0,7%)	0,7%	20,20%	36
A - 40%	8.035,50	6.465,03	½	6.644,85 (+2,8%)	6.717,00 (+3,9%)	1,2%	16,40%	56
			¼	6.465,03 (+0,0%)	6.508,27 (+0,7%)	0,8%	19,00%	37
A - 60%	8.230,63	6.404,56	½	6.428,44 (+0,4%)	6.706,18 (+4,7%)	2,6%	18,50%	57
			¼	6.404,56 (+0,0%)	6.578,70 (+2,7%)	1,5%	20,10%	37
B - 20%	10.281,30	9.143,79	½	9.314,87 (+1,9%)	9.407,89 (+2,9%)	0,6%	8,50%	152
			¼	9.143,79 (+0,0%)	9.254,56 (+1,2%)	1,0%	10,00%	100
B - 40%	10.431,20	9.236,31	½	9.305,73 (+0,8%)	9.368,26 (+1,4%)	0,5%	10,20%	153
			¼	9.236,31 (+0,0%)	9.286,97 (+0,5%)	0,7%	11,00%	99
B - 60%	10.433,30	9.329,89	½	9.329,89 (+0,0%)	9.579,55 (+2,7%)	1,7%	8,20%	155
			¼	9.366,85 (+0,4%)	9.489,21 (+1,7%)	0,9%	9,00%	101
C - 20%	14.374,40	12.794,70	½	12.838,90 (+0,3%)	13.030,64 (+1,8%)	0,8%	9,30%	340
			¼	12.794,70 (+0,0%)	12.972,18 (+1,4%)	0,8%	9,80%	214
C - 40%	14.425,10	12.456,30	½	12.797,80 (+2,7%)	13.035,80 (+4,7%)	1,6%	9,60%	341
			¼	12.456,30 (+0,0%)	12.928,30 (+3,8%)	2,3%	10,40%	214
C - 60%	14.435,90	12.668,80	½	13.125,00 (+3,6%)	13.180,08 (+4,0%)	0,5%	8,70%	341
			¼	12.668,80 (+0,0%)	12.998,18 (+2,6%)	1,7%	10,00%	214

Pode-se perceber de imediato que, em relação à solução inicial, os ciclos de desalocação e re-alocação geraram melhorias entre 9% e 20%.

Os resultados obtidos pela taxa de desalocação de ¼, de forma geral, são melhores que aqueles obtidos pela taxa de desalocação de ½. É possível também perceber que o desvio padrão das rodadas com a taxa ½ foram sempre superiores aos das rodadas de taxa ¼.

As rodadas com a taxa de desalocação de ¼ foram mais eficientes em termos de tempo de processamento. Isso é fácil de ser entendido, pois o esforço computacional é menor para sortear e desalocar, pois são menos veículos desalocados que para a taxa ½. Os tempos de processamentos são bastante reduzidos, mesmo para instâncias grandes (3 a 5 minutos para instâncias de 100 pedidos).

Não houve diferença de desempenho significativo para alteração entre os níveis de restrições 20%, 40% e 60%.

6.6. Análise Detalhada dos Resultados Obtidos

- **Resultado Médio e Desvio**

O resultado médio de custo resultou de acordo com o que seria intuitivamente esperado: na média, níveis de restrição maiores tendem a diminuir a quantidade de combinações possíveis e penalizar o custo total. Ressalta-se, no entanto, que como as restrições são geradas de forma aleatória e um único conjunto de restrições foi sorteado para cada classe, pode acontecer de um sorteio, mesmo que mais restrito, seja mais favorável que o outro.

O desvio padrão entre as soluções geradas, para uma mesma taxa de desalocação, conforme apresentado na Tabela 6.3 são em média da ordem de 1,1% do valor da solução (variaram de 2,5% a 0,5% nos testes). Apesar dos resultados variarem a cada rodada, não mudam a ordem de grandeza da solução alcançada. No entanto essa variação indica que efetuar algumas rodadas e considerar o menor valor é de fato uma boa estratégia de solução. Ou seja, a heurística parece ser variável, mas suficientemente rápida para que sejam adotadas várias rodadas ao se tentar resolver os problemas propostos. A coluna “Melhor Solução” indica as melhores soluções encontradas na Tabela 6.3.

- **Tempo de Processamento**

O tempo de processamento de execução do aplicativo elaborado é expresso em segundos, em um computador pessoal com processador Pentium IV - 3 GHz.

A primeira observação é que, para uma mesma instância, o tempo de processamento não varia nunca. Pode-se perceber também que o nível de restrição não afeta de forma significativa o tempo de processamento. Para problemas mais restritos, a heurística tende a gerar soluções inviáveis mais vezes, re-inserindo mais vezes a melhor solução vigente, por consequência. Mas o esforço computacional dessa ação adicional é irrisório.

Os dois fatores que geram significativa variação no tempo de processamento são a taxa de desalocação e o tamanho do problema. A taxa de desalocação interfere no tempo de processamento, sendo que a desalocação e re-alocação de $\frac{1}{4}$ da frota alocada é mais rápido que o ciclo que envolva $\frac{1}{2}$ da mesma frota. Os ciclos de $\frac{1}{4}$ são em média 35% mais rápidos que os de $\frac{1}{2}$. Isso parece ser uma relação constante.

O tamanho do problema, como esperado, é o mais importante influenciador no tempo de processamento. A instância B possui 1,5 vezes o número de pedidos, veículos e clientes que a instância A, mas ele demora, em média, 2,7 vezes o tempo de processamento. A instância C, por sua vez, possui 2,0 vezes o número de pedidos, veículos e clientes, porém demorou em média 5,9 vezes o tempo para processar. Percebe-se que a relação é crescente não-linear.

- **Comparação com o Resultado Inicial**

O resultado obtido no primeiro resultado da heurística (“Resultado Inicial”, na Tabela 6.3), sem os ciclos de desalocação e re-alocação aleatórios, foi referenciado para permitir uma avaliação isolada do desempenho dos ciclos de desalocação e re-alocação.

A solução inicial nada mais é do que uma primeira solução factível e gerada de forma míope. Considerando que os custos fixos são entre R\$ 100,00 e R\$ 200,00 e os custos variáveis entre R\$ 1,00 e R\$ 1,50, a Tabela 6.3 indica que a economia proporcionada pelos ciclos é altamente considerável, indicando economia grande de veículos e tamanho das rotas. Porém, como infelizmente não foi possível obter a partir do método exato as soluções ótimas, fica difícil avaliar se essas economias aproximam suficientemente a solução da solução ótima.

- **Taxa de Desalocação**

A solução inicial é dada pela construção de uma solução a partir de uma frota totalmente livre, isto é, com todos os veículos desalocados. Por isso que, sendo a heurística aplicada para alocação sempre a mesma, os resultados da alocação inicial são sempre iguais.

