

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ADMINISTRAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO
MESTRADO ACADÊMICO EM ADMINISTRAÇÃO

Gilmar D'Agostini Oliveira Casalinho

**PROPOSIÇÃO DE UMA HEURÍSTICA UTILIZANDO BUSCA-
TABU PARA A RESOLUÇÃO DO PROBLEMA DE
ESCALONAMENTO DE VEÍCULOS COM MÚLTIPLAS
GARAGENS**

Porto Alegre
2012

Gilmar D'Agostini Oliveira Casalinho

Proposição de uma Heurística utilizando Busca-Tabu para a
Resolução do Problema de Escalonamento de Veículos com
Múltiplas Garagens

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Administração (PPGA) da Universidade Federal do Rio Grande do Sul (UFRGS) como requisito parcial à obtenção do título de Mestre em Administração.

Orientador: Prof. Dr. Denis Borenstein

**Porto Alegre
2012**

DEDICATÓRIA

Dedico este trabalho a todos aqueles que, em algum momento, acreditaram no meu potencial para vencer desafios e torceram pelo meu sucesso, dentre eles os meus pais que sempre colocaram a minha educação em primeiro lugar, permitindo que eu construísse uma sólida base de conhecimento, indispensável para a formação de um bom profissional.

AGRADECIMENTOS

- À Universidade Federal do Rio Grande do Sul por provir excelentes condições para minha formação e desenvolvimento profissional;
- Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela concessão de bolsa durante o curso;
- Ao meu orientador, Prof. Denis Borenstein pelo afinco que dedica à “área” de Pesquisa Operacional no PPGA, fazendo com que esta, certamente, tenha em um futuro próximo sua posição merecida na Escola.
- Aos professores da banca avaliadora deste trabalho. Primeiramente ao Prof. Fábio Schramm, por ser um exemplo profissional para mim. À Prof^a. Denise Bandeira, pelas valiosas conversas que tínhamos nos bastidores que, muitas vezes, abriam minha cabeça acerca de qual rumo deveria tomar. E, de forma não menos importante, à Prof^a Maria Claudia Boeres por ter aceitado de última hora o convite para participar desta banca, em substituição à Prof^a Luciana Buriol, a quem também devo muitos agradecimentos.
- Aos professores do Departamento de Engenharia de Produção e Transportes José Luis Ribeiro, Tarcisio Saurin, Francisco Kliemann Neto e Flávio Fogliatto, por terem acreditado no meu potencial e terem permitido que eu faça parte do corpo docente (ainda que de forma substituta) desta Universidade que tenho tanto apreço. Começar a carreira em uma universidade como a UFRGS é algo extremamente valioso e agradeço a eles por este “investimento”.
- Aos meus amigos e colegas Gabriel Braido e Monize Visentini que, juntamente comigo, compunham o Trio Ternura (apelido concedido pelo nosso orientador). Mais do que importantes, eles foram indispensáveis durante estes dois anos. Amigos que dividiram não apenas todas as disciplinas comigo, mas que também dividiram angústias, incertezas, horas e horas de estudos, além dos momentos felizes (como os jogos do Inter, fartos almoços no Gourmet da Redenção e a tradicional corrida na Beira-Rio).
- Ao melhor grupo de amigos que existe: Bruna, Cacá, Esther, Guilherme, João Pedro, Juliana, Max e Paula. Como é importante saber que sempre nos apoiamos, seja em Pelotas, Porto Alegre, São Paulo, Sydney ou em qualquer outro lugar que um de nós venha a morar.

- Aos meus *flatmates*, Fernando e George por terem sido amigos de extrema paciência e importância nos meses finais à entrega deste trabalho, assim como à Ju e à Vivi pela companhia nas noites de estudo na “república” Casemiro.
- E, por fim, não poderia deixar de agradecer ao grupo de estudos Armagedom. Grupo que se formou desde o primeiro mês de aula da nossa turma do mestrado e que seguiu firme até o final com periódicas reuniões noturnas.

RESUMO

Os problemas logísticos estão se apoiando de forma bastante expressiva na pesquisa operacional a fim de obter uma maior eficiência em suas operações. Dentre os vários problemas relacionados à designação de veículos em um sistema logístico, o de escalonamento de veículos com múltiplas garagens, MDVSP (*Multiple Depot Vehicle Scheduling Problem*), vem sendo abordado em diversas pesquisas. O MDVSP pressupõe a existência de garagens que interferem no planejamento das sequências com as quais as viagens devem ser executadas. Frequentemente, métodos exatos não podem resolver as grandes instâncias encontradas na prática e, para poder levá-las em consideração, várias abordagens heurísticas estão sendo desenvolvidas. O principal objetivo deste trabalho, portanto, foi solucionar o MDVSP através de uma heurística utilizando o método de busca-tabu. A principal motivação para a realização deste trabalho surgiu a partir da indicação de que apenas recentemente o uso de meta-heurísticas está sendo aplicado ao MDVSP (Pepin *et al.* 2008) e das limitações elencadas no estudo de Rohde (2008), o qual utilizou o algoritmo *branch-and-bound* em uma das etapas da heurística apresentada para resolver o problema, o que fez aumentar o tempo de resolução do problema. O método de pesquisa para solução deste problema foi baseado em adaptações das tradicionais técnicas de pesquisa operacional, e propiciou a resolução do MDVSP apresentando resultados bastante competitivos quanto ao custo da função objetivo, número de veículos utilizados e tempo computacional necessário.

Palavras-Chave: Pesquisa Operacional; Escalonamento de Veículos; Busca-Tabu

ABSTRACT

Currently the logistical problems are relying quite significantly on Operational Research in order to achieve greater efficiency in their operations. Among the various problems related to the vehicles scheduling in a logistics system, the Multiple Depot Vehicle Scheduling Problem (MDVSP) has been addressed in several studies. The MDVSP presupposes the existence of depots that affect the planning of sequences to which travel must be performed. Often, exact methods cannot solve large instances encountered in practice and in order to take them into account, several heuristic approaches are being developed. The aim of this study was thus to solve the MDVSP using a meta-heuristic based on tabu-search method. The main motivation for this work came from the indication that only recently the use of meta-heuristics is being applied to MDVSP context (Pepin et al. 2008) and, also, the limitations listed by Rohde (2008) in his study, which used the branch-and-bound in one of the steps of the heuristic presented to solve the problem, which has increased the time resolution. The research method for solving this problem was based on adaptations of traditional techniques of Operational Research, and provided resolutions presenting very competitive results for the MDVSP such as the cost of the objective function, number of vehicles used and computational time.

Key words: Operational Research; Vehicle Scheduling; Tabu-Search.

LISTA DE FIGURAS

Figura 1 - Exemplo de Rede para o MDVSP	27
Figura 2 - Desenho de Pesquisa	34
Figura 3 - Método Heurístico Desenvolvido	44
Figura 4 - <i>Out</i> do programa em modo <i>debug</i> para a instância m4n500s0	54

LISTA DE TABELAS

Tabela 1 - Configuração do Cenários	52
Tabela 2 - Comparação entre as soluções inicial e final para o Cenário 1.....	55
Tabela 3 - Comparação entre as soluções inicial e final para o Cenário 2.....	56
Tabela 4 - Comparação entre as soluções inicial e final para o Cenário 3.....	57
Tabela 5 - Comparação entre as soluções inicial e final para o Cenário 4.....	58
Tabela 6 - Comparação entre as soluções inicial e final para o Cenário 5.....	59
Tabela 7 - Comparação entre as soluções inicial e final para o Cenário 6.....	60
Tabela 8 - Comparação entre as soluções finais e as de Pepin et al. (2008) para o Cenário 1	63
Tabela 9 - Comparação entre as soluções finais e as de Pepin et al. (2008) para o Cenário 2	64
Tabela 10 - Comparação entre as soluções finais e as de Pepin et al. (2008) para o Cenário3	65
Tabela 11 - Comparação entre as soluções finais e as de Pepin et al. (2008) para o Cenário 4	66
Tabela 12 - Comparação entre as soluções finais e as de Pepin et al. (2008) para o Cenário 5	67
Tabela 13 - Comparação entre as soluções finais e as de Pepin et al. (2008) para o Cenário 6	68
Tabela 14 - Resumo das soluções inicial e final para os seis Cenários	69
Tabela 15 - Resumo dos Custos Finais e N° de Veículos Utilizados comparados aos de Pepin et al. (2008)	70

LISTA DE QUADROS

Quadro 1 - Pseudo-código para o algoritmo de Busca-Tabu	33
Quadro 2 - Principais artigos que abordam o MDVSP - Método de solução e formulação utilizada	36
Quadro 3 - Procedimento heurístico para geração da solução inicial	45
Quadro 4 - Procedimento heurístico para o <i>Trip-Exchange</i>	46
Quadro 5 - Procedimento heurístico para o <i>Depot-Exchange</i>	46
Quadro 6 - Estrutura de Memória para as Instâncias	49

SUMÁRIO

1	INTRODUÇÃO	13
2	CONTEXTO E JUSTIFICATIVA DO PROBLEMA	16
3	REVISÃO DE LITERATURA	20
3.1	PROBLEMAS LOGÍSTICOS E O EMPREGO DA PESQUISA OPERACIONAL COMO FERRAMENTA DE APOIO À TOMADA DE DECISÃO	20
3.2	PROBLEMAS DE DESIGNAÇÃO DE VEÍCULOS EM UM SISTEMA LOGÍSTICO	21
3.3	O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS (VSP)	22
3.4	FORMULAÇÕES PARA O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS (VSP)	23
3.5	O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS COM MÚLTIPLAS GARAGENS (MDVSP)	25
3.6	FORMULAÇÕES PARA O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS COM MÚLTIPLAS GARAGENS (MDVSP)	27
3.7	BUSCA-TABU (BT)	30
4	MÉTODO DE PESQUISA	32
4.1	ETAPA 1: FORMULAÇÕES E MÉTODOS PARA RESOLUÇÃO DO MDVSP	33
4.2	ETAPA 2: DESENVOLVIMENTO DE UMA ABORDAGEM HEURÍSTICA	38
4.2.1	Heurísticas	38
4.2.2	Trade-off entre tempo de execução e qualidade da solução	40
4.2.3	Construção heurística desenvolvida	41
4.2.3.1	Geração da Solução Viável Inicial	42
4.2.3.2	Outros mecanismos de geração de soluções	43
4.2.3.2.1	Algoritmo de <i>Trip-Exchange</i>	43
4.2.3.2.2	Algoritmo de <i>Depot-Exchange</i>	44
4.2.3.3	Critério de Aceitação	45
4.2.3.4	Critério de Parada	45
4.2.4	Procedimento tabu	45
4.2.4.1	Listas Tabu	46
4.3	ETAPA 3: IMPLEMENTAÇÃO COMPUTACIONAL	46
4.3.1	Obtenção dos dados	47
4.4	ETAPA 4: AVALIAÇÃO DA SOLUÇÃO	47
5	EXPERIMENTOS COMPUTACIONAIS E ANÁLISES	49
5.1	CONFIGURAÇÃO DOS EXPERIMENTOS	50
5.2	RESULTADOS E ANÁLISES	51
5.2.1	Formato da apresentação dos resultados	51
5.2.2	Comparação entre a solução inicial e a solução final encontrada	53
5.2.2.1	Cenário 1: 4 Garagens e 500 Viagens	53
5.2.2.2	Cenário 2: 4 Garagens e 1000 Viagens	54
5.2.2.3	Cenário 3: 4 Garagens e 1500 Viagens	55
5.2.2.4	Cenário 4: 8 Garagens e 500 Viagens	56

5.2.2.5 Cenário 5: 8 Garagens e 1000 Viagens	57
5.2.2.6 Cenário 6: 8 Garagens e 1500 Viagens	58
5.2.3 Comparação entre a solução final encontrada e a solução apresentada por Pepin et al. (2008)	59
5.2.3.1 Comparação com os resultados de Pepin <i>et al.</i> (2008) para o Cenário 1.....	60
5.2.3.2 Comparação com os resultados de Pepin <i>et al.</i> (2008) para o Cenário 2.....	62
5.2.3.3 Comparação com os resultados de Pepin <i>et al.</i> (2008) para o Cenário 3.....	63
5.2.3.4 Comparação com os resultados de Pepin <i>et al.</i> (2008) para o Cenário 4.....	64
5.2.3.5 Comparação com os resultados de Pepin <i>et al.</i> (2008) para o Cenário 5.....	65
5.2.3.6 Comparação com os resultados de Pepin <i>et al.</i> (2008) para o Cenário 6.....	66
5.2.4 Resumo das soluções iniciais e finais encontradas neste trabalho.....	67
5.2.5 Resumo dos custos finais e número de veículos utilizados pela heurística proposta quando comparados ao trabalho de Pepin et al. (2008).....	68
6 CONCLUSÕES.....	71
6.1 SÍNTESE DAS CONCLUSÕES.....	71
6.2 LIMITAÇÕES.....	72
6.3 SUGESTÕES PARA TRABALHOS FUTUROS	73
REFERÊNCIAS.....	74

1 INTRODUÇÃO

A Pesquisa Operacional (PO) é responsável pela estruturação de processos, através da proposição de alternativas de ação, prevendo e comparando valores de eficiência e de custos (LOESCH; HEIN, 1999). Gaither e Frazier (2002) consideram que a PO, à semelhança da administração científica, procura substituir a tomada de decisões intuitiva para grandes e complexos problemas por uma abordagem que identifique a alternativa ótima, ou a melhor, por meio da realização de análises.

Os modelos de PO são estruturados de forma lógica e amparados no ferramental matemático de representação, objetivando claramente a determinação das melhores condições de funcionamento para os sistemas representados. Os principais modelos de PO são denominados de Programação Matemática e constituem uma das mais importantes variedades dos modelos quantitativos (GOLDBARG, LUNA; 2005).

Por esta razão, a PO é conhecida como ciência e tecnologia de decisão, pois os componentes científicos estão ligados à ideias e processos para articulação de modelagem de problemas relacionados à tomada de decisões, determinando os objetivos do decisor (ARENALES *et al.*, 2007).

Atualmente os problemas logísticos estão se apoiando de forma bastante expressiva na PO a fim de obter uma maior eficiência em suas operações. Às questões relativas ao sequenciamento de produção, instalações de novas plantas, planejamento de distribuição, roteamento de veículos, entre outras, utilizam-se técnicas de PO com o intuito de proporcionar não apenas uma solução eficiente, mas, em muitos casos, uma solução ótima para essa variedade de problemas (BOFFEY, 1982).

Neste contexto, alguns problemas logísticos abordam a distribuição geográfica das garagens (ou depósitos), a quantidade de veículos disponíveis, o tempo de início e término das viagens e como elas se inter-relacionam na organização e planejamento das sequências ou rotas as quais devem estar relacionadas. A fim de não comprometer a otimização ou a tratabilidade do sistema, todos estes critérios devem ser levados conjuntamente em consideração na resolução do problema (ROHDE, 2008).

Dentre os vários problemas relacionados à designação de veículos em um sistema logístico, o de Escalonamento de Veículos com Múltiplas Garagens, MDVSP (*Multiple Depot Vehicle Scheduling Problem*), vem sendo abordado em variadas pesquisas. Este tipo de problema pressupõe a existência de diversas garagens que interferem no planejamento das sequências com as quais as viagens devem ser executadas, tal qual em sistemas reais de frotas de aeronaves ou caminhões de recolhimento de lixo urbano (PEPIN *et al.*, 2008).

Devido à maioria dos problemas de otimização combinatória ser da classe NP-*hard*, entre eles o MDVSP, frequentemente é inviável a aplicação computacional de tais modelos em problemas do mundo real. Para poder levar em consideração instâncias cada vez maiores ou mais próximas das reais, várias abordagens heurísticas estão sendo desenvolvidas.

Estas heurísticas variam desde os métodos de busca local aos métodos baseados em programação matemática e técnicas de decomposição, incluindo as meta-heurísticas. Desta forma, muitos dos modelos que representam esse problema são apresentados de modo genérico, o que dificulta sua implementação em um contexto real, haja vista a variedade de pormenores que estão por de trás de cada ambiente logístico (BERTOSI; CARRARES; GALLO, 1987).

Utilizada em vários trabalhos na área de otimização, a meta-heurística de busca-tabu foi escolhida para o desenvolvimento deste trabalho devido à sua grande adaptação a problemas de roteamento e escalonamento de veículos (GLOVER, 1989), além de ser referenciada e indicada por vários autores para tratar problemas desta natureza, como descrito no próximo capítulo.

A principal ideia desta meta-heurística consiste em avaliar a vizinhança da solução atual a cada iteração, sendo que a solução inicial pode ser construída aleatoriamente ou através de algum outro método. Uma heurística de busca local convencional pararia quando achasse uma solução atual melhor do que suas soluções vizinhas, no entanto, a busca-tabu aceita soluções de não-melhora como estratégia para ir além desse ótimo local.

Assim, a busca-tabu surgiu como uma técnica para guiar uma heurística de busca local tradicional na exploração do espaço de soluções além da

otimalidade local, usando para isso, basicamente, estruturas de memória. Esta técnica é uma das meta-heurísticas mais usadas e seu sucesso decorre de sua eficiência em produzir soluções de alta qualidade para vários problemas combinatórios, entre os quais: programação de horários (SANTOS et al.; 2005; SOUZA et al.; 2004), roteirização (ARCHETTI et al.; 2006; CORDEAU et al.; 2002; GENDREAU et al.; 1999, 1996) e sequenciamento (AL-LAHVERDI et al.; 2008).

É importante lembrar que o problema de Escalonamento (ou designação) de Veículos, seja ele simples ou com múltiplas garagens, consiste apenas na designação de veículos (que pertencem à determinada garagem K) às rotas que já se encontram configuradas em sua otimalidade (RIBEIRO, SOUMIS, 1994). Assim, fica claro que com este trabalho não se pretende otimizar rotas de veículos, mas apenas fazer uma designação às rotas já previamente otimizadas.

