

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ADMINISTRAÇÃO  
MESTRADO EM ADMINISTRAÇÃO

William Prigol Lopes

**Desenvolvimento de uma Heurística  
para o Problema de Escalonamento de  
Veículos com Múltiplas Garagens**

Porto Alegre  
2013

### CIP - Catalogação na Publicação

Prigol Lopes, William

Desenvolvimento de uma Heurística para o Problema de Escalonamento de Veículos com Múltiplas Garagens / William Prigol Lopes. -- 2013.

65 f.

Orientador: Denis Borenstein.

Dissertação (Mestrado) -- Universidade Federal do Rio Grande do Sul, Escola de Administração, Programa de Pós-Graduação em Administração, Porto Alegre, BR-RS, 2013.

1. Escalonamento de Veículos. 2. Geração de Colunas. 3. Pesquisa Operacional. I. Borenstein, Denis, orient. II. Título.

William Prigol Lopes

**Desenvolvimento de uma Heurística  
para o Problema de Escalonamento de  
Veículos com Múltiplas Garagens**

Dissertação apresentada como requisito para a  
obtenção do grau de Mestre em Administração  
pelo Programa de Pós-Graduação em  
Administração da Universidade Federal do Rio  
Grande do Sul.

Orientador:  
Prof. Dr. Denis Borenstein

Porto Alegre  
2013

## AGRADECIMENTOS

Agradeço aos meus pais - meus primeiros educadores. Serei grato eternamente.

À minha namorada Jessica Nunes, pela paciência e compreensão nos dias e noites em que não pude estar presente.

Ao meu professor Denis Borenstein pela condução excepcional dos trabalhos e pelos ensinamentos, sem estes ensinamentos seria muito mais difícil ter chegado aos objetivos deste trabalho e almejar novos rumos.

Ao meu colega Camilo José Bornia Poulsen, não apenas pela mútua ajuda, mas pela grande amizade conquistada durante o mestrado. À colega Monize Vissentini pelas tardes no *Skype* tentando solucionar a geração de colunas. Ao colega Pablo Guedes pelo trabalho em conjunto para resolver a geração de colunas, e por fornecer instâncias para os experimentos computacionais. Ao Professor Cândido Fonseca da Silva, seu ensinamentos serão levados por toda a vida.

Ao professor Olinto, pelo apoio com grandes ideias para aceleração da resolução, bem como pelo apoio no desenvolvimento do software. Ao professor Jing-Quan Li, pelo apoio para resolver os problemas VSP de forma rápida.

Aos colegas, professores e funcionários do PPGA que, durante todo o tempo forneceram um grande suporte para o aprendizado. Aos colegas da Solis e da Univates, pela compreensão em todo o período do meu mestrado.

Aos meus tios e tias, meus segundos pais.

À Deus.

*"If I have seen farther than others,  
it is because I stood on the shoulders of giants."*  
— SIR ISAAC NEWTON

## RESUMO

O escalonamento de veículos para múltiplas garagens é um problema clássico da Pesquisa Operacional de grande relevância à otimização da malha de transportes, de forma a buscar a melhor alocação dos recursos disponíveis. O problema é conhecido por ser do tipo NP-*hard*, e portanto a sua solução para grandes instâncias é realizada por heurísticas. Este trabalho aborda o desenvolvimento de uma heurística simples e eficiente para solucionar o MDVSP, buscando através da redução de possibilidades de viagens, pela retirada de viagens com baixas chances de pertencer a solução ótima reduzir a complexidade da rede. O problema reduzido, é então resolvido pela técnica de geração de colunas truncado e modificado, para resolver o problema de forma aceitável, encontrando soluções com bom compromisso entre tempo de execução e valor da função objetivo. A geração de colunas foi testada e validada através de comparações com trabalhos similares, enquanto que, cada técnica de redução de espaço de estados (de forma conjunta e individual) foram validados através da comparação com os resultados validados da geração de colunas modificada. A heurística mostrou uma melhoria considerável por meio da otimização do tempo de resolução, sem prejudicar os resultados de melhor valor, ficando com uma diferença máxima de 1% em comparação com os valores obtidos com geração de colunas. Se a redução for usada parcialmente, os tempos de solução podem ser reduzidos mais de sete vezes, com um pequeno incremento no valor da função objetivo. Pelas validações e experimentos realizados, pode-se afirmar que a heurística tem potencial para ser utilizada em problemas do mundo real, bem como servir como parte da solução de problemas correlatos mais complexos, como o *crew scheduling*, o *disruption management* e o escalonamento em tempo real.