Os ciclos de desalocação e re-alocação, então, são os responsáveis por introduzir um fator aleatório e permitir que sejam atingidas soluções melhores. Deve ser compreendido, no entanto, que quanto maior a frota desalocada a cada iteração, mais próxima da solução inicial tendem a ser as soluções geradas.

A primeira conclusão é que a taxa de desalocação $\frac{1}{4}$ obteve resultados melhores que a taxa $\frac{1}{2}$. Os resultados não apenas foram melhores, como o tempo de processamento também é menor.

Provavelmente, os melhores resultados foram obtidos porque as soluções conseguiram se desprender mais da solução inicial com a taxa de $\frac{1}{4}$ do que com $\frac{1}{2}$.

Um valor muito baixo é também limitante, uma vez que se o número de desaloções forem reduzidos demais, também existirá dificuldade da solução se afastar da solução anterior. Mais teste podem ser feitos, porém o objetivo aqui não é calibrar com exatidão o parâmetro para o problema proposto, mas sim explicar e ressaltar a importância do mesmo.

- **Rotas Geradas**

As rotas geradas pela heurística são em geral ótimas, dado o pequeno número de atendimento que em média é feito por um veículo. Mas a geração dessas rotas é pelo ponto mais próximo, o que de fato não garante que elas sejam sempre ótimas. Para até três pontos (a base e dois clientes) a rota sempre será ótima, formando um triângulo. No caso de mais de três pontos podem ocorrer rotas sub-ótimas. O limite teórico é o atendimento de seis clientes, dado que o caminhão mais compartimentado possui seis compartimentos.

A proposta que aqui se faz é no desenvolvimento de melhorias da estratégia aqui proposta, seja inserida uma sub-rotina que faça tentativas de trocas entre a ordem dos pedidos para otimizar rotas com mais de dois clientes e gerar de forma garantida o melhor roteiro de atendimento para cada veículo automaticamente.

6.7. Considerações Finais

Os experimentos computacionais a partir dos testes gerados permitiram perceber a adequação da solução heurística proposta ao problema de distribuição de derivados de petróleo, além de realizar algumas análises e conclusões:

- A partir da solução inicial, os ciclos de desalocação e re-alocação geraram melhorias entre 9% e 20%.
- Taxa de desalocação de $\frac{1}{4}$ geralmente atinge melhores resultados que a taxa de desalocação de $\frac{1}{2}$;
- Taxa de desalocação de $\frac{1}{4}$ é mais eficiente em termos de tempo de processamento;
- Os tempos de processamentos são bastante reduzidos, mesmo para instâncias grandes (100 pedidos);

- Tendência de comportamento não exponencial do tempo de processamento em relação ao tamanho do problema:

Apesar de, como discutido anteriormente, não ter sido possível encontrar as soluções ótimas exatas para problemas de tamanho real, os testes realizados permitiram avaliar a solução proposta.

7. CONCLUSÕES E RECOMENDAÇÕES

Neste trabalho tratou-se do problema de roteirização de entregas de derivados de petróleo em postos de abastecimento a partir de uma base de distribuição através de caminhões tanque. Este problema configura-se como a roteirização de parte uma frota disponível heterogênea e veículos compartimentados com objetivo de minimizar o custo total para realizar a entrega de um conjunto de pedidos a diversos clientes. Embora seja um problema de roteirização de veículos, suas características e restrições, em particular as restrições de capacidade dos diferentes compartimentos de cada veículo, bem como as restrições de tamanho e tipo de veículo que pode atender cada cliente, tornam esse problema complexo e impedem a adoção das heurísticas propostas para outros problemas de roteirização. A pesquisa bibliográfica realizada indicou que este é um problema pouco estudado na literatura. Os únicos trabalhos encontrados foram de Bechara e Galvão (1984), Brown e Graves (1981), Brown *et al.* (1987) e Avella, Boccia e Sforza (2004).

Inicialmente foi proposta uma formulação matemática inédita que representasse corretamente o problema. O modelo foi testado e validado através de um problema de pequeno porte, e resolvido inicialmente em ambiente de planilha eletrônica, utilizando o Solver do Excel. Tendo em vista as restrições de tamanho de máximo de problema que pode ser resolvido no Solver, bem como a dificuldade de estruturar um problema de maior porte em ambiente de planilha eletrônica e a ausência de algoritmos específicos para problemas de programação inteira no Solver, o modelo matemático foi implementado no CPLEX 9.0 da ILOG, utilizando a biblioteca *Concert Technology* para linguagem C++. Algumas instâncias para teste maiores foram geradas, entretanto, mesmo estando bastante aquém do tamanho usual de problema encontrado na prática, não foi possível determinar as respectivas soluções ótimas, evidenciando a complexidade exponencial desse problema.

Assim, com a finalidade de resolver instâncias de problemas reais de entrega de derivados, foi então proposta uma estratégia de solução heurística, inspirada em conceitos do GRASP, originalmente proposto por Feo e Resende (1989). Concluiu-se por uma estratégia de múltiplas inicializações de uma heurística rápida, com a inclusão de um fator aleatório que permita, de forma guiada, gerar variabilidade nas soluções encontradas.

A estratégia de solução proposta compreende uma heurística com duas partes principais. A primeira é uma heurística rápida de alocação de pedidos aos caminhões, baseado no menor custo adicional e orientado por parâmetros de capacidade de agrupamento e de restrições de alocação. A segunda parte constitui a geração de ciclos de desalocação baseados em sorteios, que permitissem gerar variabilidade nas soluções encontradas.

Como não existe uma base de dados de problemas teste disponível para esse tipo de problema abordado e os métodos exatos não puderam resolver instâncias nem menores que as de tamanho real, foram criadas classes de problema de tamanhos compatíveis com os problemas reais dessa natureza. Para garantir a confiabilidade nesses resultados, foram geradas e testadas diversas instâncias, com variações no tamanho do problema e nível de restrição.

Os testes desenvolvidos para avaliar o método heurístico foram elaborados com características que permitiram a avaliação da variação quantitativa de suas características nos resultados. Uma vez que fatores aleatórios inerentes à heurística tornam seu resultado variável, foram executadas cinco rodadas de cada instância.

Os ciclos de desalocação são efetuados sobre uma fração da frota total alocada. Os testes demonstraram que uma fração de $\frac{1}{4}$ da frota obteve resultados significativamente melhores que ciclos aplicados sobre $\frac{1}{2}$, tanto em termos da qualidade da solução quanto no tempo total de processamento.

Ressalva-se que esse e outros parâmetros que compõem o algoritmo não foram testados à exaustão, porém, recomenda-se que sejam sempre testados em instâncias de diferentes tamanhos e características como a avaliação aqui realizada.