Em alguns sistemas reais, muitas vezes, os veículos saem e voltam a garagens distintas, tentando aproveitar ao máximo essa rede, diminuindo a complexidade de resolução. Na literatura, trabalhos que utilizam esta relaxação são comuns, incluindo o de Rohde (2008). Porém, neste trabalho a restrição de fluxo conservativo (posteriormente discutida) é mantida em sua forma original.

Portanto, o principal objetivo deste trabalho é **propor uma heurística para solucionar o MDVSP utilizando busca-tabu.**

Além disso, como objetivos secundários encontram-se (i) apresentar alguns dos métodos de solução para o MDVSP encontradas na literatura; (ii) desenvolver uma abordagem heurística para resolução do problema; (iii) implementar o modelo computacionalmente e, (iv) avaliar a qualidade da heurística e da solução encontrada para o problema.

2 CONTEXTO E JUSTIFICATIVA DO PROBLEMA

Um escalonamento de veículos considerado ótimo é aquele caracterizado por um tamanho de frota e custo operacional mínimos, incluindo os custos de movimentação com o veículo vazio, os tempos de espera, ociosidade e cancelamento de viagens. Além disso, no caso do MDVSP, pressupõe-se a existência de diversas garagens – que podem ser aeroportos, rodoviárias, estações de reciclagem de lixo, etc. –, uma vez que a maioria dos sistemas reais é assim configurada.

O desenvolvimento de modelos com múltiplas garagens deve-se ao fato prático de que poucos são os sistemas reais que utilizam uma única garagem. Em sistemas aéreos, por exemplo, cada aeroporto pode ser considerado como uma garagem, em um sistema rodoviário cada estação rodoviária pode ser considerada como uma garagem, como também em um sistema público de transporte cada ponto de reabastecimento de veículo pode ser considerado como uma garagem, etc. Não é difícil perceber que são poucos os exemplos de sistemas logísticos que utilizam uma única garagem, de modo que seu planejamento deve fazer uso de modelos capazes de representar essa realidade.

Não obstante a literatura sobre o problema de escalonamento de veículos, VSP (*Vehicle Scheduling Problem*), ser bastante abrangente, os trabalhos que abordam o MDVSP são mais escassos, sendo a maioria limitada a modelos que representam pequenas instâncias do problema trabalhando com instâncias inferiores a quinhentas viagens e quatro garagens (ESKIOGLU; VURAL; REISMAN, 2009).

Mesmo assim, recentemente alcançou-se algum progresso na resolução de problemas práticos do tipo MDVSP incluindo-se diversas instâncias, porém o conjunto atual de tecnologias ligadas à otimização limita o tamanho dos modelos que podem ser resolvidos (GINTNER; KLIEWER; SUHL, 2005).

Em adição a isto, a principal motivação para a realização deste projeto surgiu a partir de sugestões e limitações elencadas no estudo de Rohde (2008), o qual utilizou o algoritmo *branch-and-bound* em uma das etapas da heurística apresentada para resolver o MDVSP.

De acordo com Rohde (2008), utilizando-se o algoritmo *branch-and-bound*, a partir de dez iterações, aproximadamente, o número de restrições acrescentadas ao modelo original começa a impactar no tempo de solução, uma vez que este algoritmo sempre procura a solução ótima para o problema.

O autor acima desenvolveu uma heurística de redução de espaços de estados para, após, resolver o problema na otimalidade através do método de *branch-and-bound*. Porém, o autor sugere que abordagens heurísticas como, principalmente, a geração de colunas e a busca-tabu, configuram-se como mais apropriadas à resolução desse tipo de problema devido a suas características. Pepin *et al.*(2008) complementam dizendo que a meta-heurística de busca-tabu é capaz de encontrar soluções muito próximas às ótimas em um tempo computacional muito pequeno.

Dos dois métodos discutidos acima, a resolução via meta-heurística de busca-tabu é um método de solução mais apropriado do que o de geração de colunas quando se deseja encontrar uma solução de boa qualidade em um tempo computacional melhor (PEPIN *et al.* 2008).

Além disso, na literatura existente, apenas no estudo de Pepin *et al.* (2008) foram encontrados desenvolvimentos utilizando a meta-heurística de busca-tabu para a resolução do MDVSP. Os próprios autores reconhecem e desconhecem a inexistência de outros trabalhos abordando este tipo de resolução.

Até os anos 1990 a preocupação dos pesquisadores era resolver o MDVSP por procedimentos heurísticos e, após, a atenção foi voltada para resolução via métodos exatos como geração de colunas e *branch-and-bound*. Recentemente é que abordagens envolvendo as meta-heurísticas estão sendo avaliadas para resolver este tipo de problema (PEPIN *et al.* 2008).

Assim, o desenvolvimento deste trabalho também acaba contribuindo para o esclarecimento de que é possível que uma abordagem meta-heurística baseada em busca-tabu apresente resultados igualmente competitivos quando se comparado a outras heurísticas ou meta-heurísticas de mais difícil construção, principalmente nos aspectos que tangenciam uso de memória computacional, velocidade e qualidade de resolução.

Em vista dos custos envolvidos, o MDVSP exige uma solução considerando sua totalidade, uma vez que a solução individual do modelo

considerando cada garagem isoladamente não conduz a otimalidade do sistema. Em outros termos, é necessário resolver o problema considerando todas as garagens simultaneamente em um modelo único. Essa peculiaridade torna o MDVSP um modelo NP difícil, ou seja, matematicamente ainda não foi provada a possibilidade de encontrar uma solução em tempo polinomial para o problema. Assim, do ponto de vista teórico é possível encontrar soluções que planejem todo o sistema, contudo no que tange os aspectos computacionais, tais soluções podem ser impraticáveis por apresentarem tempo de solução exponencial. (ROHDE, 2008).

Deste modo, deve-se identificar e separar o essencial do dispensável na criação de sistemas, permitindo simplificar a realidade focando os objetivos mais importantes (SIMON, 1989).

Todas as razões apresentadas explicam a escassez de modelos que conseguem alcançar uma efetiva aplicabilidade em problemas do mundo real. Diversas restrições e peculiaridades de sistemas logísticos são ignoradas deixando margem para futuras pesquisas na área. Deste modo, poucos trabalhos abordam o MDVSP de modo a conseguir solucionar o problema para uma grande quantidade de viagens e garagens.

Outra justificativa para o desenvolvimento desse estudo deve-se à peculiaridade dos problemas VSP, VRSP e MDVSP estarem associados diretamente à solução de problemas logísticos e de áreas afins. Atualmente diversos pesquisadores investigam algoritmos eficientes para solução destes modelos. Dada a alta complexidade envolvida em algumas instâncias destes problemas, as pesquisas de heurísticas e meta-heurísticas também foram, e ainda são, muito comuns (LI *et al.*, 2004).

Apesar da ampla literatura, os algoritmos e modelos de VSP, infelizmente, não podem ser utilizados diretamente no tratamento de problemas MDVSP, face às peculiaridades do último, não contempladas pelo primeiro. É justamente por essas características que o desenvolvimento de modelos apropriados de MDVSP se justifica. (ROHDE, 2008).

Por fim, um modelo bem estruturado e com soluções que representem a realidade de um MDVSP, é capaz de trazer não apenas uma redução dos custos e do tamanho da frota, mas também uma melhor qualidade de

atendimento ao cliente e, por consequência, uma habilidade maior de se adequar às contingências.

3 REVISÃO DE LITERATURA

Neste capítulo serão abordados aspectos teóricos relativos aos problemas logísticos e como a pesquisa operacional pode auxiliar na tomada de decisões relativas a este contexto. Logo após, serão apresentados os problemas de designação de veículos em um sistema logístico, bem como o problema de escalonamento de veículos (VSP), o problema de escalonamento de veículos com múltiplas garagens (MDVSP) e suas formulações mais utilizadas, bem como algumas questões adicionais sobre o MDVSP. Por fim, uma explicação sobre o que consiste a meta-heurística de busca-tabu também é apresentada.

3.1 PROBLEMAS LOGÍSTICOS E O EMPREGO DA PESQUISA OPERACIONAL COMO FERRAMENTA DE APOIO À TOMADA DE DECISÃO

A partir do início da década de 1950 até o final da década de 1960, a Pesquisa Operacional foi aplicada em uma variedade de problemas oriundos dos setores público e privado. Exemplos de aplicações envolviam diversos setores industriais e financeiros, como: mineração, metalúrgico, construção civil e militar, têxtil, farmacêutico, bancário e transportes (ARENALES *et al.* 2007).

Em 1967, foram identificados 766 grupos de PO, dos quais 553 estavam envolvidos no setor industrial ou em serviços financeiros. Exemplos de aplicações no setor público envolviam coleta de lixo, transporte, polícia, entre outros. Desde então, a pesquisa operacional tem sido aplicada às mais diversas áreas de produção e logística, incluindo indústrias de alimentação, automóveis, aviação, computadores, eletrônica, metalurgia, móveis, papel, petróleo, transportes; além de organizações de serviços (ARENALES *et al.* 2007).

Para Goldberg e Luna (2005), as organizações são fenômenos em que o esforço por reduzir a variabilidade de respostas é uma tarefa crítica e, por isso, demanda substancial atenção e esforço. A programação matemática, por sua

vez, é fortemente direcionada ao apoio da tomada de decisão no gerenciamento de sistemas de grande porte, especialmente no que tange o tratamento de variáveis quantificadas. A técnica permite a modelagem de inter-relações entre variáveis que dificilmente seriam visíveis de forma intuitiva.

Com a utilização dos meios de processamento automático de dados, os chamados programas de programação matemática podem examinar inúmeras configurações viáveis do problema proposto pelo tomador de decisão e selecionar, dentro vários critérios (ou restrições), as “melhores” (GOLDBARG, LUNA; 2005).

3.2 PROBLEMAS DE DESIGNAÇÃO DE VEÍCULOS EM UM SISTEMA LOGÍSTICO

O atual interesse na logística de transportes, a contínua busca pela redução dos custos de produção e os incrementos cada vez mais frequentes na tecnologia da computação faz com que muitos dos pesquisadores operacionais foquem seus interesses nos problemas de roteamento e escalonamento de veículos (GOLDBARG; LUNA, 2005).

Esses problemas geralmente envolvem a designação de veículos a viagens, de forma que os custos de designação e da rota correspondente sejam minimizados. Os aspectos relacionados ao tempo de duração dessas viagens tornam-se muito importantes em contextos industriais, de serviços e das empresas de transporte, que tentam não apenas diminuir ou acabar com alguns de seus custos logísticos, mas também competir com serviços diferenciados (DESROSIERS *et al.* 1995).

Vários são os problemas que trabalham com a designação de veículos em um sistema logístico. Os mais tradicionais são o VRP (*Vehicle Routing Problem*, ou Problema de Roteamento de Veículos) e sua variação, o VRRP (*Vehicle Rerouting Problem*, ou Problema de Re-Roteamento de Veículos) e também o MDVSP (*Multiple Depot Vehicle Scheduling Problem*, ou Problema de Escalonamento de Veículos com Múltiplas Garagens) e sua variação, o MDVRSP (*Multiple Depot Vehicle Rescheduling Problem*, ou Problema de Re-Escalonamento de Veículos com Múltiplas Garagens) (DESROSIERS *et al.*, 1995, ESKIOGLU; VURAL; REISMAN, 2009).

De forma geral, os problemas de roteamento de veículos buscam traçar as melhores rotas que vários veículos devem executar de modo a reduzir os custos operacionais, enquanto os problemas de escalonamento de veículos buscam identificar qual veículo deve executar determinado conjunto de viagens pré-estabelecidas, a fim de minimizar os custos de ociosidade ou espera entre viagens, além de visar diminuir a quantidade de veículos e garagens utilizadas (ROHDE, 2008).

Estes tipos de problemas são, geralmente, de muito difícil resolução e pertencem à classe de problemas conhecida como *NP-hard*. Isto quer dizer que o tempo computacional para resolução aumenta exponencialmente de acordo com o tamanho do problema. Na maioria das vezes estes problemas somente podem ser resolvidos utilizando heurísticas (HAGHANI; BANIHASHEMI; CHIANG, 2003).

3.3 O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS (VSP)

O Problema de Escalonamento de Veículos (VSP) é um clássico problema de otimização utilizado para o planejamento operacional de sistemas de transporte público, por exemplo. O VSP consiste em indicar um conjunto de viagens escalonadas a um conjunto disponível de veículos, de forma que cada viagem seja associada a um veículo e que o custo dessa designação seja minimizado (BAITA *et al.*, 2000).

Para várias versões deste problema, como quando todos os veículos são iguais e dividem a mesma garagem, existem muitos algoritmos eficientes. Porém, aplicações deste problema no sistema real podem se tornar mais complexas não somente pela dimensão do problema, mas também pelos requisitos particulares presentes em situações reais e que são de difícil modelagem. Aplicações práticas para o VSP incluem critérios de avaliação de performances - produzindo diferentes soluções alternativas -, dando sugestões de como os dados podem ser modificados para alcançar a qualidade na solução (BAITA *et al.*, 2000).

A definição do VSP, de acordo com Desrosiers *et al.* (1995) consiste em descobrir um conjunto de n viagens, $\{T_1, T_2, \dots, T_n\}$, onde a viagem T_i possui um certa duração e começa no tempo a_i , onde $i = 1, 2, \dots, n$. Deve-se considerar uma única garagem onde v veículos estão estacionados e um conjunto de nós $N = \{1, 2, \dots, n\}$, representando o conjunto de viagens e o nó $\{n+1\}$, representando a única garagem. Uma vez t_{ij} denotando o tempo de viagem entre o ponto final da viagem T_i e o ponto inicial da viagem T_j , por simplicidade assume-se que t_{ij} inclui a duração da viagem T_i . Um conjunto de par ordenado de viagens (T_i, T_j) é compatível se satisfaz a relação $a_i + t_{ij} \leq a_j$, para todo $i, j \in N$. Se duas viagens são cobertas pelo mesmo veículo, restrições adicionais devem ser impostas para tornar o problema compatível.

Assim, o problema consiste em designar um conjunto de veículos a viagens de tal forma que: a) cada viagem seja coberta exatamente por um veículo; b) cada veículo utilizado na solução do problema saia da garagem, realize sua função e, no final, retorne à garagem; c) o número de veículos que deixa a garagem não pode exceder v (esse número pode ser fixo, minimizado durante o processo ou livre); e, d) a soma dos custos de viagem das rotas sejam minimizadas.

3.4 FORMULAÇÕES PARA O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS (VSP)

No VSP, uma sequência de viagens é possível se cada par de viagens consecutivas é compatível, levando-se em consideração as restrições discutidas acima. O modelo apresentado a seguir para o VSP foi extraído do trabalho de Freling, Wagelmans e Paixão (2001), e se caracteriza como um dos mais utilizados para modelagem deste tipo de problema.

Primeiramente, deve-se considerar $N = \{1, 2, \dots, n\}$ como o conjunto de viagens numeradas de acordo com os tempos de início, e $E = \{(i, j) \mid i < j, i \in N, j \in N\}$ como sendo o conjunto de arcos correspondentes às viagens. Os nós s e t representam a garagem no local d , inseridos na rede de escalonamento de

veículos $G = (V, A)$ (uma rede não-cíclica e direcionada) com nós $V = N \cup \{s, t\}$ e arcos $A = E \cup (s \times N) \cup (N \times t)$.

Um caminho de s para t nesta rede representa um escalonamento viável para um veículo. Diz-se que um escalonamento viável completo incide quando um conjunto de caminhos disjuntos de s para t ocorre, de tal forma que cada nó em N é coberto.

Uma vez c_{ij} sendo o custo do veículo no arco $(i, j) \in A$, o qual é frequentemente alguma função da viagem e do tempo ocioso, um custo fixo K relativo ao uso do veículo pode ser adicionado ao custo dos arcos (s, i) ou (j, t) para todo $i, j \in N$. O principal objetivo deste modelo é minimizar o número de veículos, ou seja, assegurar que todo K é suficiente para garantir que este número mínimo será utilizado (1). Utilizando-se variáveis de decisão Y_{ij} , com $Y_{ij} = 1$ se um veículo cobre a viagem j diretamente após a viagem i e $Y_{ij} = 0$ caso contrário, o VSP pode ser formulado como segue:

$$\text{Minimizar } \sum_{(i,j) \in A} c_{ij} Y_{ij} \quad (1)$$

Sujeito a:

$$\sum_{j:(i,j) \in A} Y_{ij} = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{i:(i,j) \in A} Y_{ij} = 1 \quad \forall j \in N \quad (3)$$

$$Y_{ij} \in \{0, 1\} \quad \forall ij \in A \quad (4)$$

Neste modelo, as restrições (2) e (3) asseguram que cada viagem é designada para exatamente um predecessor e um sucessor, ou seja, garantem que a rede seja particionada em um conjunto de caminhos disjuntos de s para t . Devido às restrições (2) e (3) definir uma matriz de restrição unimodular, condições binárias nas variáveis são relaxadas para $Y_{ij} \geq 0$.

3.5 O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS COM MÚLTIPLAS GARAGENS (MDVSP)

O problema de escalonamento de veículos com múltiplas garagens (MDVSP) é um problema bastante conhecido e que possui aplicações em vários campos, como no transporte público e privado de passageiros e em sistemas de entrega e coleta de mercadorias através de caminhões e aeronaves (PEPIN *et al.*, 2008).

A definição do problema consiste em encontrar um conjunto T de viagens (ou *tasks*) com o tempo já determinado e uma frota de veículos pertencente a um conjunto K de garagens que seja capaz de apresentar os menores custos possíveis de designação dos veículos, de modo que cada viagem seja realizada por um veículo apenas uma vez e que o número v_k de veículos disponíveis em cada garagem $k \in K$ não seja excedido.