## ABSTRACT

The Multi Depot Vehicle Scheduling is a classical problem of Operations Research, with a great relevance to optimize the transportation network, as to find the best allocation of available resources. This problem is known to be NP-Hard, and therefore the solution of large instances is generally carried out by heuristics. This work proposes the development of a simple and efficient heuristic to solve the MDVSP, seeking by the reduction of travel possibilities, dropping travels with low chances to be in the best solution. The reduced problem is then solved by a modified truncated column generation technique, to solve the problem by an acceptable form, founding solutions for the MDVSP with a good compromise between time and costs. The column generation was tested and validated by the comparison of execution time and the value of objective function. The column generation was tested and compared with similar works, whereas, each space state reduction technique (individual and joint) was validated in comparison with validated results of column generation truncated and modified. The heuristics showed considerable improvements by the optimization of resolution time, without harm the best value results, obtaining differences lower as 1% comparing with the obtained values by the column generation solution. If the reduction is used partially, the solution times can be reduced in more than seven times, with a small increment in the objective function value. By the validations and experiments realized, we can confirm that the heuristics has potential to be used in real world situations, as well as to serve as basis for the development of solution methods for more complex and correlated problems, as the crew scheduling problem, disruption management, and the real time assignment.

## LISTA DE ABREVIATURAS E SIGLAS

GC	Geração de Colunas
JV	Algoritmo de <i>Jonker Volgenant</i>
MDVSP	<i>Multi-Depot Vehicle Scheduling Problem</i> ou Problema de Escalonamento de Veículos com Múltiplas Garagens
PO	Pesquisa Operacional
RMP	<i>Restrictive Master Problem</i> ou Problema Mestre Restritivo
VSP	<i>Vehicle Scheduling Problem</i> ou Problema de Escalonamento de Veículos



## LISTA DE SÍMBOLOS

$N = \{1, 2, \dots, n\}$	Conjunto de viagens
$b_i$	Local de saída da viagem $i$
$e_i$	Local de chegada à viagem $i$
$bt_i$	Horário de saída da viagem $i$
$et_i$	Horário de chegada à viagem $i$
$n$	Número de viagens
$d$	Garagem
$s$	Pontos de saída das garagem $d$
$t$	Pontos de chegada das garagem $d$
$c_{ij}$	Custos de <i>deadheading</i> da viagem $i$ , para a viagem $j$
$k$	Número de garagens
$K = \{1, 2, \dots, k\}$	Conjunto de garagens de tamanho $k$
$s_k$	Pontos de saída da garagem $k$
$t_k$	Pontos de chegada das garagem $K$
$\Omega = \{1, 2, \dots, p\}$	Conjunto de blocos possíveis de serem realizados por um veículo

# SUMÁRIO

<b>LISTA DE FIGURAS</b> . . . . .	11
<b>LISTA DE QUADROS</b> . . . . .	12
<b>1 INTRODUÇÃO</b> . . . . .	13
<b>2 REVISÃO DA LITERATURA</b> . . . . .	16
2.1 O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS (VSP) . . . . .	16
2.1.1 <b>Formulação matemática para o VSP</b> . . . . .	17
2.2 O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS PARA MÚLTIPLAS GARAGENS (MDVSP) . . . . .	19
2.2.1 <b>Formulação matemática do tipo <i>commodity</i> único para o MDVSP</b> . . . . .	20
2.2.2 <b>Formulação matemática do tipo <i>multicommodity</i> para o MDVSP</b> . . . . .	21
2.2.3 <b>Formulação matemática do tipo conjunto de partições para o MDVSP</b> . . . . .	23
2.3 SOLUÇÕES PROPOSTAS . . . . .	24
2.3.1 <b>Redução de espaço de estados</b> . . . . .	27
2.3.2 <b>Geração de colunas truncado para o MDVSP</b> . . . . .	29
<b>3 PROCEDIMENTOS METODOLÓGICOS</b> . . . . .	34
<b>4 MÉTODO DE SOLUÇÃO DESENVOLVIDO</b> . . . . .	36
4.1 <b>DEFINIÇÃO DA REDE</b> . . . . .	36
4.2 <b>REDUÇÃO DO ESPAÇO DE ESTADOS</b> . . . . .	38
4.2.1 <b>Redução por escolha de melhores caminhos pelo VSP</b> . . . . .	39
4.2.2 <b>Redução pelo critério estatístico</b> . . . . .	42
4.3 <b>GERAÇÃO DE COLUNAS TRUNCADO</b> . . . . .	43
<b>5 EXPERIMENTOS COMPUTACIONAIS</b> . . . . .	46
5.1 <b>CALIBRAGENS DA REDUÇÃO DO ESPAÇO DE ESTADOS</b> . . . . .	46
5.2 <b>CALIBRAGENS DA GERAÇÃO DE COLUNAS TRUNCADO</b> . . . . .	47
5.3 <b>VERIFICAÇÃO DA IMPLEMENTAÇÃO DA GERAÇÃO DE COLUNAS TRUNCADO</b> . . . . .	48
5.4 <b>VALIDAÇÃO DA HEURÍSTICA</b> . . . . .	50
5.5 <b>EXPERIMENTOS COMPUTACIONAIS</b> . . . . .	55
5.6 <b>DISCUSSÕES</b> . . . . .	57
<b>6 CONCLUSÃO</b> . . . . .	60
<b>REFERÊNCIAS</b> . . . . .	62