Os resultados obtidos são comprovadamente factíveis. Se o nível de otimalidade do resultado não pode ser comprovado, ao menos foi comprovada a aplicabilidade prática do método desenvolvido no suporte à alocação feita hoje manualmente em diversos locais. Vale lembrar que ambos os trabalhos de Brown e Graves (1981) e Brown *et al.* (1987) descreveram sistemas de suporte que dificilmente poderiam eliminar o operador humano, responsável pelo julgamento e atuação em casos de exceção. Porém é incontestável o auxílio e a economia de recursos que um método computacional de otimização pode fornecer.

Para permitir o teste das soluções aqui obtidas, permanece o desafio por buscar referências. A primeira sugestão é realizar testes com dados reais e compará-los aos resultados obtidos manualmente pelas distribuidoras.

Outra sugestão é a formulação do problema como um problema de particionamento de conjuntos (*set partitioning*), que pode ser resolvido através da enumeração de todas as possíveis combinações de alocação para gerar colunas de entrada para um método de solução exata de programação de viagens. A enumeração de todas as combinações possíveis pode ser gerada computacionalmente, porém o número de colunas também é exponencial em relação ao tamanho do problema, o que pode inviabilizar o método. Alternativamente pode-se pensar numa solução mais sofisticada baseada no método de geração de colunas.

Outra sugestão que pode ser feita é a introdução de um novo mecanismo de melhoria de resultados, que efetue ao final de cada iteração uma busca por trocas do tipo *2-opt* ou *3-opt*, possivelmente inserido num esquema de controle utilizando busca tabu. Entretanto, conforme discutido ao longo do trabalho, estima-se que poucas trocas sejam possíveis, visto que existem muitas restrições além da variação de volume dos pedidos.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGÊNCIA NACIONAL DE PETRÓLEO - ANP; **Anuário Estatístico Brasileiro do Petróleo e do Gás Natural – 2001**, 2001.
- ASSAD, A. A. Modeling and implementation issues in vehicle routing. In: **Vehicle Routing: Methods and Studies**, B.L.Golden, A.A.Assad (eds), Amsterdam: North Holland, p. 7-46, 1988.
- AVELLA, P.; BOCCIA, M.; SFORZA, A. Solving a Fuel Delivery Problem by Heuristic and Exact Approaches. **European Journal of Operational Research**, vol.152, pp.170-179, 2004.
- BECHARA, J. J. B.; GALVÃO, R. D. O Uso de Sistemas Computacionais Interativos na Solução de Problemas de Roteamento de Veículos. **Anais do XVII Simpósio da SOBRAPO**, Rio de Janeiro: pp.279-293, 1984.
- BODIN, L.; GOLDEN, B.; ASSAD, A.; BALL, M. Routing and Scheduling of Vehicles and Crews – The State of the Art. **Computers and Operations Research**, v.10, pp.63-211, 1983.
- BROWN, G. G. *et al.* Real-Time, Wide Area Dispatch of Mobil Tank Trucks, **Interfaces**, v. 17, pp.107-120, 1987.
- BROWN, G. G.; GRAVES, G. Real-time dispatch of petroleum tank trucks, **Management Science**, v. 27, pp.19-32, 1981.
- CUNHA, C. B. Aspectos práticos da aplicação de modelos de roteirização de veículos a problemas reais. **Transportes**, v.8 , n.2, pp.51-74, 2000.
- CUNHA, C. B. **Uma contribuição para o problema de roteirização de veículos com restrições operacionais**. São Paulo: EPUSP, Departamento de Engenharia de Transportes. 222p. (Tese de Doutorado), 1997.
- DANTZIG, G. B.; RAMSER, J. H. The Truck Dispatching Problem. **Management Science**, vol. 6, pp. 80-91, 1959.

- DELORME, X.; GANDIBLEUX, X.; RODRIGUEZ, J. GRASP for set packing problems. **European Journal of Operational Research**, v. 153, pp. 564-580, 2004.
- ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO. **Diretrizes para Apresentação de Dissertações e Teses**, 2 ed., 2001.
- FEO, T.A.; RESENDE, M.G. A probabilistic heuristic for a computationally difficult set covering problem, **Operations Research Letters**, v.8, pp. 67–71, 1989.
- FEO, T.A.; RESENDE, M.G.C. Greedy randomized adaptive search procedures, **J. of Global Optimization**, v.6, pp. 109--133, 1995.
- FONSECA, C. H. F. **Alocação de atendimentos de coleta no transporte de remessas expressas em grandes centros urbanos**. São Paulo: EPUSP, Departamento de Engenharia de Transportes. 91p. (Dissertação de Mestrado), 2002.
- FUNARO, V. M. B.O. (Coord.). **Diretrizes para Apresentações de Dissertações e Teses da USP**. São Paulo: SIBi-USP, 2004.
- GALVÃO JÚNIOR, F. A. **Otimização do sistema de coleta de resíduos de biomassa de madeira para fins energéticos**. São Paulo: EPUSP, Departamento de Engenharia de Transportes, 196 p (Dissertação de Mestrado), 2004.
- GAVISH, B.; GRAVES, S. **The Traveling Salesman Problem and Related Problems**. Graduate School of Management, University of Rochester, 1978.
- GEIPOT – EMPRESA BRASILEIRA DE PLANEJAMENTO DOS TRANSPORTES. **Anuário Estatístico dos Transportes 2001**. Disponível em: < <http://www.geipot.gov.br> >. Acesso em: 6 setembro 2004
- GENDREAU, M., LAPORTE, G., MASARAGANYI, C.; TAILLARD, E.D. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. **Computers and Operations Research**, V. 26, p.1153-1173, 1999.
- HALL, R.W.; PARTYKA, J.G. On the road to efficiency. **OR/MS Today**, jun/97, p.38-47, 1997.

- INSTITUTO NACIONAL DE METROLOGIA, NORMALIZAÇÃO E QUALIDADE INDUSTRIAL - INMETRO. **Norma N° NIE-DIMEL-020: Verificação e Inspeção de Veículos-Tanque Rodoviários**. Rio de Janeiro: INMETRO, 1998.
- INSTITUTO NACIONAL DE METROLOGIA, NORMALIZAÇÃO E QUALIDADE INDUSTRIAL - INMETRO. **Portaria nº 059**, de 19 de março de 1993. Rio de Janeiro: INMETRO, 1993.
- LAPORTE, G.; GENDREAU, M.; POTVIN, J.Y.; SEMET, F. Classical and modern heuristics for the vehicle routing problem. **International Transactions in Operational Research**, v.7, n4/5, p.285-300, 2000.
- PIRLOT, M. General Local Search Methods. **European Journal of Operational Research**, v.92, pp. 493-511, 1996.
- SEDGEWICK, R. **Algorithms in C++**. 3rd Edition. Boston: Addison-Wesley, 2002. v.2.
- SILVEIRA, J. P. **A Abertura do Mercado de Abastecimento de Combustíveis: A Nova Estrutura Tributária e a Evolução da Desregulamentação de Preços**. Brasília: Agência Nacional de Petróleo, Superintendência de Estudos Estratégicos. 15p., 2002.
- SOUZA, E. C. **Modelagem e resolução de um problema de transporte do tipo: "carga única - coleta e entrega" com janelas de tempo**. São Paulo: EPUSP, Departamento de Engenharia de Transportes. 89p. (Dissertação de Mestrado), 1999.
- TEIXEIRA, R. G. **Heurísticas para o problema de dimensionamento e roteirização de uma frota heterogênea utilizando o algoritmo Out-of-Kilter**. São Paulo: EPUSP, Departamento de Engenharia de Transportes. 121p. (Dissertação de Mestrado), 2001.
- TEIXEIRA, R. G.; CUNHA, C. B. Heurísticas para o Problema de Dimensionamento e Roteirização de uma Frota Heterogênea Utilizando o Algoritmo Out-of-Kilter. **Revista Transportes**. São Paulo: , v.10, n.2, pp. 9-30, 2002.
- THE C++ RESOURCES NETWORK. **Description of C++**. Disponível em: <<http://www.cplusplus.com/info/description.html>>. Acesso em: 4 agosto 2004.