Cada viagem $i \in T$ é definida por uma localização inicial s_i , uma localização final e_i (que pode ser a mesma de s_i), um tempo inicial a_i e uma duração δ_i que inclua o tempo de viagem entre s_i e e_i . A designação de um veículo deve iniciar e finalizar na mesma garagem e é composta por uma sequência ordenada de viagens. Isto é possível se, para cada par i e j de viagens consecutivas desta designação, a relação $a_i + \delta_i + t_{ij} \leq a_j$ for mantida, onde t_{ij} é o tempo da viagem entre os locais e_i e s_j .

O custo de uma viagem (*task*) para um veículo na garagem k é dada pela soma dos custos de viagem e de espera (*wait costs*) incorridos entre as viagens consecutivas (*deadheads trips*), dos custos incorridos do momento em que o veículo deixa a garagem até iniciar sua tarefa (*pull-out trips*), bem como os custos envolvidos a partir de quando o veículo acaba sua tarefa até chegar de volta à garagem (*pull-in trips*). A estes custos também podem ser acrescentados os custos fixos do veículo. Restrições limitando a qual conjunto de garagens os veículos devem pertencer para executar determinada viagem podem ser acrescentadas ao modelo. Estas restrições simplificam o problema, reduzindo o número de soluções possíveis (PEPIN *et al.*, 2008). A Figura 1 apresenta a configuração de uma rede exemplificando cada um desses custos envolvidos no problema de escalonamento de veículos com múltiplas garagens e seus

respectivos caminhos, considerando-se $o(k)$ como o ponto de saída de determinada garagem k ; e $d(k)$ o ponto de retorno à garagem k .

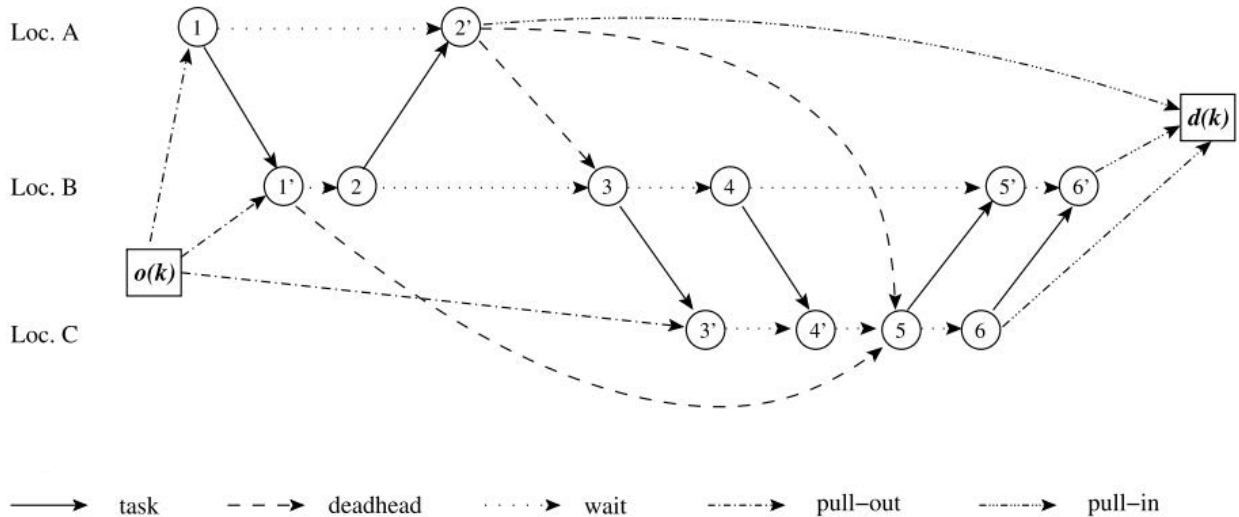


Figura 1: Exemplo de Rede para o MDVSP

Fonte: Pepin *et al.* (2008).

Apesar de métodos exatos serem capazes de resolver certas classes de instâncias do MDVSP no sistema real, existem diversas razões que levam ao desenvolvimento de heurísticas para o MDVSP e à comparação de performances de algumas heurísticas.

Primeiramente, os MDVSP precisam ser resolvidos em pequeno tempo computacional em algumas situações. Isto é exigido, por exemplo, quando uma solução planejada precisa ser re-otimizada em um contexto operacional ou quando o MDVSP aparece como um subproblema em problemas mais complexos. Neste último caso, métodos de decomposição como o proposto por Huisman *et al.* (2004), para o escalonamento de veículos com múltiplas garagens em associação ao escalonamento de tripulação, resolvem o subproblema diversas vezes durante o processo geral de resolução. Além disso, boas heurísticas também podem ajudar a resolver este tipo complexo do problema (HADJAR; MARCOTTE; SOUMIS, 2006).

Em segundo lugar, vários softwares utilizados por empresas de transporte público e de entregas são capazes de resolver o MDVSP. Porém, na

maioria das vezes, heurísticas são usadas para encontrar soluções mais rapidamente (HADJAR; MARCOTTE; SOUMIS, 2006).

O MDVSP pode ser formulado como um problema de programação inteira. Contudo, devido ao grande número de restrições e variáveis, a aplicação em problemas reais faz com que os modelos de programação inteira não possam encontrar uma solução exata. Sendo assim, procedimentos heurísticos são necessários para encontrar uma solução aceitável para estes problemas. De modo geral os objetivos de otimização do MDVSP são os mesmo do SDVSP. Entretanto adiciona-se a estes problemas a possibilidade de minimizar-se os custos de investimentos com garagens, ou seja, revendo a necessidade do atual número de garagens que compõe o sistema (ROHDE, 2008).

3.6 FORMULAÇÕES PARA O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS COM MÚLTIPLAS GARAGENS (MDVSP)

A literatura demonstra que o problema de escalonamento de veículos com múltiplas garagens pode ser formulado de duas formas: como uma formulação inteira de um fluxo com *multi-commodity* e como uma formulação particionada em conjuntos (*set partitioning type*). A primeira é a base para heurísticas como *branch-and-cut* e relaxamento lagrangeano, enquanto a segunda é mais utilizada para a heurística de Geração de Colunas (PEPIN *et al.* 2008).

Ribeiro e Soumis (1994) formularam o MDVSP como um modelo inteiro de fluxo de rede de *multi-commodity*, onde uma *commodity* é definida para cada garagem em K . Os mesmos autores também formularam o MDVSP como um modelo particionado em conjuntos, derivado do modelo de fluxo de rede, através da decomposição de Dantzig-Wolfe.

A seguir apresenta-se a formulação para o MDVSP de acordo com o modelo de fluxo de rede proposto por Ribeiro e Soumis (1994). Esta formulação ainda é uma das mais utilizadas em trabalhos recentes.

De acordo com Ribeiro e Soumis (1994), deve-se considerar a rede $G_k = (V_k, A_k)$ para cada garagem $k \in K$, onde V_k e A_k denotam os conjuntos de nós e de arcos, respectivamente. O conjunto V_k contém um nó para cada viagem $i \in T$ e um par de nós, $o(k)$ e $d(k)$, que representam o início e o fim de uma designação do veículo associado à garagem k , respectivamente. Assim, $V_k = \{o(k), d(k)\} \cup T$. O conjunto A_k contém três tipos de arcos: *pull-out*, *pull-in* e arcos de conexão. Existe um arco *pull-out* $(o(k), i)$ para cada viagem do nó $i \in T$. Simetricamente, existe um arco *pull-in* $(i, d(k))$ para cada viagem do nó $i \in T$. Finalmente, existe um arco de conexão (i, j) para cada par de nós de viagens, i e j em T , de tal forma que $a_i + \sigma_i + t_{ij} \leq a_j$. O custo de um arco $(i, j) \in A_k$, ou simplesmente c_{ij} , é igual aos custos de viagem e de espera associados a ele. Se $i = o(k)$ e um custo fixo deve ser pago para cada veículo utilizado, c_{ij} também inclui este custo.

A formulação proposta envolve as variáveis binárias X_{ij}^k , $(i, j) \in A^k$, $k \in K$. Esta variável é igual ao fluxo da *commodity* k no arco (i, j) . Utilizando esta notação, o MDVSP pode ser modelado como:

$$\text{Minimizar } \sum_{k \in K} \sum_{(i,j) \in A^k} c_{ij} X_{ij}^k \quad (5)$$

Sujeito a:

$$\sum_{k \in K} \sum_{j: (i,j) \in A^k} X_{ij}^k = 1, \quad \forall i \in T \quad (6)$$

$$\sum_{j: (o(k),j) \in A^k} X_{o(k),j}^k \leq v_k \quad \forall k \in K \quad (7)$$

$$\sum_{j: (j,i) \in A^k} X_{ji}^k - \sum_{j: (i,j) \in A^k} X_{ij}^k = 0 \quad \forall i \in V^k \setminus \{o(k), d(k)\}, k \in K \quad (8)$$

$$X_{ij}^k \in \{0, 1\}, \forall (i, j) \in A^k, k \in K. \quad (9)$$

Neste modelo, a função (5) minimiza o total dos custos, enquanto a restrição (6) assegura que cada viagem seja executada apenas uma vez por

veículo. Por sua vez, a restrição (7) limita o número de veículos que pode ser usado por garagem e a restrição (8) é a de fluxo conservativo, a qual define estruturas de caminhos múltiplos para cada garagem. Por fim, os requisitos para as variáveis binárias são encontrados em (9).

Já para a formulação particionada em conjuntos, Ribeiro e Soumis (1994) consideram Ω^k o conjunto de todos os escalonamentos possíveis para a garagem $k \in K$. Para cada designação $p \in \Omega^k$, deve-se definir a variável de custo c_p e, para cada viagem $i \in T$, um parâmetro binário a_{ip} igual a 1 se o escalonamento inclui a viagem i e 0 se não. Além disso, com cada escalonamento $p \in \Omega^k$, deve-se definir a variável binária θ_p que assume valor 1 se p é encontrado na solução e 0 caso contrário.

Assim, o MDVSP pode ser modelado como:

$$\text{Minimizar } \sum_{k \in K} \sum_{p \in \Omega^k} c_p \theta_p \quad (10)$$

Sujeito a:

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip} \theta_p = 1, \quad \forall i \in T \quad (11)$$

$$\sum_{p \in \Omega^k} \theta_p \leq v_k \quad \forall k \in K \quad (12)$$

$$\theta_p \in \{0,1\}, \quad \forall p \in \Omega^k, k \in K. \quad (13)$$

A função objetivo (10) minimiza os custos totais. A restrição de particionamento de conjuntos (11) impõe que cada viagem seja feita por apenas um veículo, enquanto a restrição de desigualdade (12) expressa a capacidade por garagem. A restrição binária (13) assume valor 1 se p é encontrado na solução e 0 caso contrário.

3.7 BUSCA-TABU (BT)

A principal ideia desta meta-heurística consiste em avaliar a vizinhança da solução atual a cada iteração, sendo que a solução inicial pode ser construída aleatoriamente ou através de algum outro método. Uma heurística de busca local convencional pararia quando achasse uma solução atual melhor do que suas soluções vizinhas, no entanto, a busca-tabu aceita soluções de não-melhora como estratégia para ir além desse ótimo local (PEPIN *et al.* 2008).

Assim, a busca-tabu surgiu como uma técnica para guiar uma heurística de busca local tradicional na exploração do espaço de soluções além da otimalidade local, usando para isso, basicamente, estruturas de memória. Esta técnica é uma das meta-heurísticas mais usadas e seu sucesso decorre de sua eficiência em produzir soluções de alta qualidade para vários problemas combinatórios (ARENALES *et al.* 2007).

O conceito da busca-tabu consiste em derivar e explorar os princípios da resolução inteligente de problemas, tendo como elemento fundamental subjacente o uso de uma memória flexível que, do ponto de vista do método, incorpora os processos de criação e exploração de estruturas, obtendo vantagens do histórico de iterações (LAGUNA, 1994).

Goldberg e Luna (2005) destacam que seus procedimentos de busca são enriquecidos por estratégias de memória que buscam evitar que configurações já examinadas sejam revisitadas durante o desenvolvimento do algoritmo.

Zäpfel, Braune e Bögl (2010) consideram a lista tabu, representada por uma memória de curto prazo, como elemento central da BT. Assim, a busca-tabu explora o espaço de solução de um problema movendo, a cada iteração, a solução atual para o seu melhor vizinho e, para evitar ciclos, alguns atributos da solução atual são armazenados em uma lista e qualquer solução que possua os mesmos atributos é declarada como proibida (ou tabu) para um determinado número de iterações (GENDREAU *et al.*, 1999).

O procedimento de busca-tabu explora o espaço de soluções movendo cada iteração de soluções para uma maior, considerando-se um conjunto de

vizinhança. A solução corrente pode decrementar de uma iteração para a próxima e novas soluções com resultados piores são aceitas somente para inibir caminhos já investigados. Isto assegura que novas regiões do espaço de soluções do problema sejam investigadas com o objetivo de evitar achar-se um mínimo local e, de forma definitiva, a solução desejada.

A fim de evitar a ciclagem, soluções que apresentem alguns atributos de algumas soluções recentemente exploradas são, temporariamente, declaradas como tabu, ou proibidas. A duração que o um atributo permanece como tabu varia de acordo com o tamanho das listas tabu e com a quantidade de movimentos, e pode variar através de diferentes intervalos de tempo.

A busca-tabu pode notada da seguinte forma: para uma solução $s \in S$, diz-se que $N(s)$ denota a vizinhança de s através de procedimentos de troca local, chamados movimentos. As técnicas de busca local visitam uma sequência s_0, \dots, s_t de soluções, onde s_0 é a solução inicial e $s_{i+1} \in N(s_i) (i = 1, \dots, t-1)$. A seguir, é apresentado um esquema geral do algoritmo de Busca-Tabu (Quadro 1), onde LT (Lista-Tabu) é a lista de movimentos proibidos. Maiores detalhes da descrição do método e de como foi concebido pode ser encontrado em Glover e Laguna (1997).

Quadro 01: Pseudo-código para o algoritmo de Busca-Tabu

PSEUDO-CÓDIGO PARA O ALGORÍTMO DE BUSCA-TABU
1. Gerar uma solução inicial viável s ;
2. Configure $LT \leftarrow \emptyset$ e $s^* \leftarrow s$;
3. enquanto nenhum critério de parada, faça
4. Determine a solução $s' \in N(s)$ com o menor valor $f(s')$ de forma que tanto s' seja obtido de s através de um movimento $m \notin LT$ ou s' seja viável e $f(s') < f(s^*)$
5. se s' é viável e $f(s) < f(s^*)$, então
6. Configure $s^* \leftarrow s'$
7. e se
8. Configure $s \leftarrow s'$ e atualize LT
9. Fim enquanto se manter esta situação.

Fonte: adaptado de Cordeau *et al.* (2001).

4 MÉTODO DE PESQUISA

Esta pesquisa se apoia nos preceitos metodológicos da Pesquisa Operacional propostos por Andrade (1998), que considera a PO, por si só, como um método de pesquisa que agrega, em sua teoria, quatro ciências fundamentais para o processo decisório: economia, matemática, estatística e computação.

A fim de seguir não apenas o processo metodológico proposto por Andrade (1998) resolveu-se utilizar algumas das etapas da pesquisa operacional também propostas por Wagner (1986) e Winston (1994) de forma a abranger de uma forma mais clara todas as etapas desta pesquisa. Desta forma, adaptou-se um desenho de pesquisa (Fig. 2) no qual foram consideradas quatro etapas distintas. Cada uma delas está relacionada diretamente com um dos objetivos secundários deste trabalho, contribuindo para que se alcance o objetivo principal de solucionar o MDVSP, já discutido na seção 3.5 deste trabalho, utilizando uma meta-heurística baseada no método de busca-tabu:

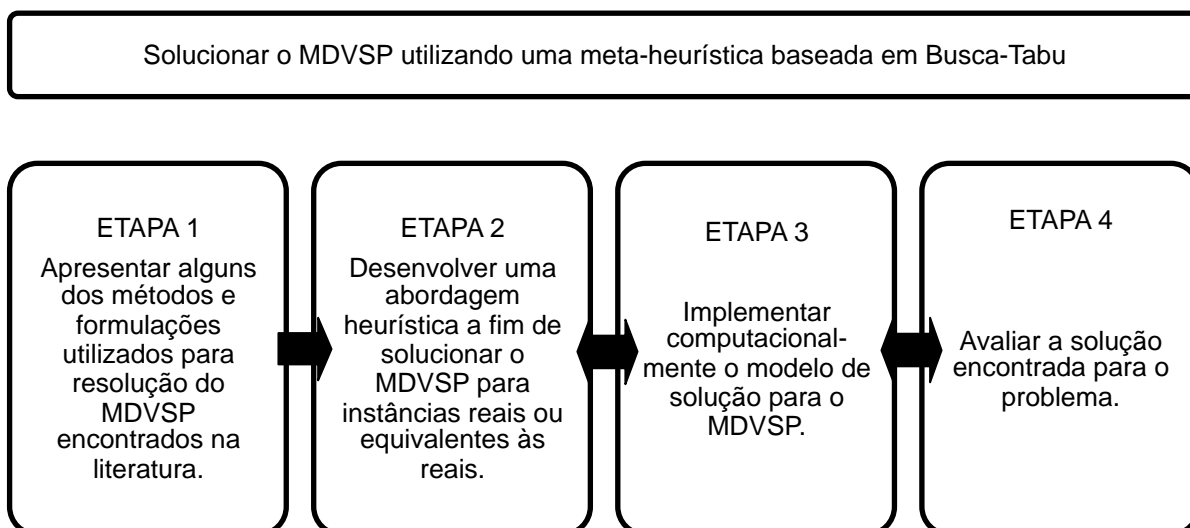


Fig. 2 - Desenho de Pesquisa.

Fonte: Adaptado de Andrade (1998), Wagner (1986) e Winston (1994).

É importante observar que as etapas dois, três e quatro podem ser realizadas concomitantemente ou, quando necessário, deve-se voltar à etapa

anterior para se fazer ajustes necessários a fim de encontrar uma solução viável para o problema proposto.