ZNAMENSKY, A. **Um modelo para a roteirização e programação do transporte de deficientes.** São Paulo: EPUSP, Departamento de Engenharia de Transportes. 144p. (Dissertação de Mestrado), 2000.

APÊNDICE A – MODELAGEM PARA EVITAR *SUBTOURS*

Este apêndice tem por objetivo apresentar, de maneira mais detalhada, as restrições (9) e (10) do modelo matemático proposto para o problema da distribuição de combustível a partir de uma base de abastecimento (seção 3.2.6), que impedem a formação de sub-rotas (*subtours*) nas soluções da modelagem matemática proposta.

Essas restrições foram formuladas e adaptadas tendo por base a proposta de Gavish e Graves (1978) para o Problema do Caixeiro Viajante. Segundo os autores, definem-se variáveis de fluxo y_{ij} relacionadas aos arcos (i, j) que fazem parte do roteiro de cada veículo, sendo essa restrição modelada com base na hipótese de que $(n-1)$ unidades são produzidas na origem (o nó 1, que corresponde à *base*) e cada outro nó da rede demanda 1 dessas unidades (lembrando que n representa o número de nós ou pontos do problema).

Assim, em torno de cada nó exceto a *base*, somadas as variáveis de fluxo y_{ij} que saem de cada nó e subtraídas as que chegam em cada nó (isto é $y_{ij} - y_{ji}$ para o nó i), o resultado do balanço deve ser igual a -1.

Se essa condição for suprida em torno de todos os nós exceto a base, não existe formação de *subtours*, como ilustrado no exemplo da Figura A.1, com $n=6$.

Figura A.1 – Rede Ilustrativa sem *Subtours*

Caso houvesse algum *subtour*, o valor dessa diferença seria diferente de -1 em algum outro ponto que não a base, como ilustrado na Figura A.2, mais especificamente no nó indicado em vermelho.

Figura A.2 – Rede Ilustrativa com Dois Subtours

Essa restrição foi generalizada para o caso do problema de roteirização de veículos objeto deste trabalho, em que as variáveis de fluxo Z_{ij}^v são definidas para cada um dos veículos.

APÊNDICE B – IMPLEMENTAÇÃO EM ILOG CPLEX PARA C++

Conforme indicado na seção 4.2.1, apresenta-se aqui a implementação do modelo matemático proposto para o problema de distribuição de derivados de petróleo no pacote *ILOG CPLEX* para C++ na Figura B.1 a seguir.

A instância deste exemplo é a mesma apresentada na seção 4.1, resolvida em aplicativo *Solver*, de um problema de entrega de 10 pedidos a partir de uma base, para 4 clientes, com 4 veículos disponíveis.

```
//
// Algoritmo_CPLEX.cpp
//
#define _ILOG_

#include <stdio.h>
#include <windows.h>
#include <string>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <math.h>
#include <time.h>

#ifdef _ILOG_
#include <ilcplex/ilocplex.h>
#include <ilconcert/ilomodel.h>
ILOSTLBEGIN //ILOG standard template library
#endif

using namespace std;

#ifdef _ILOG_
typedef IloArray<IloNumArray> IloNumArray2;
typedef IloArray<IloFloatArray> IloFloatArray2;
typedef IloArray<IloIntArray> IloIntArray2;

typedef IloArray<IloNumVarArray> IloNumVarArray2;
typedef IloArray<IloNumVarArray2> IloNumVarArray3;
#endif

void main()
{
    // indices
    int p, v, i, j, k;

    //nomes para variaveis
    char str[30];

    // criando ambiente
    IloEnv env;

    // criando modelo
    IloModel mymodel(env);

    // criando parametros f(v) (Veiculos)
```

```

IloNumArray cf(env, 4, 10, 10, 10, 10);
IloNumArray cv(env, 4, 1.00, 1.10, 0.75, 1.00);
IloIntArray v5(env, 4, 2, 6, 3, 2);
IloNumArray vMax(env, 4, 30, 30, 15, 30);

// criando parametros f(i,j) (Clientes x Clientes)
IloNumArray2 d(env, 5);
d[0] = IloNumArray(env, 5, 0.0, 1.0, 1.4, 2.0, 2.8);
d[1] = IloNumArray(env, 5, 1.0, 0.0, 2.0, 3.0, 3.6);
d[2] = IloNumArray(env, 5, 1.4, 2.0, 0.0, 3.6, 3.0);
d[3] = IloNumArray(env, 5, 2.0, 3.0, 3.6, 0.0, 2.0);
d[4] = IloNumArray(env, 5, 2.8, 3.6, 3.0, 2.0, 0.0);

// criando parametros f(i,v) (Clientes x Veiculos)
IloIntArray2 r(env, 5);
// r[0] é a base
r[1] = IloIntArray(env, 4, 1, 1, 1, 1, 1);
r[2] = IloIntArray(env, 4, 1, 1, 1, 1, 1);
r[3] = IloIntArray(env, 4, 1, 1, 1, 1, 1);
r[4] = IloIntArray(env, 4, 1, 1, 1, 1, 1);

// criando parametros f(p) (Pedidos)
IloIntArray cliente(env, 10, 1, 1, 1, 2, 2, 2, 3, 3, 4, 4); //precisa
respeitar indice i
IloNumArray vol(env, 10, 10, 25, 5, 5, 10, 10, 5, 5, 5, 10);
IloIntArray r5(env, 10, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0); //Pode ser
automatizado

//criando variaveis de decisao f(v,i,j) (Veiculo x Cliente x Cliente)

IloNumVarArray3 x(env, 4); //veiculos
for(v=0; v<4; v++){
    x[v] = IloNumVarArray2(env, 5);
    for(i=0; i<5; i++){
        x[v][i] = IloNumVarArray(env, 5, 0, 1, ILOINT);
        for (int j=0; j<5; j++)
        {
            sprintf(str, "x[%d,%d,%d]", v,i,j);
            x[v][i][j].setName(str);
        }
    }
}

//criando variaveis de decisao f(p,v) (Pedido x Veiculo)
IloNumVarArray2 y(env, 10);
for(p=0; p<10; p++){
    y[p] = IloNumVarArray(env, 4, 0, 1, ILOINT);
    for (int v=0; v<4; v++)
    {
        sprintf(str, "y[%d,%d]", p,v);
        y[p][v].setName(str);
    }
}

//criando variaveis de decisao f(v,i,j) (Veiculo x Cliente x Cliente)
IloNumVarArray3 z(env, 4); //veiculos
for(v=0; v<4; v++){
    z[v] = IloNumVarArray2(env, 5);
    for(i=0; i<5; i++){
        z[v][i] = IloNumVarArray(env, 5, 0, IloInfinity);
        for (int j=0; j<5; j++)
        {
            sprintf(str, "z[%d,%d,%d]", v,i,j);
            z[v][i][j].setName(str);
        }
    }
}

```



```

    }
}

//restricao de Pedido x Veiculo
for(p=0; p<10 ;p++){
    IloExpr expr(env);
    for(v=0; v<4; v++){
        expr += y[p][v];
    }
    mymodel.add(expr == 1);
    expr.end();
}

//restricao de Compartimentos5
for(v=0; v<4 ;v++){
    IloExpr expr(env);
    for(p=0; p<10; p++){
        expr += r5[p] * y[p][v];
    }
    mymodel.add(expr <= v5[v]);
    expr.end();
}

//restricao de Volumna Maximo
for(v=0; v<4 ;v++){
    IloExpr expr(env);
    for(p=0; p<10; p++){
        expr += vol[p] * y[p][v];
    }
    mymodel.add(expr <= vMax[v]);
    expr.end();
}

// restricao Caminhão atende Cliente do Pedido
for(p=0; p<10; p++){
    for(v=0; v<4; v++){
        IloExpr expr(env);
        for(i=0; i<5; i++){
            expr += x[v][i][cliente[p]];
        }
        mymodel.add(expr >= y[p][v]);
        expr.end();
    }
}

// restricao Veículo respeita Cliente
for(v=0; v<4; v++){
    for(j=1; j<5; j++){ //exclui a base
        IloExpr expr(env);
        for(i=0; i<5; i++){
            expr += x[v][i][j];
        }
        mymodel.add(expr <= r[j][v]);
        expr.end();
    }
}

// restricao Continuidade das Rotas
for(v=0; v<4; v++){
    for(k=0; k<5; k++){
        IloExpr expr(env);
        for(i=0; i<5; i++){
            for(j=0; j<5; j++){
                expr += x[v][i][k] - x[v][k][j];
            }
        }
        mymodel.add(expr == 0);
    }
}

```

```

        expr.end();
    }
}

// restricao Veículo faz no máximo uma Rota
for(v=0; v<4; v++){
    IloExpr expr(env);
    for(j=0; j<5; j++){
        expr += x[v][0][j]; //a partir da base
    }
    mymodel.add(expr <= 1);
    expr.end();
}

// restricao Evitar sub-rotas - Sequência
for(v=0; v<4; v++){
    for(k=1; k<5; k++){ //exclui a base
        IloExpr expr(env);
        IloExpr expr2(env);
        for(i=0; i<5; i++){
            for(j=0; j<5; j++){
                expr += z[v][k][j] - z[v][i][k];
                expr2 += -1 * x[v][k][j];
            }
        }
        mymodel.add(expr == expr2);
        expr.end();
        expr2.end();
    }
}

// restricao Evitar sub-rotas - Respeito
for(v=0; v<4; v++){
    for(i=0; i<5; i++){
        for(j=0; j<5; j++){
            mymodel.add(z[v][i][j] <= 4 * x[v][i][j]); //4 =
|I|-1
        }
    }
}

//criando funcao objetivo
IloExpr obj(env);
for(v=0; v<4; v++){
    for(j=0; j<5; j++){
        obj += cf[v] * x[v][0][j]; // parcela custo fixo
        for(i=0; i<5; i++){
            obj += cv[v] * d[i][j] * x[v][i][j]; // parcela
custo variável
        }
    }
}
mymodel.add(IloMinimize(env, obj));

// adicionando funcao objetivo ao modelo
mymodel.add(obj);

// ligando modelo ao algoritmo
IloCplex mycplex(mymodel);

mycplex.exportModel("distcomb.mps");

// resolvendo o problema
mycplex.solve();

//extraindo resultados

```

```
cout << "Solution Status is " << mycplex.getStatus() << endl;  
cout << " The objective value is " << mycplex.getObjValue() << endl;  
  
// finalizando  
env.end();  
  
} // main()
```

Figura B.1 – Exemplo de Implementação em ILOG CPLEX para C++

APÊNDICE C – PLANILHA DE AUXÍLIO NO DESENVOLVIMENTO

Conforme descrito na seção 5.1, para subsidiar o desenvolvimento da estratégia heurística proposta foi inicialmente desenvolvido um exemplo contendo 26 clientes (postos de combustíveis), 36 pedidos e uma frota disponível de 55 caminhões tanque.

O exemplo foi todo modelado em uma planilha eletrônica para testar e verificar, passo a passo, a consistência dos resultados de cada etapa de desenvolvimento do algoritmo.

Inicialmente todos os dados foram listados, com as respectivas características, conforme figuras a seguir. Conforme a Figura C.1, para cada cliente, nomeados “A” a “Z”, foram indicadas as coordenadas do local, se o cliente aceita (1) ou não (0) cada padrão de pintura de veículo (cor) e se existe limitação do cliente a aceitar apenas caminhões menores, do tipo *truck*.

Cliente	X	Y	Cor 0	Cor 1	Cor 2	Cor 3	Cor 4	Só T?
BASE	40	60						
A	93,7	70,6	1	1	1	1	1	0
B	13,3	16,9	1	1	1	1	1	0
C	23,4	99,8	1	1	1	1	1	0
D	27	91	1	1	1	1	1	0
E	37,5	11,1	1	1	1	1	1	1
F	74,3	84,5	1	0	0	0	0	0
G	30,5	18,1	1	0	0	0	0	0
H	2,9	94	1	1	0	0	0	1
I	34,4	44,9	1	1	0	0	0	0
J	5,1	8,2	1	1	0	0	0	0
K	55,7	70,4	1	1	0	0	0	0
L	14,5	52,2	1	0	1	0	0	0
M	62,3	13,2	1	0	1	0	0	0
N	24,6	90	1	0	1	0	0	0
O	7,5	28,5	1	0	1	0	0	1
P	4	95,7	1	0	1	0	0	0
Q	7,6	26,9	1	0	0	1	0	0
R	57,8	57,7	1	0	0	1	0	1
S	97,7	31,7	1	0	0	1	0	0
T	55,1	94,5	1	0	0	1	0	0
U	31,4	88,2	1	0	0	0	1	0
V	78,1	59	1	0	0	0	1	1
W	16	94	1	0	0	0	1	0
X	98,9	18,1	1	0	0	0	1	0
Y	76,9	80,3	0	0	0	0	1	0
Z	50,8	85,6	0	0	0	0	1	1

Figura C.1 – Dados de clientes

Para cada pedido foram indicados o cliente e o volume do pedido. Automaticamente são verificados os dados do cliente correspondente, ilustrado na Figura C.2.