4.1 ETAPA 1: FORMULAÇÕES E MÉTODOS PARA RESOLUÇÃO DO MDVSP

Na primeira etapa da pesquisa foram apresentados alguns dos métodos de solução para o problema de escalonamento de veículos com múltiplas garagens (MDVSP) encontrados na literatura. Além desses métodos, também foram discutidos os problemas que deram origem ao MDVSP, como o problema de escalonamento de veículos com uma única garagem e os problemas de roteamento de veículos. Esta etapa, portanto, caracterizou-se como de revisão bibliográfica.

Agora, serão abordadas as formulações mais utilizadas para a modelagem deste tipo específico de problemas de escalonamento. As análises das formulações utilizadas bem como dos métodos mais eficientes e carentes na literatura, através desta etapa do trabalho, permitiram a escolha do método de resolução do problema que será utilizado neste trabalho, além de guiar a o referencial teórico necessário.

Ambas as formulações para o MDVSP apresentadas no capítulo anterior (fluxo de *multi-commodity* e de particionamento de conjuntos) são utilizadas atualmente. Esta última pode ser apresentada a partir da aplicação da Decomposição de Dantzig-Wolfe na primeira (LÖBEL, 1997).

Porém, nos casos referenciados na literatura que procuram resolver o problema de escalonamento de veículos com múltiplas garagens várias são as abordagens de solução utilizadas, encontrando-se o uso não somente de uma ou outra formulação, mas também de uma séria de heurísticas e raras meta-heurísticas.

Partindo deste pressuposto, o Quadro 2 apresenta alguns dos principais trabalhos encontrados na literatura alusivos ao MDVSP. Nele pode-se observar, o método e formulação utilizados para resolução do problema nos artigos analisados.

Quadro 2 - Principais artigos que abordam o MDVSP - Método de solução e formulação utilizada.

Publicação	Método Utilizado para Resolução	Formulação Utilizada
Bodin <i>et al.</i> (1983)	Heurística de Duas Fases: SDVSP e Problema de Transporte	SVSP e Problema de Transportes.
Bertossi, Carraresi e Gallo (1987)	Relaxamento Lagrangeano	Fluxo de <i>Multi-commodity</i> .
Carpaneto <i>et al.</i> (1989)	<i>Branch-and-Bound</i>	Fluxo de <i>Multi-commodity</i> .
Lamatsch (1990, 1992)	Relaxamento Lagrangeano	Fluxo de <i>Multi-commodity</i> .
Mesquita e Paixão (1992)	Relaxamento Lagrangeano	Fluxo de <i>Multi-commodity</i> .
Bianco, Mingozzi e Ricciardelli (1994)	<i>Branch-and-Bound</i> e Soluções Duais	Formulação de Conjuntos Particionados
Forbes <i>et al.</i> (1994)	Simplex Dual	Fluxo de <i>Multi-commodity</i> .
Ribeiro e Soumis (1994)	Geração de Colunas para o limite inferior, <i>Branch-and-Cut</i>	Formulação de Conjuntos Particionados
Desrosiers <i>et al.</i> (1995)	Geração de Colunas	Formulação de Conjuntos Particionados
Barnhart <i>et al.</i> (1996, 1997)	Geração de Colunas	Formulação de Conjuntos Particionados
Löbel (1997, 1998)	<i>Branch-and-Cut</i> / Geração de Colunas / <i>Lagrangian Pricing</i>	Fluxo de <i>Multi-commodity</i> .
Soumis (1997)	Geração de Colunas	Formulação de Conjuntos Particionados
Löbel (1999)	Geração de Colunas e Relaxamento Lagrangeano	Fluxo de <i>Multi-commodity</i> .
Mesquita e Paixão (1999)	Geração de Colunas e Relaxamento Lagrangeano	Fluxo de <i>Multi-commodity</i> .
Hadjar, Marcotte e Soumis (2006)	<i>Branch-and-Bound</i> / Geração de Colunas	Fluxo de <i>Multi-commodity</i> .
Pepin <i>et al.</i> (2008)	<i>Branch-and-Cut</i> / Geração de Colunas / Relaxamento Lagrangeano / Busca-Tabu / <i>Large Neighborhood Search</i>	Fluxo de <i>Multi-commodity</i>

Fonte: do Autor.

Os trabalhos envolvendo o problema de escalonamento de veículos com múltiplas garagens inicialmente utilizavam um método para resolução através

de uma heurística de duas fases: primeiro resolvendo um problema de escalonamento de veículos com uma única garagem e, após, fazendo a designação dos veículos através do uso da formulação tradicional de um problema de transportes, como no exemplo de Bodin *et al.* (1983).

Logo após, através da formulação do MDVSP como um fluxo de *multi-commodity*, diversos autores como Bertossi, Carraresi e Gallo (1987), Carpaneto *et al.*(1989), Lamatsch (1990, 1992) e Mesquita e Paixão (1992) conseguiram encontrar soluções para o problema através do método *branch-and-bound* e do uso do relaxamento lagrangeano.

Na década de 90, Bianco, Mingozzi e Ricciardelli (1994) e Ribeiro e Soumis (1994) desenvolveram para o MDVSP formulações baseadas no particionamento de conjuntos através da decomposição de Dantzig Wolfe, ou seja, um conjunto X é particionado em subconjuntos não-vazios de tal forma que cada elemento $x \in X$ seja exatamente um desses subconjuntos.

Ribeiro e Soumis (1994) foram um dos primeiros autores a resolver o problema de escalonamento de veículos com múltiplas garagens através do método de geração de colunas. Estes autores utilizaram este método a fim de gerar limites inferiores à resolução do problema, aplicando, após, um algoritmo *branch-and-cut*.

Outros autores também começaram a aplicar aos seus problemas o método de geração de colunas, combinando-o com diversas outras heurísticas, como o próprio *branch-and-cut* e relaxamento lagrangeano (DESROSIERS *et al.* 1995; BARNHART *et al.* 1996; 1997).

Löbel (1997, 1998) criou uma heurística própria, muito eficiente, que combina o uso de geração de colunas, *branch-and-cut*, e relaxamento lagrangeano, conhecida como *Lagrangian Pricing*. Muitos trabalhos (LÖBEL 1999, MESQUITA; PAIXÃO, 1999, HADJAR; MARCOTTE; SOUMIS, 2006) utilizam para a resolução do MDVSP o método de geração de colunas juntamente com o relaxamento lagrangeano.

Entre as últimas pesquisas que abordam o problema de escalonamento de veículos com múltiplas garagens encontra-se o trabalho de Pepin *et al.* (2008), o qual resolve o MDVSP utilizando cinco abordagens heurísticas distintas, a saber: i) *Branch-and-Cut*; ii) Geração de Colunas; iii) Relaxamento Lagrangeano; iv) Busca-Tabu; e, v) *Large Neighborhood Search*.

Neste trabalho, a primeira e a terceira heurística foram desenvolvidas através de adaptações dos métodos exatos propostos por Kliewer *et al.* (2006) e Ribeiro e Soumis (1994). A segunda heurística é uma variação da abordagem heurística desenvolvida por Lamatsch (1992), enquanto as duas outras meta-heurísticas (Busca-Tabu e *Large Neighborhood Search*) foram apresentadas no estudo de Pepin *et al.* (2008) sob caráter de ineditismo para resolução do Problema de Escalonamento de Veículos com Múltiplas Garagens.

É importante observar que de todos os trabalhos de maior expressão acadêmico, somente este último utiliza algumas meta-heurísticas, além de comparar seus resultados, tempos computacionais e uso de memória com outros trabalhos utilizando a mesma instância de muitos deles, geradas por Carpaneto *et al.* (1989).

A formulação do MDVSP como fluxo de *multi-commodity* é a dominante nas pesquisas atuais, apesar de muitos autores utilizarem a formulação de conjuntos particionados após uma decomposição da formulação inicial, uma vez que é mais aplicável ao uso de métodos que utilizem etapas de decomposição em sua resolução.

O modelo utilizado para resolução deste trabalho baseia-se na formulação do MDVSP proposta por Ribeiro e Soumis (1994), previamente apresentada no capítulo anterior. Por esta razão, optou-se apenas por apresentar novamente o modelo nesta seção, uma vez que já são de conhecimento do leitor sua estrutura e notação.

Assim, neste modelo, a função (14) minimiza o total dos custos, enquanto a restrição (15) assegura que cada viagem seja executada apenas uma vez por veículo. Por sua vez, a restrição (16) limita o número de veículos que pode ser usado por garagem e a restrição (17) é a de fluxo conservativo, a qual define estruturas de caminhos múltiplos para cada garagem. Por fim, os requisitos para as variáveis binárias são encontrados em (18):

$$\text{Minimizar } \sum_{k \in K} \sum_{(i,j) \in A^k} c_{ij} X_{ij}^k \quad (14)$$

Sujeito a:

$$\sum_{k \in K} \sum_{j:(i,j) \in A^k} X_{ij}^k = 1, \quad \forall i \in T \quad (15)$$

$$\sum_{j:(o(k),j) \in A^k} X_{o(k),j}^k \leq v_k \quad \forall k \in K \quad (16)$$

$$\sum_{j:(j,i) \in A^k} X_{ji}^k - \sum_{j:(i,j) \in A^k} X_{ij}^k = 0 \quad \forall i \in V^k \setminus \{o(k), d(k)\}, k \in K \quad (17)$$

$$X_{ij}^k \in \{0, 1\}, \forall (i, j) \in A^k, k \in K. \quad (18)$$

O custo de uma viagem para um veículo na garagem k é dada pela soma dos custos de viagem e de espera incorridos entre as viagens consecutivas, dos custos incorridos do momento em que o veículo deixa a garagem até iniciar sua tarefa, bem como os custos envolvidos a partir de quando o veículo acaba sua tarefa até chegar de volta à garagem. A estes custos também podem ser acrescentados os custos fixos do veículo. Restrições limitando a qual conjunto de garagens os veículos devem pertencer para executar determinada viagem podem ser acrescentadas ao modelo. Estas restrições simplificam o problema, reduzindo o número de soluções possíveis (PEPIN *et al.*, 2008).

A escolha por esta formulação em detrimento da outra apresentada é justificada devido à sua melhor aplicação em desenvolvimentos heurísticos como o de busca-tabu, que é utilizado neste trabalho (LÖBEL 1999, MESQUITA; PAIXÃO, 1999, HADJAR; MARCOTTE; SOUMIS, 2006).

4.2 ETAPA 2: DESENVOLVIMENTO DE UMA ABORDAGEM HEURÍSTICA

A segunda etapa da pesquisa visou desenvolver uma abordagem heurística a fim de solucionar o MDVSP para problemas reais ou para instâncias que bem os representem. Para encontrar a solução do problema, desenvolveu-se um método heurístico baseado em busca-tabu, a qual apresenta como principal objetivo melhorar uma solução corrente por meio da execução de movimentos dentro de uma vizinhança (CHIANG *et al.*, 2009). A fim de dirimir quaisquer dúvidas, julgou-se prudente apresentar a seguir uma pequena seção sobre como as heurísticas podem ser utilizadas em métodos de pesquisa operacional.

4.2.1 Heurísticas

Quando encontra-se situações em que um método de resolução exato não está disponível ou exige um tempo computacional ou quantidade de memória além dos recursos do computador disponível, o uso de heurísticas é bastante indicado. As heurísticas também são indicadas em situações em que a busca por uma solução ótima não é viável (devido ao tamanho da instância ou complexidade do problema). Desta forma, a solução heurística é capaz de se aproximar ao valor ótimo, muitas vezes utilizada conjuntamente com um método de solução exato. (ARENALES *et al.* 2007).

Para Arenales *et al.* (2007) as vantagens das heurísticas em relação aos métodos exatos devem-se, principalmente, a uma maior flexibilidade no tratamento das características de um problema, principalmente quando estas são difíceis de serem incorporadas em um modelo de otimização matemática, e à possibilidade de uma heurística oferecer mais de uma solução, principalmente em situações onde fatores intangíveis não podem ser agregados aos modelos mas precisam ser considerados. No que se refere às desvantagens, os autores consideram o fato das heurísticas não garantirem o

encontro de uma solução viável ou ótima e, em geral, não ser possível conhecer a qualidade da solução encontrada.

As meta-heurísticas, de acordo com Arenales *et al.* (2007), são técnicas que guiam e modificam heurísticas de modo a produzir soluções além daquelas geradas por heurísticas de busca local, sendo que cada técnica utiliza diferentes estratégias para explorar o espaço de busca.

Devido ao fato de que os métodos heurísticos são ultimamente desenvolvidos para resolverem problemas do “mundo real”, a flexibilidade aplicada à heurística apresenta-se como um fator de extrema relevância. Um algoritmo precisa ser capaz de rapidamente realizar trocas na formatação do modelo, restrições e função objetivo.

Uma heurística considerada robusta não deve apresentar resultados ruins para dada instância. Além disso, o algoritmo precisa ser capaz de produzir boas soluções a qualquer momento que seja aplicado a alguma instância. Isto é de extrema importância uma vez que muitas heurísticas não são determinísticas e contém alguns componentes aleatórios como a escolha de valores dos parâmetros (BRÄYSY, DULLAERT, GENDREAU; 2004).

O resultado apresentado na execução destes métodos não-determinísticos para o mesmo problema, na prática, nunca são os mesmos. Isto torna extremamente difícil a análise e comparação de resultados. Utilizando apenas os melhores resultados de uma heurística não-determinística, como geralmente é feito na literatura, pode-se fazer com que se crie uma falsa impressão da sua real performance. Portanto, a média dos resultados obtidos através da múltipla execução de cada um dos problemas é uma importante base para a comparação de métodos não-determinísticos. Mais do que isso, também é importante elencar o caso que apresenta a pior performance (*worst-case performance*) (CORDEAU *et al.*, 2002).

O tempo que uma heurística leva para produzir soluções de boa qualidade pode ser crucial quando da escolha entre diferentes técnicas. De forma similar, a qualidade da solução final, medida pela função objetivo, é igualmente importante. O quão perto a solução apresentada está da solução ótima é um padrão para se medir a qualidade ou, se a heurística é desenvolvida para simplesmente produzir soluções viáveis, a capacidade da

heurística produzir este tipo de soluções se torna importante (BRÄYSY, DULLAERT, GENDREAU; 2004).

4.2.2 Trade-off entre tempo de execução e qualidade da solução

Métodos clássicos de busca local formam uma classe de heurísticas aproximativas baseadas no conceito de iteratividade, incrementando a solução de um problema explorando as soluções vizinhas. Para o desenvolvimento de um algoritmo de busca local, deve-se especificar algumas escolhas que devem ser feitas: i) Como é realizada a geração da solução viável inicial; ii) Quais os outros mecanismos de geração de novas soluções são utilizados; iii) Critério de aceitação; e, iv) Critério ou teste de parada (BRÄYSY, DULLAERT, GENDREAU; 2004).

Os mecanismos de geração criam novas soluções vizinhas trocando um atributo ou a combinação de alguns atributos de uma determinada solução. Refere-se a atributo, por exemplo, os arcos que conectam um par de clientes. Assim, uma solução de vizinhança é identificada e comparada com a solução viável corrente. Se a solução vizinha é melhor assume, portanto, o lugar da solução corrente, e continua-se a procura por novas soluções melhores. Duas estratégias de aceitação são comuns neste contexto, nomeadas “*First-accept*” e “*Best-accept*”. A estratégia *First-accept* seleciona o primeiro vizinho que satisfaça um critério de aceitação pré-definido. Já a estratégia *Best-accept* examina todos os vizinhos que satisfazem o critério e seleciona o melhor entre todos eles.

O valor ótimo local produzido por qualquer procedimento de busca local pode ser muito diferente da solução ótima. Métodos de busca local apresentam uma performance míope porque somente aceitam soluções que sequencialmente produzem redução no valor da função objetivo (em funções de minimização). Assim, o resultado depende fortemente das soluções iniciais e do mecanismo de geração de vizinhos utilizados.

Geralmente há um *trade-off* entre tempo de execução e qualidade da solução – quanto mais tempo a heurística rodar, melhor será a qualidade da solução final. Um bom-senso é necessário de forma que soluções de boa qualidade sejam produzidas em um espaço razoável de tempo. Basicamente este *trade-off* pode ser visto em termos de uma otimização com múltiplos objetivos, os quais os dois objetivos estejam balanceados. A medida de performance de uma heurística pode ser analisada plotando-se, em espaço bidimensional, o tempo de execução em uma dimensão e a qualidade de execução na outra (BRÄYSY, GENDREAU; 2005).

Até agora, o método mais comum de avaliação da solução encontrada por um algoritmo heurístico é a análise empírica. Geralmente, análises empíricas envolvem testar a heurística com uma ampla variedade de instâncias do problema para ter-se uma ideia da sua performance total.

4.2.3 Construção heurística desenvolvida

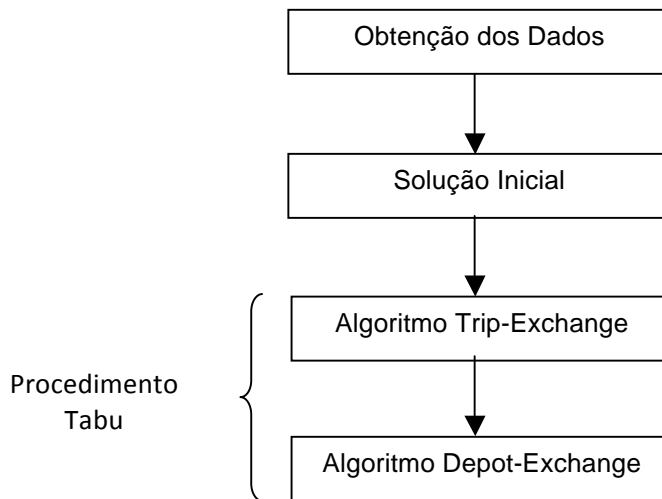
Para reduzir a complexidade da busca, alguns autores propõem estratégias especiais a fim de limitar a vizinhança. Por exemplo, Garcia *et al.* (1994), somente permitem movimentos envolvendo arcos que são próximos em distância.