Pedido [/]	Local	Volume	X	Y	Cor 0	Cor 1	Cor 2	Cor 3	Cor 4	Só T?
0	A	30	93,7	70,6	1	1	1	1	1	0
1	A	15	93,7	70,6	1	1	1	1	1	0
2	A	5	93,7	70,6	1	1	1	1	1	0
3	B	5	13,3	16,9	1	1	1	1	1	0
4	B	5	13,3	16,9	1	1	1	1	1	0
5	B	5	13,3	16,9	1	1	1	1	1	0
6	C	5	23,4	99,8	1	1	1	1	1	0
7	D	20	27	91	1	1	1	1	1	0
8	E	15	37,5	11,1	1	1	1	1	1	1
9	F	20	74,3	84,5	1	0	0	0	0	0
10	F	15	74,3	84,5	1	0	0	0	0	0
11	F	10	74,3	84,5	1	0	0	0	0	0
12	G	20	30,5	18,1	1	0	0	0	0	0
13	H	15	2,9	94	1	1	0	0	0	1
14	I	20	34,4	44,9	1	1	0	0	0	0
15	J	20	5,1	8,2	1	1	0	0	0	0
16	K	10	55,7	70,4	1	1	0	0	0	0
17	L	25	14,5	52,2	1	0	1	0	0	0
18	M	30	62,3	13,2	1	0	1	0	0	0
19	N	10	24,6	90	1	0	1	0	0	0
20	O	5	7,5	28,5	1	0	1	0	0	1
21	P	20	4	95,7	1	0	1	0	0	0
22	Q	10	7,6	26,9	1	0	0	1	0	0
23	Q	10	7,6	26,9	1	0	0	1	0	0
24	Q	20	7,6	26,9	1	0	0	1	0	0
25	R	5	57,8	57,7	1	0	0	1	0	1
26	S	20	97,7	31,7	1	0	0	1	0	0
27	T	10	55,1	94,5	1	0	0	1	0	0
28	U	10	31,4	88,2	1	0	0	0	1	0
29	V	5	78,1	59	1	0	0	0	1	1
30	V	5	78,1	59	1	0	0	0	1	1
31	V	10	78,1	59	1	0	0	0	1	1
32	W	25	16	94	1	0	0	0	1	0
33	X	30	98,9	18,1	1	0	0	0	1	0
34	Y	10	76,9	80,3	0	0	0	0	1	0
35	Z	5	50,8	85,6	0	0	0	0	1	1

Figura C.2 – Dados de pedidos

Para cada veículo da frota, na Figura C.3 são indicados o tipo do veículo (1, 2 ou 3, conforme compartimentos), se é próprio (P) ou terceirizado (T) e o padrão de pintura (Cor, de 0 a 4).

Caminhão [/]	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Tipo	3	2	1	3	2	1	3	1	1	3	3	2	1	2	2	1	2	1	3	1	2	2	3	2	1
Próprio/Terceiro	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
Cor	0	0	0	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	4

25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
3	1	3	1	3	3	2	2	2	2	3	1	2	2	1	3	2	3	2	2	2	2	2	2	3	3	2	3	3	3
P	P	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
4	0	0	0	0	0	1	1	1	1	2	2	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	4	4

Figura C.3 – Dados de frota

A codificação utilizada para identificar o tipo dos caminhões apresenta-se na Tabela C.1 a seguir.

Tabela C.1 – Codificação dos Tipos de Caminhões

Tipo	Tamanho	Volume Máximo (litros)	Compartimentalização (10³ litros)
1	Truck	15.000	5 - 5 - 5
2	Carreta	30.000	5 - 5 - 5 - 5 - 5 - 5
3	Carreta	30.000	10 - 5 - 10 - 5

Os pontos das coordenadas dos clientes foram mapeadas para a visualização de sua distribuição espacial, conforme observa-se na Figura C.4.

Figura C.4 – Mapeamento dos clientes

Todos as restrições foram identificadas e mapeadas para a verificação da viabilidade das soluções geradas, conforme Figura C.5 a seguir. Os parâmetros de facilidade e serventia, também foram calculados.

Conforme se verifica na figura, para cada par pedido (i) - caminhão (j), são verificadas as restrições e indicados se é possível (1) ou não é possível (0) a alocação de cada pedido em cada caminhão. A serventia e a facilidade são as respectivas somas dos valores de cada coluna e linha.

$[i] \setminus [j]$	0	1	2	3	4	5	50	51	52	53	54	Facilidade
0	1	1	0	1	1	0	1	1	1	1	1	42
1	1	1	1	1	1	1	1	1	1	1	1	55
2	1	1	1	1	1	1	1	1	1	1	1	55
3	1	1	1	1	1	1	1	1	1	1	1	55
4	1	1	1	1	1	1	1	1	1	1	1	55
5	1	1	1	1	1	1	1	1	1	1	1	55
30	0	0	1	0	0	0	0	0	0	0	0	5
31	0	0	1	0	0	0	0	0	0	0	0	5
32	1	1	0	0	0	0	1	1	1	1	1	19
33	1	1	0	0	0	0	1	1	1	1	1	19
34	0	0	0	0	0	0	1	1	1	1	1	16
35	0	0	0	0	0	0	0	0	0	0	0	2
Serventia	27	27	21	11	11	9	12	12	12	12	12	

Figura C.5 – Matriz de restrições

De maneira análoga, foi concebida uma matriz para indicação da alocação em curso e verificação das outras restrições, exemplificada na Figura C.6. São indicadas com o valor “1” as alocações de um pedido i em um veículo j . As restrições de compartimentos de 5.000 necessários (*Min. Co. 5*) e disponíveis (*Max. Co. 5*), volume ocupado (*Volume*) e volume máximo (*Max Vol.*) também estão indicados para cada veículo.