Taillard *et al.* (1997) decompõem a solução em subconjuntos disjuntos de rotas, utilizando o ângulo polar associado com o centro de gravidade de cada rota. A busca-tabu é aplicada, então, para cada subconjunto de forma separada. Uma solução completa é reconstruída adicionando-se as novas rotas encontradas pela Busca-tabu (BRÄYSY, DULLAERT, GENDREAU; 2004).

Neste trabalho, após a obtenção dos dados das instâncias (procedimentos de *input* de dados serão discutidos um pouco mais a frente), uma solução inicial é gerada conforme o algoritmo ilustrado no Quadro 2. Uma vez esta solução inicial sendo encontrada, dois algoritmos distintos, o denominado *Trip-Exchange* (Quadro 3) e o outro denominado *Depot-Exchange* (Quadro 4) são implementados orientados por uma lista-tabu. O resumo do

método heurístico desenvolvido pode ser visualizado na Fig. 3, apresentada abaixo:

Figura 3: Método Heurístico Desenvolvido



Fonte: do autor.

4.2.3.1 Geração da Solução Viável Inicial

Para geração da solução inicial, primeiramente organizou-se as viagens que devem ser realizadas de forma crescente de acordo com seu custo atribuído, verificando-se a possibilidade de viagem entre todos os nós e respeitando-se as viagens incompatíveis já pré-determinadas em cada uma das instâncias.

Para obtenção das viagens de *pull-out* e *pull-in*, é verificada a possibilidade de cada garagem (ou depósito) atender cada uma das sequências de viagens, criando-se dois índices (um para *pull-out* e outra para *pull-in*) e igualmente ordenando os custos de forma crescente.

As viagens entre os nós são selecionadas levando-se em consideração a melhor garagem usando o custo das viagens de *pull-out*. Assim, um nó é selecionado de forma randômica usando o índice de *pull-out* do depósito selecionado. A seguir, adiciona-se um vizinho do nó anteriormente selecionado enquanto o nó não esteja no índice de *pull-in*. Caso o nó esteja no índice de

pull-in ou não haja mais vizinhos, selecionam-se vizinhos para novas viagens de *pull-in* ou, se todos os nós estiverem alocados, é verificado se todas as viagens são possíveis. Caso não sejam tenta-se inseri-las nas viagens possíveis.

O procedimento heurístico para geração da solução inicial é realizado conforme o Quadro 3.

Quadro 3 – Procedimento heurístico para geração da solução inicial

<p>Passo 1: Obtendo vizinhos</p> <ul style="list-style-type: none">- Verifica-se a possibilidade da viagem entre todos os nós;- Aloca-se em um índice somente as viagens possíveis, e organiza-as em forma crescente conforme o custo; <p>Passo 2: Obtendo <i>pull-out</i> e <i>pull-in</i></p> <ul style="list-style-type: none">- Verifica-se a possibilidade de cada depósito atender cada nó;- Aloca-os em dois índices organizando-os em forma crescente conforme o custo. <p>Passo 3: Obtendo as viagens</p> <ul style="list-style-type: none">- Seleciona-se o melhor depósito usando o custo de <i>pull-out</i>;- Seleciona-se um nó de forma randômica usando o índice de <i>pull-out</i>, do depósito selecionado;- Continua-se adicionando um vizinho do nó anteriormente selecionado enquanto o nó não esteja no índice de <i>pull-in</i>;- Caso o nó esteja no índice de <i>pull-in</i> ou não haja mais vizinhos, retorna ao passo 2 ou se todos nós estão alocados passa para o seguinte.- Verifica-se a possibilidade de execução de todas as viagens, caso não seja, tenta inseri-las nas possíveis.

Fonte: do autor.

4.2.3.2 Outros mecanismos de geração de soluções

No procedimento de geração de vizinhança, foram desenvolvidos dois algoritmos regidos por duas listas-tabu. O primeiro faz uma modificação das viagens possíveis, enquanto o segundo tenta trocar as garagens as quais provêm os veículos que executarão determinada rota. Estes procedimentos serão relatados a seguir:

4.2.3.2.1 Algoritmo de Trip-Exchange

Este algoritmo seleciona uma sequência de viagens e um nó pertencente a ela de forma randômica. Caso seja possível uma troca com

redução do custo, esta é realizada. Se a viagem estiver presente na lista tabu, volta-se ao passo anterior e, se não, verifica-se com todas as viagens e todos os nós se a troca é possível e se há redução do custo. Caso haja a troca, retorna-se ao primeiro passo 1500 vezes (número escolhido devido ao número máximo de viagens que pode ser configurada no sistema). Executa-se, então, o algoritmo até não encontrar mais trocas possíveis.

O procedimento heurístico de *Trip-Exchange* é apresentado no Quadro 4:

Quadro 4 – Procedimento heurístico para o *Trip-Exchange*

<p>Passo 1. Seleciona uma sequência de viagens e um nó de forma randômica. Passo 2. Se a viagem estiver na lista tabu volta para o passo 1, senão, Passo 3: Verifica com todas as viagens e todos os nós se a troca é possível e se há redução do custo. Caso haja a troca, parte para o passo 4. Passo 4. Retorna ao Passo 1 1500 vezes. Passo 5. Executa o <i>trip exchange</i> até não encontrar trocas possíveis.</p>

Fonte: do autor.

4.2.3.2.2 Algoritmo de *Depot-Exchange*

Este algoritmo inicialmente seleciona duas viagens de forma randômica e faz a consulta na lista tabu a fim de garantir que a viagem selecionada não esteja contida nela. Caso a viagem esteja na lista tabu, volta-se ao passo anterior, conforme o Quadro 5. Senão, verifica-se a possibilidade de troca de garagens entre as viagens e se há redução do custo. Caso haja, efetua-se a troca e retorna-se ao primeiro passo 500 vezes. Executa-se o algoritmo até não se encontrar mais trocas possíveis.

Quadro 5 – Procedimento heurístico para o *Depot-Exchange*

<p>Passo 1. Seleciona duas sequências de viagens de forma randômica. Passo 2. Se a viagem estiver na lista tabu, volta para o passo 1. Senão, Passo 3. Verifica a possibilidade de troca das garagens entre as sequências de viagens e se há redução do custo. Caso haja, efetua a troca e parte para o passo 4. Passo 4. Retorna ao Passo 1 500 vezes. Passo 5. Executa o <i>depot exchange</i> até não encontrar trocas possíveis.</p>
--

Fonte: do autor.

4.2.3.3 Critério de Aceitação

Após a criação de uma solução inicial, diversas outras tentativas são feitas para a melhora, utilizando-se os algoritmos apresentados acima. Assim, a estratégia *Best-accept* foi aplicada, a fim de selecionar o melhor dos valores encontrados (a partir de uma repetição das iterações), e não, simplesmente, a primeira solução encontrada.

4.2.3.4 Critério de Parada

Diversos critérios de parada podem ser utilizados para a construção de heurísticas deste tipo. Neste trabalho, optou-se por utilizar um critério máximo de iterações para cada um dos algoritmos, parametrizados pela configuração das instancias, sendo 1500 iterações para o algoritmo de *Trip Exchange* e 500 para o de *Depot-Exchange*.

4.2.4 Procedimento tabu

O procedimento de Busca-Tabu explora o espaço de soluções movendo cada iteração de soluções para uma maior, considerando-se um conjunto de vizinhança. A solução corrente pode decrementar de uma iteração para a próxima e novas soluções com resultados piores são aceitas somente para inibir caminhos já investigados. Isto assegura que novas regiões do espaço de soluções do problema sejam investigadas com o objetivo de evitar achar-se um mínimo local e, de forma definitiva, a solução desejada.

A fim de evitar a ciclagem, soluções que apresentem alguns atributos de algumas soluções recentemente exploradas (aumento dos custos, por exemplo) são, temporariamente, declaradas como tabu, ou proibidas. A duração que o um atributo permanece como tabu varia de acordo com o

tamanho das listas tabu e com a quantidade de movimentos, e pode variar através de diferentes intervalos de tempo.

O *status* tabu pode ser anulado se algumas condições são satisfeitas, isto é chamado de critério de aspiração e acontece, por exemplo, quando uma solução tabu é melhor que qualquer outra solução encontrada anteriormente.

A Busca-Tabu pode ser notada da seguinte forma: para uma solução $s \in S$, diz-se que $N(s)$ denota a vizinhança de s através de procedimentos de troca local, chamados movimentos. As técnicas de busca local visitam uma sequência s_0, \dots, s_t de soluções, onde s_0 é a solução inicial e $s_{i+1} \in N(s_i) (i = 1, \dots, t-1)$. O esquema geral do algoritmo de busca-tabu já foi apresentado no capítulo anterior, e maiores detalhes da descrição do método e de como foi concebido pode ser encontrado em Glover e Laguna (1997).

4.2.4.1 Listas Tabu

Para resolução do problema foram utilizadas duas listas tabu. A primeira, para o procedimento de *Trip-Exchange* e a segunda para o procedimento de *Depot-Exchange*. Quanto ao tamanho das listas, fixou-se um número de 10 para a primeira e 3 para a segunda, por ter-se notado que tamanho de listas maiores não faziam impacto positivo na composição da solução final.

A cada configuração da lista, 1500 repetições foram feitas para o primeiro algoritmo e 500 repetições para o segundo.

4.3 ETAPA 3: IMPLEMENTAÇÃO COMPUTACIONAL

Quanto às instâncias utilizadas neste trabalho, foram utilizadas as geradas aleatoriamente por Carpaneto *et al.* (1989) e posteriormente utilizadas por diversos outros autores. A escolha das instâncias foi baseada justamente

na possibilidade de comparação com outros trabalhos que já apresentaram soluções para o MDVSP a partir do mesmo conjunto de dados.

O programa foi desenvolvido na linguagem C ANSI sob plataforma Linux através do compilador GCC 4.6.3 e MAKE 3.81.

4.3.1 Obtenção dos dados

Os dados foram obtidos através dos arquivos *.inp contidos nas instâncias como em Carpaneto *et al.* (1989) disponibilizadas em meio eletrônico e de domínio público (<http://people.few.eur.nl/huisman/instances.htm>) e organizados em uma estrutura de memória da seguinte forma, conforme apresentado no Quadro 7 abaixo:

Quadro 7: Estrutura de Memória para as Instâncias

O nome do arquivo indica o número de depósitos (“m”), o número de viagens (“n”) e, finalmente, o número da instância (0,1,2,3,4).

(primeira linha) m <tab> n <tab> para cada garagem o número máximo de veículos;

(segunda linha e posteriores) matriz de custo de tamanho $(m + n) \times (m + n)$;

O número na linha é indicado por i, e coluna por j.

- Se $i \leq m$ e $j > m$: o custo da viagem *pull-out* de depósito i a viagem j (incluindo 50% do custo fixo para o veículo),

- Se $i > m$ e $j \leq m$: o custo da viagem *pull-in* de i para garagem j (incluindo 50% do custo fixo para o veículo),

- Se $i > m$ e $j > m$: o custo de realizar viagem j após a viagem i.

A notação “-1” indica que é inviável a realização da viagem j após viagem i.

Fonte: do autor.

4.4 ETAPA 4: AVALIAÇÃO DA SOLUÇÃO

A quarta e última etapa da pesquisa corresponde à avaliação da solução encontrada para o problema, onde o modelo implementado foi rodado para

cenários variados com o objetivo de mostrar a capacidade de solucionar problemas do tipo MDVSP.

Quanto à avaliação de heurísticas, Braysy, Dullaert e Gendreau (2004) consideram que ela está sujeita à comparação de uma série de critérios que se relacionam com vários aspectos do desempenho do algoritmo, podendo ser o tempo de execução, qualidade da solução encontrada, facilidade da implementação, robustez e flexibilidade.

O tempo que uma heurística leva para produzir boas soluções pode ser crucial na escolha entre diferentes técnicas, da mesma forma que a qualidade da solução final, mensurada pelo resultado da função objetivo, também é importante (BRAYSY; DULLAERT; GENDREAU, 2004).

Assim, no próximo capítulo o procedimento heurístico proposto será avaliado levando-se em consideração estes e outros critérios.

5 EXPERIMENTOS COMPUTACIONAIS E ANÁLISES

Nesta seção, serão apresentados os resultados computacionais dos experimentos realizados comparando-se seis cenários distintos, de acordo com o número de garagens (*depots*) e de viagens (*trips*) no sistema. Para estes testes, foram utilizadas as instâncias para o MDVSP geradas aleatoriamente assim como em Carpaneto *et al.* (1989). Nessas instâncias, que representam problemas diários do tipo MDVSP, a tarefa (ou viagem) de início e a localização final são selecionadas a partir de um conjunto restrito de pontos aleatórios em um plano, de forma que cada ponto seja escolhido, aproximadamente, como viagem de início ou localização final.

As garagens são localizadas de forma aleatória neste plano. Os tempos de início das viagens são escolhidos aleatoriamente, simulando horários de pico matutinos e vespertinos. Aproximadamente 40% das viagens têm curta duração (menos de 125 minutos), enquanto as demais apresentam longa duração (entre 180 e 360 minutos). O número de veículos disponível em cada garagem também é gerado de forma randômica, de forma que veículos de várias garagens sejam necessários para cumprir todas as viagens.

Porém, o número total de veículos não se configura como uma restrição nestas instâncias. Nestas, o objetivo consiste em minimizar, primeiramente, o número de veículos utilizados e, após, os custos totais de operação. Para que isso possa ocorrer, um alto custo fixo (10000) é aplicado para cada veículo utilizado, a fim de priorizar fortemente o primeiro objetivo.

Em uma solução ótima que utilize um número mínimo de veículos, cada um realiza, em média, de quatro a cinco viagens. De acordo com Pepin *et al.* (2008), a maior destas instâncias que pode ser resolvida na otimalidade corresponde a 800 viagens e 6 garagens. Nos testes utilizados neste trabalho, assim como no estudo supracitado, foram utilizadas instâncias onde $|T| \in \{500, 1000, 1500\}$ e $|K| \in \{4, 8\}$. Todos os resultados correspondem a cinco diferentes instâncias do mesmo tamanho geradas aleatoriamente.

Todos testes foram rodados em um *laptop* com Linux FastGene 3.2.0-24-generic #37-Ubuntu, equipado com processador Intel® Core™2 Duo com 2.66 GHz e 4 Gb de memória.

5.1 CONFIGURAÇÃO DOS EXPERIMENTOS

Os experimentos foram configurados em diferentes cenários, levando-se em consideração o número de garagens e o número de veículos presentes em cada uma das instâncias.

O primeiro cenário apresentado corresponde a cinco instâncias (s0; s1; s2; s3; s4) do MDVSP as quais contém em seus sistemas quatro garagens disponíveis e quinhentas viagens a serem executadas. A Tabela 1 apresenta a configuração deste cenário e dos outros cinco. Na última coluna apresenta-se o número máximo de veículos disponíveis para cada garagem (respectivamente de k_0 até k_3 para instâncias com quatro garagens; e k_0 até k_7 para instâncias com 8 garagens) em cada instância.

Tabela 1 – Configuração do Cenários

Cenário	Instância	Nº Garagens	Nº Viagens	Nº Máximo de Veículos em cada Garagem
1	0	4	500	62/59/56/56
	1	4	500	57/59/50/62
	2	4	500	49/54/55/59
	3	4	500	57/61/60/52
	4	4	500	54/63/45/54
2	0	4	1000	102/118/100/127
	1	4	1000	96/112/120/110
	2	4	1000	106/96/113/92
	3	4	1000	96/112/106/115
	4	4	1000	127/106/99/97
3	0	4	1500	129/180/188/178
	1	4	1500	175/154/147/180
	2	4	1500	159/180/166/171
	3	4	1500	144/171/180/153
	4	4	1500	129/171/141/152
4	0	8	500	24/28/33/25/31/32/33/30
	1	8	500	27/30/27/30/24/25/26/33
	2	8	500	30/25/31/26/25/27/30/33
	3	8	500	34/30/33/33/26/27/32/24
	4	8	500	21/28/28/29/26/31/26/32
5	0	8	1000	61/58/52/47/51/51/54/49
	1	8	1000	56/50/51/64/52/61/61/45
	2	8	1000	46/56/57/62/59/57/57/45

	3	8	1000	56/62/61/48/63/51/51/45
	4	8	1000	49/48/49/60/54/49/49/46
6	0	8	1500	79/71/76/74/69/81/84/80
	1	8	1500	74/66/76/87/82/71/83/96
	2	8	1500	82/67/87/79/68/78/71/96
	3	8	1500	82/67/87/79/68/78/71/96
	4	8	1500	68/68/78/92/72/90/76/96

Fonte: do autor

5.2 RESULTADOS E ANÁLISES

5.2.1 Formato da apresentação dos resultados

Na tela de *out* do programa, são apresentados o custo total, a quantidade de veículos utilizada, a variação do custo entre a solução inicial e a final e o tempo de execução para as duas soluções.

Tanto para solução inicial quanto para a final é necessário verificar o aqui chamado “*Sanity Check*”, a fim de confirmar a veracidade dos dados exibidos. Em modo *debug* são exibidas maiores informações, de acordo com a Fig. 4, a qual representa a rodada para a Instância 0 do Cenário 1 (m4n500s0).