[I] \ [J]	0	1	2	3	4	5	50	51	52	53	54	Facilidade
0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	50
4	0	0	0	0	0	0	0	0	0	0	0	50
5	0	0	1	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	4
31	0	0	0	0	0	0	0	0	0	0	0	4
32	0	0	0	0	0	0	0	0	0	0	0	17
33	0	0	0	0	0	0	0	0	0	0	0	17
34	0	0	0	0	0	0	0	0	0	0	0	16
35	0	0	0	0	0	0	0	0	0	0	0	2
Serventia	0	0	0	0	3	0	8	8	8	8	8	
Min Co.5	0	1	1	0	0	0	0	0	0	0	0	
Max Co.5	2	6	3	2	6	3	2	6	2	2	2	
Volume	30	30	15	30	20	15	0	0	0	0	0	
Max Vol.	30	30	15	30	30	15	30	30	30	30	30	

Figura C.6 – Matriz de alocação

APÊNDICE D – RESULTADOS DOS PROCESSAMENTOS

Conforme indicado na seção 6.5, estão indicados aqui todos os resultados de todas as rodas de teste realizadas.

Para cada teste são apresentados os seguintes resultados:

- Custo da Melhor Solução Obtida – Menor custo obtido em relação às 10 rodadas (5 rodadas com cada parâmetro de desalocação), que resulta da melhor solução encontrada.
- Custo Médio – Resultado da função objetivo médio obtido pelas cinco rodadas.
- Desvio Padrão – Desvio padrão entre os resultados das cinco rodadas.
- Custo da Solução Inicial – Custo do resultado obtido no primeiro resultado da heurística, sem os ciclos de desalocação e re-alocação aleatórios.
- Melhoria Média – Percentual ou valor absoluto médio de melhoria da solução inicial para o fim dos ciclos de sorteio.
- Tempo Médio de Processamento (s) – Tempo médio de processamento, em segundos, em um computador Pentium IV, 3GHz.
- Custo da Solução da Rodada 1 a 5 – Resultados da função objetivo (custo total da solução) para cinco rodadas independentes de uma mesma instância.
- Diferença na Rodada 1 a 5 – Desvio percentual em relação ao melhor resultado obtido (o custo da melhor solução)

A Classe A corresponde ao problema com 50 pedidos, 50 veículos e 35 clientes. A Tabela D.1, Tabela D.2 e Tabela D.3 apresentam os resultados para os níveis de restrição de 20%, 40% e 60%, tanto com taxa de desalocação de $\frac{1}{2}$ quanto de $\frac{1}{4}$.

A Classe B corresponde ao problema com 75 pedidos, 75 veículos e 53 clientes. A Tabela D.4, Tabela D.5 e Tabela D.6 apresentam os resultados para os níveis de restrição de 20%, 40% e 60%, tanto com taxa de desalocação de $\frac{1}{2}$ quanto de $\frac{1}{4}$.

A Classe C corresponde ao problema com 100 pedidos, 100 veículos e 70 clientes. A Tabela D.7, Tabela D.8 e Tabela D.9 apresentam os resultados para os níveis de restrição de 20%, 40% e 60%, tanto com taxa de desalocação de $\frac{1}{2}$ quanto de $\frac{1}{4}$.

Tabela D.1 – Resultados para Classe A – Restrição 20%

Instância	A - 20% - 1/2	A - 20% - 1/4
Veículos Disponíveis	35	35
Nº de Pedidos	50	50
Nº de Caminhões	50	50
Nível de Restrição	20%	20%
Taxa de Desalocação	1/2	1/4
Custo da Melhor Solução Obtida	6.343,40	
Custo Médio	6.580,06	6.385,28
Desvio Padrão	3.468,65	3.365,48
Custo da Solução Inicial	7.998,23	7.998,23
Melhoria Média	17,7%	20,2%
Tempo Médio de Processamento (s)	57	36
Custo da Solução na Rodada 1	6.676,23	6.343,40
Diferença na Rodada 1	5,2%	0,0%
Custo da Solução na Rodada 2	6.536,46	6.398,60
Diferença na Rodada 2	3,0%	0,9%
Custo da Solução na Rodada 3	6.632,07	6.456,51
Diferença na Rodada 3	4,6%	1,8%
Custo da Solução na Rodada 4	6.432,78	6.364,12
Diferença na Rodada 4	1,4%	0,3%
Custo da Solução na Rodada 5	6.622,77	6.363,75
Diferença na Rodada 5	4,4%	0,3%

Tabela D.2 – Resultados para Classe A – Restrição 40%

Instância	A - 40% - 1/2	A - 40% - 1/4
Veículos Disponíveis	35	35
Nº de Pedidos	50	50
Nº de Caminhões	50	50
Nível de Restrição	40%	40%
Taxa de Desalocação	1/2	1/4
Custo da Melhor Solução Obtida	6.465,03	
Custo Médio	6.717,00	6.508,27
Desvio Padrão	3.540,56	3.430,37
Custo da Solução Inicial	8.035,50	8.035,50
Melhoria Média	16,4%	19,0%
Tempo Médio de Processamento (s)	56	37
Custo da Solução na Rodada 1	6.644,85	6.480,29
Diferença na Rodada 1	2,8%	0,2%
Custo da Solução na Rodada 2	6.799,55	6.595,68
Diferença na Rodada 2	5,2%	2,0%
Custo da Solução na Rodada 3	6.672,60	6.472,01
Diferença na Rodada 3	3,2%	0,1%
Custo da Solução na Rodada 4	6.800,61	6.528,33
Diferença na Rodada 4	5,2%	1,0%
Custo da Solução na Rodada 5	6.667,38	6.465,03
Diferença na Rodada 5	3,1%	0,0%

Tabela D.3 – Resultados para Classe A – Restrição 60%

Instância	A - 60% - ½	A - 60% - ¼
Veículos Disponíveis	35	35
Nº de Pedidos	50	50
Nº de Caminhões	50	50
Nível de Restrição	60%	60%
Taxa de Desalocação	½	¼
Custo da Melhor Solução Obtida	6.404,56	6.404,56
Custo Médio	6.706,18	6.578,70
Desvio Padrão	3.536,47	3.467,95
Custo da Solução Inicial	8.230,63	
Melhoria Média	18,5%	20,1%
Tempo Médio de Processamento (s)	57	37
Custo da Solução na Rodada 1	6.791,17	6.404,56
Diferença na Rodada 1	6,0%	0,0%
Custo da Solução na Rodada 2	6.428,44	6.636,76
Diferença na Rodada 2	0,4%	3,6%
Custo da Solução na Rodada 3	6.841,93	6.614,60
Diferença na Rodada 3	6,8%	3,3%
Custo da Solução na Rodada 4	6.664,30	6.616,76
Diferença na Rodada 4	4,1%	3,3%
Custo da Solução na Rodada 5	6.805,06	6.620,80
Diferença na Rodada 5	6,3%	3,4%

Tabela D.4 – Resultados para Classe B – Restrição 20%

Instância	B - 20% - ½	B - 20% - ¼
Veículos Disponíveis	53	53
Nº de Pedidos	75	75
Nº de Caminhões	75	75
Nível de Restrição	20%	20%
Taxa de Desalocação	½	¼
Custo da Melhor Solução Obtida	9.143,79	
Custo Médio	9.407,89	9.254,56
Desvio Padrão	4.958,55	4.877,97
Custo da Solução Inicial	10.281,30	10.281,30
Melhoria Média	8,5%	10,0%
Tempo Médio de Processamento (s)	152	100
Custo da Solução na Rodada 1	9.467,63	9.224,17
Diferença na Rodada 1	3,5%	0,9%
Custo da Solução na Rodada 2	9.446,28	9.216,18
Diferença na Rodada 2	3,3%	0,8%
Custo da Solução na Rodada 3	9.404,13	9.143,79
Diferença na Rodada 3	2,8%	0,0%
Custo da Solução na Rodada 4	9.406,54	9.336,46
Diferença na Rodada 4	2,9%	2,1%
Custo da Solução na Rodada 5	9.314,87	9.352,18
Diferença na Rodada 5	1,9%	2,3%

Tabela D.5 – Resultados para Classe B – Restrição 40%

Instância	B - 40% - ½	B - 40% - ¼
Veículos Disponíveis	53	53
Nº de Pedidos	75	75
Nº de Caminhões	75	75
Nível de Restrição	40%	40%
Taxa de Desalocação	½	¼
Custo da Melhor Solução Obtida	9.236,31	
Custo Médio	9.368,26	9.286,97
Desvio Padrão	4.937,63	4.894,85
Custo da Solução Inicial	10.431,20	10.431,20
Melhoria Média	10,2%	11,0%
Tempo Médio de Processamento (s)	153	99
Custo da Solução na Rodada 1	9.429,31	9.273,89
Diferença na Rodada 1	2,1%	0,4%
Custo da Solução na Rodada 2	9.402,95	9.393,91
Diferença na Rodada 2	1,8%	1,7%
Custo da Solução na Rodada 3	9.372,60	9.261,40
Diferença na Rodada 3	1,5%	0,3%
Custo da Solução na Rodada 4	9.330,71	9.269,36
Diferença na Rodada 4	1,0%	0,4%
Custo da Solução na Rodada 5	9.305,73	9.236,31
Diferença na Rodada 5	0,8%	0,0%

Tabela D.6 – Resultados para Classe B – Restrição 60%

Instância	B - 60% - ½	B - 60% - ¼
Veículos Disponíveis	53	53
Nº de Pedidos	75	75
Nº de Caminhões	75	75
Nível de Restrição	60%	60%
Taxa de Desalocação	½	¼
Custo da Melhor Solução Obtida	9.329,89	
Custo Médio	9.579,55	9.489,21
Desvio Padrão	5.050,08	5.001,58
Custo da Solução Inicial	10.433,30	10.433,30
Melhoria Média	8,2%	9,0%
Tempo Médio de Processamento (s)	155	101
Custo da Solução na Rodada 1	9.669,97	9.531,65
Diferença na Rodada 1	3,6%	2,2%
Custo da Solução na Rodada 2	9.329,89	9.366,85
Diferença na Rodada 2	0,0%	0,4%
Custo da Solução na Rodada 3	9.676,58	9.560,37
Diferença na Rodada 3	3,7%	2,5%
Custo da Solução na Rodada 4	9.703,99	9.443,19
Diferença na Rodada 4	4,0%	1,2%
Custo da Solução na Rodada 5	9.517,30	9.544,00
Diferença na Rodada 5	2,0%	2,3%

Tabela D.7 – Resultados para Classe C – Restrição 20%

Instância	C - 20% - ½	C - 20% - ¼
Veículos Disponíveis	70	70
Nº de Pedidos	100	100
Nº de Caminhões	100	100
Nível de Restrição	20%	20%
Taxa de Desalocação	½	¼
Custo da Melhor Solução Obtida	12.794,70	
Custo Médio	13.030,64	12.972,18
Desvio Padrão	6.868,17	6.837,34
Custo da Solução Inicial	14.374,40	14.374,40
Melhoria Média	9,3%	9,8%
Tempo Médio de Processamento (s)	340	214
Custo da Solução na Rodada 1	13.065,40	12.988,20
Diferença na Rodada 1	2,1%	1,5%
Custo da Solução na Rodada 2	13.106,60	12.794,70
Diferença na Rodada 2	2,4%	0,0%
Custo da Solução na Rodada 3	13.069,20	12.983,00
Diferença na Rodada 3	2,1%	1,5%
Custo da Solução na Rodada 4	12.838,90	13.068,70
Diferença na Rodada 4	0,3%	2,1%
Custo da Solução na Rodada 5	13.073,10	13.026,30
Diferença na Rodada 5	2,2%	1,8%

Tabela D.8 – Resultados para Classe C – Restrição 40%

Instância	C - 40% - ½	C - 40% - ¼
Veículos Disponíveis	70	70
Nº de Pedidos	100	100
Nº de Caminhões	100	100
Nível de Restrição	40%	40%
Taxa de Desalocação	½	¼
Custo da Melhor Solução Obtida	12.456,30	
Custo Médio	13.035,80	12.928,30
Desvio Padrão	6.871,85	6.816,91
Custo da Solução Inicial	14.425,10	14.425,10
Melhoria Média	9,6%	10,4%
Tempo Médio de Processamento (s)	341	214
Custo da Solução na Rodada 1	13.195,20	13.050,30
Diferença na Rodada 1	5,9%	4,8%
Custo da Solução na Rodada 2	12.797,80	12.456,30
Diferença na Rodada 2	2,7%	0,0%
Custo da Solução na Rodada 3	13.244,90	13.246,80
Diferença na Rodada 3	6,3%	6,3%
Custo da Solução na Rodada 4	13.072,40	12.939,00
Diferença na Rodada 4	4,9%	3,9%
Custo da Solução na Rodada 5	12.868,70	12.949,10
Diferença na Rodada 5	3,3%	4,0%

Tabela D.9 – Resultados para Classe C – Restrição 60%

Instância	C - 60% - ½	C - 60% - ¼
Veículos Disponíveis	70	70
Nº de Pedidos	100	100
Nº de Caminhões	100	100
Nível de Restrição	60%	60%
Taxa de Desalocação	½	¼
Custo da Melhor Solução Obtida	12.668,80	
Custo Médio	13.180,08	12.998,18
Desvio Padrão	6.946,63	6.852,25
Custo da Solução Inicial	14.435,90	14.435,90
Melhoria Média	8,7%	10,0%
Tempo Médio de Processamento (s)	341	214
Custo da Solução na Rodada 1	13.283,70	13.081,70
Diferença na Rodada 1	4,9%	3,3%
Custo da Solução na Rodada 2	13.125,00	13.095,80
Diferença na Rodada 2	3,6%	3,4%
Custo da Solução na Rodada 3	13.176,40	12.927,60
Diferença na Rodada 3	4,0%	2,0%
Custo da Solução na Rodada 4	13.147,30	12.668,80
Diferença na Rodada 4	3,8%	0,0%
Custo da Solução na Rodada 5	13.168,00	13.217,00
Diferença na Rodada 5	3,9%	4,3%