Figura 4: *Out* do programa em modo *debug* para a instância m4n500s0

```

DEPOTS 4 TRIPS 500
MAX 0->62 MAX 1->59 MAX 2->56 MAX 3->56
SANITY CHECK...
TRIP 108 IMPOSSIBLE IN!
INSERT 397 IN 9 POS 1
INSERT 86 IN 9 POS 2
INSERT 184 IN 8 POS 1
INSERT 183 IN 8 POS 2
INSERT 359 IN 8 POS 2
INSERT 423 IN 81 POS 4
INSERT 407 IN 55 POS 1
INSERT 498 IN 8 POS 3
INSERT 497 IN 8 POS 3
INSERT 496 IN 8 POS 3
TRIP 108
IMPOSSIBLE IN!
INSERT 0 IN 8 POS 3
INSERT 248 IN 8 POS 2
TRIP 116
IMPOSSIBLE IN!
INSERT 373 IN 8 POS 4
INSERT 209 IN 8 POS 6
INSERT 321 IN 8 POS 3
INSERT 387 IN 114 POS 5
TRIP 117 IMPOSSIBLE IN!
INSERT 274 IN 8 POS 8
INSERT 324 IN 8 POS 4
INSERT 436 IN 73 POS 1
INITIAL SOLUTION
-----
TOTALCOST      VEHICLES      VAR      TIME/s
1358431        117           0%       0.021
SOLUTION
-----
TOTALCOST      VEHICLES      VAR      TIME/s
1316453        117           3%       0.031
DEPOT 0 DEPOT 1 DEPOT 2 DEPOT 3
38%      62%      100%    0%

SANITY CHECK...PASS!

```

Fonte: do autor.

A notação utilizada no *out* do programa pode ser traduzida da seguinte forma para um exemplo aleatório:

- INSERT 330 IN 9 POS 1: é inserido o nó 330 na viagem 9, na posição 1;
- TRIP EXCHANGE 50,185 -> 0,301: é efetuada a troca do nó 185 da viagem 50 pelo nó 301 da viagem 0 e vice-versa;
- DEPOT EXCHANGE 11 -> 57: efetua-se a troca de garagem entre as viagens 11 e 57;
- TRIP 95 DEPOT 2 5273 {416->415->388->452} 209 {0;91;53}: correspondem, respectivamente a: [viagem] [garagem] [custo de saída] {viagem entre nós}[custo de retorno] {custo da viagem entre nós}.

5.2.2 Comparação entre a solução inicial e a solução final encontrada

Após o procedimento heurístico para obtenção da solução inicial, foram aplicados, conforme demonstrado anteriormente, dois outros algoritmos que se utilizam de procedimentos tabu para geração de novos resultados.

Desta forma, é possível que se faça uma comparação entre os resultados iniciais e os finais (após aplicados os outros algoritmos), aqui denominados de “Solução Final”.

É interessante perceber que, apesar da redução percentual ser aparentemente pequena, a redução em unidades monetárias representa uma boa economia, principalmente se levarmos em consideração que na Solução Inicial os custos já apresentam-se de forma bastante competitiva.

A redução percentual (ou também posteriormente chamada de *gap*) entre a Solução Final e a Inicial apresenta, para todos os experimentos, valores positivos, ou seja, conseguiu-se uma melhora na função objetivo em todos os casos após aplicação do procedimento tabu. É importante observar também que os *gaps* apresentados entre a Solução Final e a Inicial demonstram a qualidade de heurística geradora da Solução Inicial, uma vez que, apesar de positivos, estes valores não são muito expressivos.

5.2.2.1 Cenário 1: 4 Garagens e 500 Viagens

No primeiro cenário analisado, correspondente às instâncias com quatro garagens e 500 viagens, obteve-se uma redução média de custos de 2%. Na Tabela 2 pode-se verificar os demais parâmetros de análise e comparação.

Em todas as cinco instâncias para este cenário, não foi possível diminuir o número de veículos utilizados para o percurso das rotas, porém, nas instâncias 0, 3 e 4, foi possível atingir a demanda das viagens com custo igual ou inferior ao anterior fechando-se uma garagem, o que representa uma economia no custo fixo de manutenção.

Tabela 2: Comparação entre as soluções inicial e final para o Cenário 1

Instâncias	Crítérios	Solução Inicial	Solução Final	Gap (redução %)
0	Custo Total	1.358.431	1.316.453	3,09
	Nº Veículos Utilizados	117	117	-
	Nº Garagens Utilizadas	4	3	25
	Tempo de Execução (s)	0.021	0.031	-
1	Custo Total	1.437.743	1.396.346	2,88
	Nº Veículos Utilizados	132	132	-
	Nº Garagens Utilizadas	4	4	-
	Tempo de Execução (s)	0.022	0.038	-
2	Custo Total	1.440.531	1.418.062	1,56
	Nº Veículos Utilizados	130	130	-
	Nº Garagens Utilizadas	4	4	-
	Tempo de Execução (s)	0.025	0.031	-
3	Custo Total	1.270.282	1.249.628	1,63
	Nº Veículos Utilizados	111	111	-
	Nº Garagens Utilizadas	4	3	25
	Tempo de Execução (s)	0.021	0.028	-
4	Custo Total	1.221.037	1.211.123	0,81
	Nº Veículos Utilizados	109	109	-
	Nº Garagens Utilizadas	4	3	25
	Tempo de Execução (s)	0.021	0.029	-

Fonte: do Autor.

5.2.2.2 Cenário 2: 4 Garagens e 1000 Viagens

O segundo cenário representa as instâncias com 4 garagens e 1000 viagens. A redução percentual média no custo de realização de todas as viagens, assim como no cenário anterior, foi de 2%.

Já neste cenário (Tabela 3), apenas nas instâncias 2 e 3 foi possível o fechamento de uma das garagens sem prejudicar as rotas pré-definidas e, tampouco, penalizar a função objetivo.

Os tempos computacionais para as rodadas de cada uma das cinco instâncias não representam muito variação, uma vez que todas são resolvidas em menos de um segundo.

Tabela 3: Comparação entre as soluções inicial e final para o Cenário 2

Instâncias	Critérios	Solução Inicial	Solução Final	Gap (redução %)
0	Custo Total	2.769.424	2.704.635	2,34
	Nº Veículos Utilizados	246	246	-
	Nº Garagens Utilizadas	4	4	-
	Tempo de Execução (s)	0.084	0.101	-
1	Custo Total	2.698.002	2.651.215	1,73
	Nº Veículos Utilizados	242	242	-
	Nº Garagens Utilizadas	4	4	-
	Tempo de Execução (s)	0.089	0.113	-
2	Custo Total	2.538.263	2.491.529	1,84
	Nº Veículos Utilizados	223	223	-
	Nº Garagens Utilizadas	4	3	25,00
	Tempo de Execução (s)	0.082	0.102	-
3	Custo Total	2.567.810	2.519.289	1,89
	Nº Veículos Utilizados	235	235	-
	Nº Garagens Utilizadas	4	3	25,00
	Tempo de Execução (s)	0.082	0.102	-
4	Custo Total	2.796.170	2.742.430	1,92
	Nº Veículos Utilizados	242	242	-
	Nº Garagens Utilizadas	4	4	-
	Tempo de Execução (s)	0.089	0.116	-

Fonte: do Autor.

5.2.2.3 Cenário 3: 4 Garagens e 1500 Viagens

O Cenário 3 (Tabela 4), por sua vez, apresenta o mesmo número de garagens disponíveis que os outros dois cenários, porém o número de viagens é de 1500. O *gap* médio do valor da função objetivo foi 1,5%. O pior caso foi relatado na instância 0 (0,84%), enquanto o maior foi encontrado na instância 1 (2,15%).

É importante notar que na instância 0, conseguiu-se atingir toda a demanda das rotas utilizando-se apenas duas das quatro garagens disponíveis, sem que fosse penalizado o custo final da função objetivo ou as rotas já pré-definidas.

Tabela 4: Comparação entre as soluções inicial e final para o Cenário 3

Instâncias	Crerios	Soluço Inicial	Soluço Final	Gap (reduço %)
0	Custo Total	3.693.191	3.662.253	0,84
	No Veculos Utilizados	344	344	-
	No Garagens Utilizadas	4	2	50,00
	Tempo de Execuo (s)	0.192	0.220	-
1	Custo Total	3.910.828	3.826.654	2,15
	No Veculos Utilizados	349	349	-
	No Garagens Utilizadas	4	3	25,00
	Tempo de Execuo (s)	0.204	0.231	-
2	Custo Total	3.777.073	3.720.400	1,50
	No Veculos Utilizados	343	343	-
	No Garagens Utilizadas	4	3	25,00
	Tempo de Execuo (s)	0.195	0.245	-
3	Custo Total	3.734.302	3.671.426	1,68
	No Veculos Utilizados	340	340	-
	No Garagens Utilizadas	4	4	-
	Tempo de Execuo (s)	0.191	0.242	-
4	Custo Total	4.019.305	3.959.818	1,48
	No Veculos Utilizados	369	369	-
	No Garagens Utilizadas	4	4	-
	Tempo de Execuo (s)	0.194	0.251	-

Fonte: do Autor.

5.2.2.4 Cenrio 4: 8 Garagens e 500 Viagens

Na Tabela 5 so apresentados os dados para o quarto cenrio analisado, o qual  composto em seu sistema por 8 garagens disponveis e a necessidade de realizar 500 viagens.

O *gap* mdio de custos entre a soluo final e a inicial chegou a 3%. O melhor caso foi encontrado na ltima instncia, onde obteve-se uma economia de quase 4% dos custos, alm da possibilidade de atender toda a demanda de rotas com apenas seis das oito garagens disponveis.

J o pior caso foi encontrado na instncia 3 (*gap* de 1,8%), mas mesmo apresentando uma economia menor que as demais, foi possvel utilizar apenas 7 garagens. Em ambas as solues, no houve alterao no nmero de veculos necessrios para realizao das viagens.

Tabela 5: Comparação entre as soluções inicial e final para o Cenário 4

Instâncias	Crerios	Soluçao Inicial	Soluçao Final	Gap (reduçao %)
0	Custo Total	1.367.528	1.322.293	3,31
	Nº Veículos Utilizados	121	121	-
	Nº Garagens Utilizadas	8	6	25,00
	Tempo de Execuçao (s)	0.021	0.028	-
1	Custo Total	1.411.747	1.380.047	2,25
	Nº Veículos Utilizados	125	125	-
	Nº Garagens Utilizadas	8	6	25,00
	Tempo de Execuçao (s)	0.025	0.031	-
2	Custo Total	1.315.647	1.269.947	3,47
	Nº Veículos Utilizados	120	120	-
	Nº Garagens Utilizadas	8	5	37,50
	Tempo de Execuçao (s)	0.023	0.028	-
3	Custo Total	1.340.134	1.316.075	1,80
	Nº Veículos Utilizados	117	117	-
	Nº Garagens Utilizadas	8	7	12,50
	Tempo de Execuçao (s)	0.024	0.030	-
4	Custo Total	1.416.222	1.361.857	3,84
	Nº Veículos Utilizados	122	122	-
	Nº Garagens Utilizadas	8	6	25,00
	Tempo de Execuçao (s)	0.022	0.028	-

Fonte: do Autor.

5.2.2.5 Cenário 5: 8 Garagens e 1000 Viagens

O Cenário 5 apresenta os resultados da soluçao heurística inicial e os resultados finais, após aplicados os outros dois algoritmos, para instâncias com 8 garagens disponíveis no sistema e necessidade de realizaçao de 1000 viagens.

Assim como no cenário anterior, o *gap* médio quando comparadas as duas soluçoes foi de 3%. Em apenas uma instância (instância 1) é que foi necessário utilizar as oito garagens disponíveis no sistema, de forma a não prejudicar os custos e viagens, como pode ser observado na Tabela 6.

Tabela 6: Comparação entre as soluções inicial e final para o Cenário 5

Instâncias	CrITÉrios	Solução Inicial	Solução Final	Gap (redução %)
0	Custo Total	2.469.038	2.395.948	2,96
	Nº Veículos Utilizados	225	225	-
	Nº Garagens Utilizadas	8	6	25,00
	Tempo de Execução (s)	0.098	0.118	-
1	Custo Total	2.597.432	2.525.733	2,76
	Nº Veículos Utilizados	232	232	-
	Nº Garagens Utilizadas	8	8	-
	Tempo de Execução (s)	0.100	0.121	-
2	Custo Total	2.769.520	2.679.415	3,25
	Nº Veículos Utilizados	244	244	-
	Nº Garagens Utilizadas	8	7	12,50
	Tempo de Execução (s)	0.094	0.110	-
3	Custo Total	2.671.393	2.608.787	2,34
	Nº Veículos Utilizados	244	244	-
	Nº Garagens Utilizadas	8	6	25,00
	Tempo de Execução (s)	0.103	0.137	-
4	Custo Total	2.559.445	2.473.958	3,34
	Nº Veículos Utilizados	230	230	-
	Nº Garagens Utilizadas	8	6	25,00
	Tempo de Execução (s)	0.087	0.104	-

Fonte: do Autor.

5.2.2.6 Cenário 6: 8 Garagens e 1500 Viagens

O Cenário 6, por sua vez, é o mais complexo dentre todos os demais analisados nesta pesquisa. Além do sistema contar com 8 garagens, o número de viagens já definidas anteriormente é de 1500 (Tabela 7). O *gap* médio quando comparadas as duas soluções foi superior a 3%.

O melhor caso foi observado na instância 1 (3,8%) e o pior na instância 2 (2,41%). Em todas as instâncias pode-se trabalhar com um número de garagens inferior, com a garantia de não penalizar a função objetivo e as viagens pré-estipuladas.

Tabela 7: Comparação entre as soluções inicial e final para o Cenário 6

Instâncias	Critérios	Solução Inicial	Solução Final	Gap (redução %)
0	Custo Total	3.916.768	3.784.325	3,38
	Nº Veículos Utilizados	360	360	-
	Nº Garagens Utilizadas	8	7	12,50
	Tempo de Execução (s)	0.203	0.243	-
1	Custo Total	3.969.693	3.818.718	3,80
	Nº Veículos Utilizados	354	354	-
	Nº Garagens Utilizadas	8	7	12,50
	Tempo de Execução (s)	0.224	0.265	-
2	Custo Total	4.017.578	3.920.566	2,41
	Nº Veículos Utilizados	360	360	-
	Nº Garagens Utilizadas	8	5	37,50
	Tempo de Execução (s)	0.212	0.245	-
3	Custo Total	4.012.240	3.911.726	2,51
	Nº Veículos Utilizados	359	359	-
	Nº Garagens Utilizadas	8	6	-
	Tempo de Execução (s)	0.212	0.245	-
4	Custo Total	3.962.513	3.845.059	2,96
	Nº Veículos Utilizados	363	363	-
	Nº Garagens Utilizadas	8	6	25,00
	Tempo de Execução (s)	0.210	0.239	-

Fonte: do Autor.

5.2.3 Comparação entre a solução final encontrada e a solução apresentada por Pepin *et al.* (2008)

As Soluções Finais (valor da função objetivo) encontradas por este trabalho foram, em média, um pouco superiores àquelas encontradas por Pepin *et al.* (2008). É importante observar que todas as soluções encontradas pelos autores acima, foram atingidos através de resolução ótima via Cplex™ ou com algoritmos de *Branch-and-cut* e Geração de Colunas (também utilizando o mesmo *solver*) para os demais cenários, o que contribui para um aumento no tempo de resolução computacional.

Desta forma, apesar dos resultados da heurística proposta neste trabalho apresentarem-se um pouco superiores aos mostrados por aqueles autores, o tempo computacional atingido foi significativamente menor por ter-se adotado um método de geração de soluções iniciais bastante adequado e pela

própria característica do método de busca-tabu de apresentar tempos computacionais menores.

É importante salientar, também, que a diferença expressiva no tempo computacional apresentado por este estudo e naquele de Pepin *et al.* (2008), deve-se ao fato de que estes autores pré-definiram o tempo de execução da heurística de busca-tabu, como pode ser visualizado nas tabelas a seguir (por esta razão, estes tempos apresentam-se de forma destacada nas tabelas abaixo).

Esta pré-definição, de acordo com os autores, foi feita através de uma noção do tempo computacional necessário para se resolver o problema via outras heurísticas, a saber: a) *branch-and-cut* e b) geração de colunas. É sabido que tais procedimentos demandam um tempo computacional bastante superior aos de meta-heurísticas como a de busca-tabu, fazendo com que aquele estudo obtivesse um custo computacional desnecessário em relação a este atributo.

Já quanto ao número de veículos utilizados para cada cenário, este não seguiu um padrão entre um trabalho e outro, às vezes apresentando resultados melhores neste trabalho e vice-versa.

Desta forma, a seguir são apresentadas tabelas comparativas entre os resultados obtidos neste trabalho com aqueles apresentados por Pepin *et al.* (2008) em seu artigo. Esta comparação é feita tomando-se como base a melhor solução apresentada por aqueles autores (dentre as cinco diferentes soluções – uma para cada heurística) e a Solução Final encontrada nesta dissertação.

Estas comparações são feitas através dos seguintes atributos: custo total, número de veículos e garagens utilizadas e tempo total de execução.

5.2.3.1 Comparação com os resultados de Pepin *et al.* (2008) para o Cenário 1

A Tabela 8 apresenta uma comparação ente os dois resultados para as instancias utilizadas para os experimentos do cenário 1. A diferença entre os custos totais obtidos nos trabalhos foi de 3,26%, ou seja, os resultados obtidos por este trabalho apresentaram função objetivo com custos, em media, 3,26%

maiores que os de Pepin *et al.* (2008). O pior caso relatado foi na instancia 1, onde essa diferença passou de 12 %, aumentando os custos de \$ 1.241.618,00 para \$ 1.,396.346,00.

Tabela 8: Comparação entre as soluções finais e as de Pepin *et al.* (2008) para o Cenário 1

Instâncias	Crítérios	Solução Final	Melhor Solução de Pepin <i>et al.</i> (2008)	Gap (% de melhoria)
0	Custo Total	1.316.453	1.289.114	-2,12
	Nº Veículos Utilizados	117	123	4,88
	Nº Garagens Utilizadas	3	4	25,00
	Tempo de Execução (s)	0.031	85*	
1	Custo Total	1.396.346	1.241.618	-12,46
	Nº Veículos Utilizados	132	118	-11,86
	Nº Garagens Utilizadas	4	4	0,00
	Tempo de Execução (s)	0.038	85*	
2	Custo Total	1.418.062	1.283.811	-10,46
	Nº Veículos Utilizados	130	123	-5,69
	Nº Garagens Utilizadas	4	4	0,00
	Tempo de Execução (s)	0.031	85*	
3	Custo Total	1.249.628	1.258.634	0,72
	Nº Veículos Utilizados	111	120	7,50
	Nº Garagens Utilizadas	3	4	25,00
	Tempo de Execução (s)	0.028	85*	
4	Custo Total	1.211.123	1.317.077	8,04
	Nº Veículos Utilizados	109	126	13,49
	Nº Garagens Utilizadas	3	4	25,00
	Tempo de Execução (s)	0.029	85*	

Fonte: do Autor.

Em compensação, o tempo computacional médio atingido pelos outros autores foi de 85 segundos para resolução desta configuração de instancias, enquanto o deste trabalho apresentou uma media inferior a um segundo para resolução. A possibilidade do fechamento de garagens nas instancias 0, 3 e 4 deste cenário também apresenta-se como um grande diferencial frente ao trabalho precedente, uma vez que diminui quantitativamente os custos fixos da facilidade.

5.2.3.2 Comparação com os resultados de Pepin *et al.* (2008) para o Cenário 2

Uma comparação ente os dois resultados para as instancias utilizadas para os experimentos do cenário 2 é apresentada na Tabela 9. A diferença entre os custos totais obtidos nos trabalhos foi de 3,26%, ou seja, os resultados obtidos por este trabalho apresentaram função objetivo com custos, em media, 5,4% maiores que os de Pepin *et al.* (2008). O pior caso relatado foi na instancia 1, onde essa diferença passou de 9,85 %, aumentando os custos de \$ 2.413.393,00 para \$ 2.651.215,00.

O tempo computacional, como no cenário anterior e nos cenários futuros é que apresentou expressiva melhora, conforme se pode verificar nas próximas tabelas, incluindo a Tabela 9, apresentada a seguir.

A possibilidade do fechamento de garagens nos cenários 2 e 3 também apresenta-se como um grande diferencial frente ao trabalho precedente.

Tabela 9: Comparação entre as soluções final e as de Pepin *et al.* (2008) para o Cenário 2

Instâncias	Crítérios	Solução Final	Melhor Solução de Pepin <i>et al.</i> (2008)	Gap (% de melhoria)
0	Custo Total	2.704.635	2.516.247	-7,49
	Nº Veículos Utilizados	246	241	-2,07
	Nº Garagens Utilizadas	4	4	
	Tempo de Execução (s)	0.101	700*	
1	Custo Total	2.651.215	2.413.393	-9,85
	Nº Veículos Utilizados	242	229	-5,68
	Nº Garagens Utilizadas	4	4	
	Tempo de Execução (s)	0.113	700*	
2	Custo Total	2.491.529	2.452.905	-1,57
	Nº Veículos Utilizados	223	233	4,29
	Nº Garagens Utilizadas	3	4	25,00
	Tempo de Execução (s)	0.102	700*	
3	Custo Total	2.519.289	2.490.812	-1,14
	Nº Veículos Utilizados	235	237	0,84
	Nº Garagens Utilizadas	3	4	25,00
	Tempo de Execução (s)	0.102	700*	
4	Custo Total	2.742.430	2.519.191	-8,86
	Nº Veículos Utilizados	242	238	-1,68
	Nº Garagens Utilizadas	4	4	
	Tempo de Execução (s)	0.116	700*	

Fonte: do Autor.

5.2.3.3 Comparação com os resultados de Pepin *et al.* (2008) para o Cenário 3

No que se refere aos custos totais do sistema para o Cenário 3, o caso de melhor impacto foi para a instancia 0, onde conseguiu-se decrementar o custo da função objetivo em 4,49%, diminuindo-se os custos de \$ 3.830.912,00 para \$ 3.662.253,00, como exposto na Tabela 10. Já o caso de pior performance da heurística pode ser observado no Cenário 4, com o aumento do custo da função objetivo em pouco mais de 11%, representando um aumento de \$ 393.696,00.

Na instancia 0, além de se ter conseguido diminuir os custos totais, foi possível realizar todos os caminhos previamente determinados pelos dados coletados na instancia utilizando-se apenas duas das quatro garagens disponíveis, sem que seja penalizada a função objetivo ou desrespeitada qualquer uma das restrições originais do problema.

Tabela 10: Comparação entre as soluções final e as de Pepin *et al.* (2008) para o Cenário 3

Instâncias	Crítérios	Solução Final	Melhor Solução de Pepin <i>et al.</i> (2008)	Gap (% de melhoria)
0	Custo Total	3.662.253	3.830.912	4,40
	Nº Veículos Utilizados	344	368	6,52
	Nº Garagens Utilizadas	2	4	50,00
	Tempo de Execução (s)	0.220	2300*	
1	Custo Total	3.826.654	3.559.576	-7,50
	Nº Veículos Utilizados	349	338	-3,25
	Nº Garagens Utilizadas	3	4	25,00
	Tempo de Execução (s)	0.231	2300*	
2	Custo Total	3.720.400	3.649.759	-1,94
	Nº Veículos Utilizados	343	350	2,00
	Nº Garagens Utilizadas	3	4	25,00
	Tempo de Execução (s)	0.245	2300*	
3	Custo Total	3.671.426	3.406.815	-7,77
	Nº Veículos Utilizados	340	326	-4,29
	Nº Garagens Utilizadas	4	4	
	Tempo de Execução (s)	0.242	2300*	

4	Custo Total	3.959.818	3.567.122	-11,01
	Nº Veículos Utilizados	369	343	-7,58
	Nº Garagens Utilizadas	4	4	
	Tempo de Execução (s)	0.251	2300*	

Fonte: do Autor.

5.2.3.4 Comparação com os resultados de Pepin *et al.* (2008) para o Cenário 4

A Tabela 11 apresenta uma comparação entre os resultados para as instâncias utilizadas para os experimentos do cenário 4. A diferença entre os custos totais obtidos nos trabalhos foi de 3,5%, ou seja, os resultados obtidos por este trabalho apresentaram função objetivo com custos, em média, 3,5% maiores que os de Pepin *et al.* (2008).

A possibilidade do fechamento de garagens em todas as instâncias deste cenário também apresenta-se como um grande diferencial frente ao trabalho precedente, uma vez que diminui quantitativamente os custos fixos das facilidades.

Na instância 2, foi possível atingir um valor de função objetivo menor que os apresentados pelos autores do artigo. Essa diferença, percentualmente, chegou a 2,63%, representando uma economia de mais de \$ 600.000,00.

Tabela 11: Comparação entre as soluções final e as de Pepin *et al.* (2008) para o Cenário 4

Instâncias	Crítérios	Solução Final	Melhor Solução de Pepin <i>et al.</i> (2008)	Gap (% de melhoria)
0	Custo Total	1.322.293	1.292.411	-2,31
	Nº Veículos Utilizados	121	124	2,42
	Nº Garagens Utilizadas	6	8	25,00
	Tempo de Execução (s)	0.028	125*	
1	Custo Total	1.380.047	1.276.419	-8,12
	Nº Veículos Utilizados	125	123	-1,63
	Nº Garagens Utilizadas	6	8	25,00
	Tempo de Execução (s)	0.031	125*	
2	Custo Total	1.269.947	1.304.251	2,63
	Nº Veículos Utilizados	120	126	4,76
	Nº Garagens Utilizadas	5	8	37,50

	Tempo de Execução (s)	0.028	125*	
3	Custo Total	1.316.075	1.277.388	-3,03
	Nº Veículos Utilizados	117	123	4,88
	Nº Garagens Utilizadas	7	8	12,50
	Tempo de Execução (s)	0.030	125*	
4	Custo Total	1.361.857	1.276.010	-6,73
	Nº Veículos Utilizados	122	123	0,81
	Nº Garagens Utilizadas	6	8	25,00
	Tempo de Execução (s)	0.028	125*	

Fonte: do Autor.

5.2.3.5 Comparação com os resultados de Pepin *et al.* (2008) para o Cenário 5

Já no Cenário 5 (Tabela 12) obteve-se melhoria em duas das cinco instâncias experimentadas. (instancias 0 e 4). O Caso de melhor resultado foi na instância 0, onde foi possível uma diminuição de mais de 1% dos custos, enquanto o caso de pior aplicação heurística foi para a instância 2, representando um aumento de mais de \$ 400.000,00 no custo total.

Tabela 12: Comparação entre as soluções final e as de Pepin *et al.* (2008) para o Cenário 5

Instâncias	Crítérios	Solução Final	Melhor Solução de Pepin <i>et al.</i> (2008)	Gap (% de melhoria)
0	Custo Total	2.395.948	2.422.112	1,08
	Nº Veículos Utilizados	225	232	3,02
	Nº Garagens Utilizadas	6	8	25,00
	Tempo de Execução (s)	0.118	900*	
1	Custo Total	2.525.733	2.524.293	-0,06
	Nº Veículos Utilizados	232	244	4,92
	Nº Garagens Utilizadas	8	8	
	Tempo de Execução (s)	0.121	900*	
2	Custo Total	2.679.415	2.256.313	-18,75
	Nº Veículos Utilizados	244	247	1,21
	Nº Garagens Utilizadas	7	8	12,50
	Tempo de Execução (s)	0.110	900*	
3	Custo Total	2.608.787	2.478.393	-5,26
	Nº Veículos Utilizados	244	237	-2,95
	Nº Garagens Utilizadas	6	8	25,00
	Tempo de Execução (s)	0.137	900*	

4	Custo Total	2.473.958	2.498.388	0,98
	Nº Veículos Utilizados	230	240	4,17
	Nº Garagens Utilizadas	6	8	25,00
	Tempo de Execução (s)	0.104	900*	

Fonte: do Autor.

5.2.3.6 Comparação com os resultados de Pepin *et al.* (2008) para o Cenário 6

A Tabela 13 apresenta uma comparação ente os dois resultados para as instancias utilizadas para os experimentos do ultimo cenário avaliado. A diferença entre os custos totais obtidos nos trabalhos foi de 6,46%, ou seja, os resultados obtidos por este trabalho apresentaram função objetivo com custos, em media, 6,46% maiores que os de Pepin *et al.* (2008). O pior caso relatado foi na instancia 3, onde essa diferença passou de 11%, aumentando os custos em mais de \$ 300.000,00

Já para a instancia 1, o aumento de custos foi de apenas 0,42%, conseguido com um tempo computacional significativamente melhores. A diferença do custo total para esta rodada ficou em torno de \$16.000,00

Tabela 13: Comparação entre as soluções final e as de Pepin *et al.* (2008) para o Cenário 6

Instâncias	Critérios	Solução Final	Melhor Solução de Pepin <i>et al.</i> (2008)	Gap
0	Custo Total	3.784.325	3.500.160	-8,12
	Nº Veículos Utilizados	360	337	-6,82
	Nº Garagens Utilizadas	7	8	12,50
	Tempo de Execução (s)	0.243	3200*	
1	Custo Total	3.818.718	3.802.650	-0,42
	Nº Veículos Utilizados	354	366	3,28
	Nº Garagens Utilizadas	7	8	12,50
	Tempo de Execução (s)	0.265	3200*	
2	Custo Total	3.920.566	3.605.094	-8,75
	Nº Veículos Utilizados	360	349	-3,15
	Nº Garagens Utilizadas	5	8	
	Tempo de Execução (s)	0.245	3200*	
3	Custo Total	3.911.726	3.515.802	-11,26
	Nº Veículos Utilizados	359	338	-6,21
	Nº Garagens Utilizadas	6	8	

	Tempo de Execução (s)	0.245	3200*	
4	Custo Total	3.845.059	3.704.953	-3,78
	Nº Veículos Utilizados	363	360	-0,83
	Nº Garagens Utilizadas	6	8	25,00
	Tempo de Execução (s)	0.239	3200*	

Fonte: do Autor.

5.2.4 Resumo das soluções iniciais e finais encontradas neste trabalho

O resumo das soluções iniciais e finais encontradas neste trabalho é apresentado de acordo com o que mostra a Tabela 14. Nela são elencadas as soluções iniciais e finais para cada uma das cinco instâncias dos seis cenários analisados e, na última coluna, a melhoria percentual atingida de uma fase para a outra. Estes resultados já foram discutidos em seções anteriores, mas pode-se perceber uma melhoria em todas as instâncias testadas.

Tabela 14: Resumo das soluções inicial e final para os seis Cenários

Cenário	Instância	Solução Inicial	Solução Final	Melhoria (%)
1	0	1.358.431	1.316.453	3,09
	1	1.437.743	1.396.346	2,88
	2	1.440.531	1.418.062	1,56
	3	1.270.282	1.249.628	1,63
	4	1.221.037	1.211.123	0,81
2	0	2.769.424	2.704.635	2,34
	1	2.698.002	2.651.215	1,73
	2	2.538.263	2.491.529	1,84
	3	2.567.810	2.519.289	1,89
	4	2.796.170	2.742.430	1,92
3	0	3.693.191	3.662.253	0,84
	1	3.910.828	3.826.654	2,15
	2	3.777.073	3.720.400	1,50
	3	3.734.302	3.671.426	1,68
	4	4.019.305	3.959.818	1,48
4	0	1.367.528	1.322.293	3,31
	1	1.411.747	1.380.047	2,25
	2	1.315.647	1.269.947	3,47
	3	1.340.134	1.316.075	1,80
	4	1.416.222	1.361.857	3,84

5	0	2.469.038	2.395.948	2,96
	1	2.597.432	2.525.733	2,76
	2	2.769.520	2.679.415	3,25
	3	2.671.393	2.608.787	2,34
	4	2.559.445	2.473.958	3,34
6	0	3.916.768	3.784.325	3,38
	1	3.969.693	3.818.718	3,80
	2	4.017.578	3.920.566	2,41
	3	4.012.240	3.911.726	2,51
	4	3.962.513	3.845.059	2,96

Fonte: do Autor.

5.2.5 Resumo dos custos finais e número de veículos utilizados pela heurística proposta quando comparados ao trabalho de Pepin *et al.* (2008)

A fim de tornar mais clara a leitura e comparação dos resultados obtidos por esta pesquisa com aqueles obtidos por Pepin *et al.* (2008), apresenta-se a Tabela 15 que trás um resumo de alguns resultados obtidos por ambos os trabalhos.

Para cada uma das cinco instâncias dos seis cenários analisados, são apresentados os custos finais da função objetivo atingidos por ambas as pesquisas, o percentual de melhoria em relação à pesquisa anterior, bem como o número de veículos utilizados em cada um dos experimentos.

Tabela 15: Resumo dos Custos Finais e Nº de Veículos Utilizados comparados aos Melhores Valores Encontrados p Pepin *et al.* (2008)

Cenário	Instância	Custo Final	Custo por Pepin <i>et al.</i> (2008)	Gap (% de melhoria)	Nº de Veículos Utilizados	Nº de Veículos Utilizados por Pepin <i>et al.</i> (2008)	Diferença no Número de Veículos
1	0	1.316.453	1.289.114	-2,12	117	123	-6
	1	1.396.346	1.241.618	-12,46	132	118	14
	2	1.418.062	1.283.811	-10,46	130	123	7
	3	1.249.628	1.258.634	0,72	111	120	-9
	4	1.211.123	1.317.077	8,04	109	126	-17
2	0	2.704.635	2.516.247	-7,49	246	241	5
	1	2.651.215	2.413.393	-9,85	242	229	13
	2	2.491.529	2.452.905	-1,57	223	233	-10

	3	2.519.289	2.490.812	-1,14	235	237	-2
	4	2.742.430	2.519.191	-8,86	242	238	4
3	0	3.662.253	3.830.912	4,40	344	368	-24
	1	3.826.654	3.559.576	-7,50	349	338	11
	2	3.720.400	3.649.759	-1,94	343	350	-7
	3	3.671.426	3.406.815	-7,77	340	326	14
	4	3.959.818	3.567.122	-11,01	369	343	26
4	0	1.322.293	1.292.411	-2,31	121	124	-3
	1	1.380.047	1.276.419	-8,12	125	123	2
	2	1.269.947	1.304.251	2,63	120	126	-6
	3	1.316.075	1.277.388	-3,03	117	123	-6
	4	1.361.857	1.276.010	-6,73	122	123	-1
5	0	2.395.948	2.422.112	1,08	225	232	-7
	1	2.525.733	2.524.293	-0,06	232	244	-12
	2	2.679.415	2.256.313	-18,75	244	247	-3
	3	2.608.787	2.478.393	-5,26	244	237	7
	4	2.473.958	2.498.388	0,98	230	240	-10
6	0	3.784.325	3.500.160	-8,12	360	337	23
	1	3.818.718	3.802.650	-0,42	354	366	-12
	2	3.920.566	3.605.094	-8,75	360	349	11
	3	3.911.726	3.515.802	-11,26	359	338	21
	4	3.845.059	3.704.953	-3,78	363	360	3

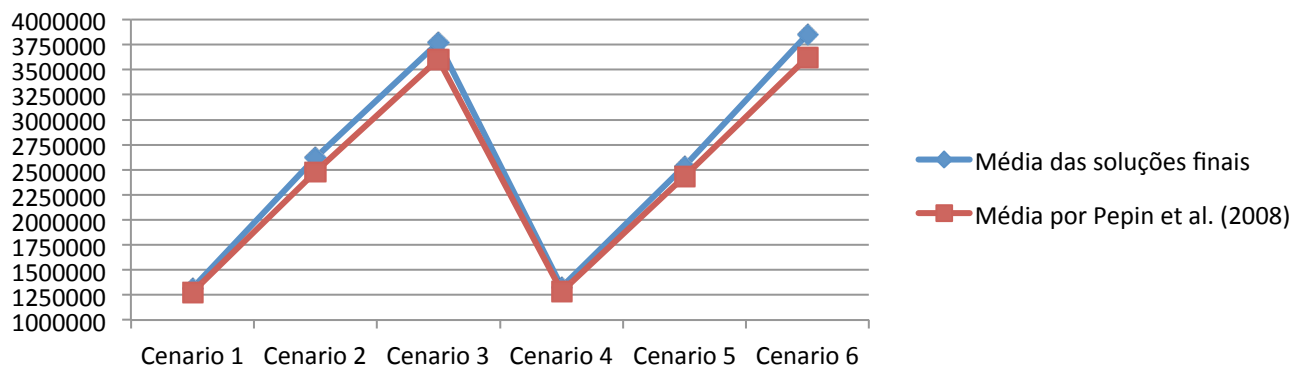
Fonte: do Autor.

Conseguiu-se uma redução no número de veículos utilizados na solução encontrada por este trabalho em praticamente a metade das instâncias testadas (16 das 30), não havendo uma predominância em determinadas instâncias.

Em cinco das trinta instâncias apresentadas, conseguiu-se um valor para a função objetivo, inclusive, menor do que os resultados apresentados pelos outros autores referenciados. Os demais resultados apresentam, em sua maioria, pequenas variações entre uma solução e outra, à exceção de alguns casos como na instância 2 para o Cenário 5 (pior caso) onde a diferença chegou a 18,75%.

Os resultados quanto ao valor da função objetivo ficam mais evidentes se analisarmos o Gráfico 1 apresentado a seguir. Nele são evidenciadas as médias para cada uma das instâncias do MDVSP resolvidas tanto por este trabalho como pelo trabalho de Pepin *et al.* (2008).

Gráfico 1 – Médias das soluções encontradas



Fonte: do Autor.

Nota-se que para os cenários 1 e 4, ou seja, aqueles em que as instâncias apresentam menor tamanho no que tange o número de viagens, os resultados obtidos neste trabalho chegaram bastante perto daqueles apresentados por Pepin *et al.* (2008).

Conforme o número de viagens aumenta, os resultados se distanciam dos apresentados por aqueles autores. Isso demonstra que, apesar de ambos os procedimentos estarem alinhados, o aumento do número de viagens (maior complexidade) faz com que a heurística apresentada retorne valores um pouco superiores aos encontrados pelos outros autores, justamente porque aqueles utilizaram métodos de solução mais robustos.

6 CONCLUSÕES

6.1 SÍNTESE DAS CONCLUSÕES

A dificuldade de se resolver um problema da classe NP-*hard* como o Problema de Escalonamento de Veículos com Múltiplas Garagens (MDVSP) requer estratégias de solução baseadas em heurísticas para a maioria das instâncias de tamanho real. Recentemente, algumas meta-heurísticas estão sendo discutidas para a resolução deste tipo específico de problema, e foi o que este trabalho se propôs a apresentar, utilizando o método de busca-tabu.

Trabalhos abordando o MDVSP através de métodos de resolução exatos ou heurísticos têm sido encontrados na literatura, como os de Rohde (2008) e Hadjar, Marcotte e Soumis (2006). A utilização da meta-heurística de busca-tabu para a resolução deste problema, por sua vez, foi utilizada pela primeira vez por Pepin *et al.* (2008).

Foi justamente essa lacuna observada entre os trabalhos de Rohde (2008) – o qual resolveu o problema através do algoritmo de *branch-and-bound* dando como uma das sugestões de pesquisas futuras sua resolução via procedimentos meta-heurísticos –, e de Pepin *et al.* (2008) – o qual resolveu o MDVSP utilizando pela primeira vez utilizando Busca-Tabu – que motivou o desenvolvimento deste trabalho, a fim de que resultados utilizando o método de busca-tabu também possam apresentar resultados igualmente competitivos quando comparados a outros métodos.

Como ponto bastante positivo deste trabalho encontra-se o algoritmo de geração de soluções iniciais, o qual apresenta excelentes resultados antes mesmo da aplicação do procedimento tabu, em um tempo computacional muito pequeno.

Ao mesmo tempo, com o procedimento tabu criado através de dois outros algoritmos, foi possível diminuir ainda mais os custos totais do sistema, sem que fossem desrespeitadas quaisquer das restrições originais da formulação utilizada para modelagem do problema.

Somando-se a esses pontos, outra grande vantagem no trabalho apresentado foi a possibilidade de fechamento de algumas garagens (ou

depósitos) igualmente sem penalizar os custos da função objetivo e sem necessidade de relaxação de nenhuma das restrições (via algoritmo *depot-exchange*).

O método implementado mostrou-se bastante rápido quando comparado ao trabalho de Pepin *et al.* (2008) e seus resultados apresentam, em muitos casos, pequenas variações nos custos totais. Isso acontece, provavelmente, pelo fato de que foram utilizadas, nesta dissertação, técnicas de alocação de memória otimizadas e uma linguagem de baixo nível.

Além disso, os autores acima utilizaram tempos computacionais pré-definidos para as rodadas utilizando a heurística de busca-tabu. Conclui-se que tal prática onerou aquele trabalho em termos de custos computacionais, uma vez que prova-se que é possível encontrar-se soluções para o MDVSP através de heurísticas utilizando busca-tabu com um tempo bastante inferior.

Isto torna o desenvolvimento deste método extremamente significativo em termos práticos. A flexibilidade deste em se adaptar a outras aplicações do mundo real, faz com que sua utilidade extrapole os limites deste trabalho e possa ser introduzido em novos contextos acadêmicos e empresariais.

Desta forma, cada vez mais seus algoritmos podem ser reavaliados e melhorados, indo ao encontro do trabalho de Cordeau *et al.* (2002), onde discutia que a pesquisa utilizando métodos flexíveis e de fácil implementação, porém efetivos, envolvendo meta-heurísticas estaria se configurando como um futuro promissor em problemas de escalonamento.

6.2 LIMITAÇÕES

Mesmo apresentando tempos computacionais para execução da heurística bastante abaixo daqueles apresentados por Pepin *et al.* (2008), os resultados das funções objetivo, em sua maioria, foram penalizados.

Acredita-se que um refinamento da heurística proposta seja capaz de conseguir resultados ainda mais competitivos para a solução desta classe de problemas.

A possibilidade de geração de viagens impossíveis também apresenta-se como limitação deste trabalho, uma vez que sempre que isto acontecia era necessário re-executar o programa. Esta deficiência pode ser corrigida em trabalhos futuros, de acordo com o que segue.

6.3 SUGESTÕES PARA TRABALHOS FUTUROS

Sugere-se para realização de trabalhos futuros um refinamento no método heurístico proposto, como por exemplo, diminuindo a possibilidade de geração de viagens impossíveis, uma vez que neste trabalho há a necessidade de re-execução do programa a cada vez que isso acontece.

Outra sugestão refere-se à implementação do método via processamento paralelo ou métodos híbridos de resolução (utilizando diferentes heurísticas e métodos exatos), na tentativa de diminuição dos custos, mesmo que, em casos raros, um tempo computacional um pouco superior seja necessário.

Além disso, a utilização desta heurística na formulação e nas instâncias geradas por Rohde (2008), trabalho o qual permitia que veículos voltassem a garagens distintas (relaxação de uma das restrições), também seria um excelente oportunidade a fim de comparar ambos os procedimentos heurísticos.

Por fim, como mais uma sugestão de pesquisa futura, indica-se a possibilidade deste procedimento heurístico adaptar-se ao estudo de Li, Borenstein e Mirchandani (2008), o qual trata do problema de realocação de veículos no contexto do recolhimento de lixo urbano na cidade de Porto Alegre. O estudo apresenta resultados importantes, mas sem a utilização de um procedimento heurístico para a formulação proposta, o que impede sua aplicação em outros contextos mais complexos.

7 REFERÊNCIAS

ALLAHVERDI, A., NG, C. T., CHENG, T. C. E.; KOVALYOV, M. Y. A survey of scheduling problems with setup times or costs, **European Journal of Operational Research** 187: 985–1032, 2008.

ARCHETTI, C., SPERANZA, M. G.;HERTZ, A. A tabu search algorithm for the split delivery vehicle routing problem, **Transportation Science** 40: 64–73, 2006.

ANDRADE, E L. **Introdução a Pesquisa Operacional – Métodos e modelos para a análise de decisão**. Rio de Janeiro: LTC, 1998.

ARENALES, M. *et al.* **Pesquisa Operacional**. Rio de Janeiro: Elsevier, 2007.

BAITA, F; PESENTI, R.; UKOWICH, K.; FAVARETTO, D. A Comparison of Different Solution Approaches do the Vehicle Scheduling Problem in a Pratical Case. **Computers & Operations Research**. v. 27, p. 1249-1269, 2000.

BARNHART *et al.* Branch-and-Price: Column Generation for Solving Huge Integer Programs. **Systems Engineering** p. 1-33, 1996.

BERTOSSI, A. A., CARRARESI, P.; GALLO, G. On some matching problems arising in vehicle scheduling models. **Networks** n.17, p. 271–281, 1987.

BIANCO, L., MINGOZZI, A., RICCIARDELLI, S. A set partitioning approach to the multiple depot vehicle scheduling problem. **Optimization Methods and Software** n.3 p. 163-194, 1994.

BODIN, L.D., GOLDEN, B., ASSAD, A., BALL, M.O., Routing and scheduling of vehicles and crews: the state of the art. **Computers and Operations Research** n. 10, p. 63–211, 1983.

BOFFEY, T.B. **Graph Theory in Operations Research**. Hong Kong: Macmillan, 1984.

BRÄYSY, O.; GENDREAU, M. Vehicle Routing Problem with Time Windows, Part II: Metaheuristics. **Transportation Science**, v. 39, n. 1, p. 119-139, 2005.

BRÄYSY, O.; DULLAERT, W.; GENDREAU, M. Evolutionary algorithms for the vehicle routing problem with time windows. **Journal of Heuristics**, v. 10, n. 6, p. 587-611, 2004.

CARPANETO, G; DELL'AMICO, M; FISCHETTI, M; TOTH P. A branch and bound algorithm for the multiple depot vehicle scheduling problem. **Networks**, n. 19, p.531-548, 1989.

COHEN, M.A., LEE, H.L.. Resource deployment analysis of global manufacturing and distribution networks. **Journal of Manufacturing and Operations Management**. n.2, p71-104, 1989.

_____. Strategic analysis of integrated production-distribution systems: Models and Methods. **Operations Research**. n6, p216-228, 1988.

CORDEAU, J. F., GENDREAU, M., LAPORTE, G., POTVIN, J. Y.; SEMET, F. A guide to vehicle routing problem, *Journal of the Operational Research Society* 53: 512–522, 2002.

CORDEAU, J.; LAPORTE, G.; MERCIER, A. A Unified Tabu-Search Heuristic for Vehicle Routing Problems with Time Windows. **Journal of the Operational Research Society**. v. 52, p. 928 – 936, 2001.

DESAULNIERS, G.; DESROSIERS, J.; SOLOMON, M. **Column Generation**. New York: Springer, 2005.

DESROSIERS, J., DUMAS, Y., SOLOMON, M. M., SOUMIS, F. (1995). Time constrained routing and scheduling. In M. O. Ball, T. L. Magnanti, C. L. Monma, & G. L. Nemhauser (Eds.), **Handbooks in operations research and management science**: Vol. 8. Network routing (pp. 35–139). Amsterdam: Elsevier Science.

EKSIOGLU, B.; VURAL, A.; REISMAN, A. The Vehicle Routing Problem: a Taxonomic Review. **Computers and Industrial Engineering**, n. 57, 2009, p. 1472 – 1483.

FRELING R; WAGELMANS A P; PAIXÃO J M. Models and algorithms for single-depot vehicle scheduling. **Transportation Science**, v.3, n.2, p. 165–180, 2001.

GAITHER, N.; FRAZIER, G. **Administração da Produção e Operações**. 8. ed. São Paulo: Thomson, 2002.

GENDREAU, M., GUERTIN, F., POTVIN, J.-Y., TAILLARD, E. D. Parallel tabu search for real-time vehicle routing and dispatching, **Transportation Science** 33: 381–390, 1999.

GENDREAU, M., HERTZ, A., LAPORTE, G. A tabu search heuristic for the vehicle routing problem, **Management Science** 40: 1276–1290, 1994.

GENDREAU, M., LAPORTE, G. and S'ÉGUIN, R. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers, **Operations Research** 44: 469–477, 1996.

GILMORE, P; GOMORY, R. A Linear Programming Approach to the Cutting-Stock Problem. **Operations Research**, v. 9, n. 6, p. 849-859, 1961.

GINTNER, V.; KLIEWER, N.; SUHL, L. Solving Large Multiple-Depot Multiple-Vehicle-Type Bus Scheduling Problems in Practice. **OR Spectrum**, n. 27, p. 507-523, 2005.

GLOVER, F. Tabu search: Part i, **ORSA Journal of Computing** 1: 190–206, 1989.

GOLDBARG, M. C. e LUNA, H. P. L. **Otimização Combinatória e Programação Linear: Modelos e Algoritmos**. Campus, 2005.

HADJAR, A.; MARCOTTE, O.; SOUMIS, F. A Branch-and-Cut Algorithm for the Multiple Depot Vehicle Scheduling Problem. **Operations Research**, v. 54, n. 1, p. 130-149, 2006.

HAGHANI, A., BANIHASHEMI, M.; CHIANG, K. A Comparative Analysis of Bus Transit Vehicle Scheduling Models. **Transportation Research Part B**, v. 37, n. 4, p. 301-322, 2003.

HUISMAN, D.; FRELING, R.; WAGELMANS, A. A Robust Solution Approach to the Dynamic Vehicle Scheduling Problem. **Transportation Science**, v. 38, n. 4, p. 447 – 458, 2004.

ICHOUA, S.; GENDREAU, M.; POTVIN, J. Diversion Issues in Real-Time Vehicle Dispatching. **Transportation Science**, v. 34, n. 4, p. 426 – 438, 2000.

JANG, Y.J., JANG, S.Y., CHANG, B.M., PARK, J. A combined model of network design and production/distribution planning for a supply network. **Computers & Industrial Engineering**. v. 43, p269-281, 2002.

LAMATSCH, A., An approach to vehicle scheduling with depot capacity constraints. Em: **Proceedings of the 5th International Workshop on Computer-aided Scheduling of Public Transit**, Montreal, Canada, 1990.

LI J Q; BORENSTEIN D; MIRCHANDANI P B. Truck schedule recovery for solid waste collection in Porto Alegre, Brazil. **International Transactions in Operational Research**, 15. p. 565-562, 2008.

LI J Q; MIRCHANDANI P B; BORENSTEIN D. Parallel auction algorithm for bus rescheduling. Apresentado em: **9th International Conference on Computer-Aided Scheduling of Public Transport**. San Diego, California, USA, 2004.

LÖBEL, A. **Optimal Vehicle Scheduling in Public Transit**. Tese de Ph.D, Technische Universität Berlin, Berlin, Germany, 1997.

_____. Vehicle Scheduling in Public Transit and Lagrangian Pricing. **Management Science** 44, p1637–1649, 1998.

LOESCH, C.; HEIN, N. **Pesquisa operacional: fundamentos e modelos**. Blumenau: Ed. da FURB, 1999.

MESQUITA, M., PAIXÃO, J. Multiple depot vehicle scheduling problem: a new heuristic based on quasi-assignment algorithms. Em: M. Desrochers & J.-M. Rousseau (Eds.), *Lecture notes in economics and mathematical systems: Vol. 386. Computer-aided transit scheduling* (pp. 167–180). Berlin: Springer, 1992.

MESQUITA, M., PAIXÃO, J. Exact algorithms for the multiple-depot vehicle scheduling problem based on multicommodity network flow type formulations. **Computer-Aided Transit Scheduling**. Springer Verlag, Berlin, Alemanha, p.223–246, 1999.

PEPIN, A.; DESAULNIERS, G.; HERTZ, A.; HUISMAN, D. A Comparison of five heuristics for the multiple depot vehicle scheduling problem. **Journal of Scheduling**. n. 12, 2008; p. 17-30.

POTVIN, J. State-of-the Art Review – Evolutionary Algorithms for Vehicle Routing. **Journal on Computing**. v.21, n.4 , p. 518-548, 2009.

- RIBEIRO, C.; SOUMIS, F. A Column Generation Approach to the multiple Depot Vehicle Scheduling Problem. **Operations Research**, v. 42, p. 41-52, 1994.
- ROHDE, L. R. **Desenvolvimento de Heurística para Solução do Problema de Escalonamento de Veículos com Múltiplas Garagens**. 2008. 122f. Tese (Doutorado em Administração) – Escola de Administração, UFRGS, Porto Alegre, 2008.
- SANTOS, H. G., OCHI, L. S.; SOUZA, M. J. F. A tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem, **ACM Journal of Experimental Algorithmics** 10: art–2.09. 15 p., 2005.
- SANTOS, G. **Método de Geração de Colunas e Meta-heurísticas para Alocação de Tripulação**. 2008. 80f. Tese (Doutorado em Ciência da Computação) – Programa de Pós-Graduação em Computação, UFMG, Belo Horizonte, 2008.
- SIMON, H. A. Prediction and Prescription in Systems Modeling. **OR Forum**, p7-13, 1989.
- SOUMIS, F. **Decomposition and column generation**. Em: Dell’Amico, F. Maffioli, S. Martello, Annotated Bibliographies in Combinatorial Optimization. John Wiley & Sons, 1997.
- SOUZA, M. J. F., OCHI, L. S.; MACULAN, N. A grasp-tabu search algorithm for solving school timetabling problems, in M. G. C. Resende, J. P. Souza (eds), **Metaheuristics: Computer Decision-Making**, Vol. 1153, Kluwer Academic Publishers, pp. 659–672, 2004.
- TRAGANTALERNGSAK, S.; HOLT, J.; RÖNNQVIST, M. Lagrangian heuristics for the two-echelon, single-source, capacitated facility location problem. **European Journal of Operational Research**. n. 102, 1997, p. 611-625.
- WAGNER, H. M. **Pesquisa Operacional**. Rio de Janeiro: Prentice-Hall, 1986.
- WINSTON, L. **Operations Research, Applications and Algorithm**. 3 ed. Belmont: Duxbury Press, 1994.