## LISTA DE FIGURAS

Figura 1	Exemplo de rede de arcos de conexão . . . . .	37
Figura 2	Processo de interpretação da redução do espaço de estados . . . . .	39
Figura 3	Redução por escolha de melhores caminhos do VSP . . . . .	41
Figura 4	Redução por escolha de melhores valores . . . . .	42
Figura 5	Algoritmo de geração de colunas truncado com modificações . . . . .	44
Figura 6	Convergência entre GC truncado e GC com relaxamento . . . . .	45
Figura 7	Comparativo de tempos entre as resoluções . . . . .	58
Figura 8	Comparativo de melhores valores entre as resoluções . . . . .	59

## LISTA DE QUADROS

Tabela 1	Parâmetros envolvidos no VSP . . . . .	18
Tabela 2	Parâmetros envolvidos no MDVSP . . . . .	19
Tabela 3	Relação de trabalhos envolvendo o MDVSP e seus métodos de solução	25
Tabela 4	Comparativo de média de tempo de resolução do MDVSP . . . . .	49
Tabela 5	Comparativo de melhores valores para a geração de colunas . . . . .	50
Tabela 6	Comparativo de melhores valores entre geração de colunas com e sem reduções . . . . .	52
Tabela 7	Comparativo de tempo entre geração de colunas com e sem reduções	53
Tabela 8	Comparativo de valores de geração de colunas e heurística . . . . .	54
Tabela 9	Experimentos computacionais com a heurística . . . . .	56

# 1 INTRODUÇÃO

Com a crescente urbanização e o acesso facilitado aos meios de transporte, sejam particulares ou coletivos, novas necessidades são impostas constantemente a governos e empresas. A melhoria da eficiência na utilização dos recursos de transporte disponíveis, é considerado como uma grande necessidade, exigindo planejamento e replanejamento contínuo, afim de evitar atrasos, transtornos e até possíveis colapsos.

A provisão de produtos e serviços por meio de pontos de distribuição é de grande importância econômica e social, sendo que, a questão mais importante é de como criar e adotar um sistema de distribuição eficiente (BOFFEY , 1984). Se o problema for analisado pela questão de custos, pode-se considerar a otimização de custos operacionais, custos fixos dos veículos, custos de espera, custos de ociosidade e custos de cancelamento como elementos de extrema relevância econômica (ROHDE, 2008).

Como exemplificação da constante necessidade de otimização de sistemas de transporte, podemos citar o transporte público e o transporte em massa, citando todos os tipos de malhas, como aéreas e viárias. No Brasil, há claras evidências da necessidade de um replanejamento do transporte, desde o aéreo até o urbano, não só para grandes eventos que ocorrerão, a exemplo da Copa do Mundo e das Olimpíadas, como também, pelos fatos constantes noticiados corriqueiramente, como engarrafamentos ou greves nos sistemas de transporte em massa. Falhas e colapsos no transporte são evidenciados justamente, pela falta de recursos ou de planejamento, levando à necessidade de revisão, para aplicação de melhorias e consequente, diminuição de custos, facilitando assim, a administração dos recursos envolvidos.

Em sistemas de transporte, a percepção das inter-relações são visíveis por itens como, distribuição geográfica das garagens, quantidade de veículos disponíveis, agen-

das de início e fim de transportes e o planejamento das rotas que devem ocorrer.

Entre os vários tipos de problemas de transportes pesquisados, um dos mais típicos é conhecido como *Multi Depot Vehicle Scheduling Problem* (MDVSP) ou Problema de Escalonamento de Veículos com Múltiplas Garagens (BERTOSSO; CARRARRESI; GALLO, 1987) que, tem como objetivo, escalonar vários veículos para percorrer um conjunto de viagens programadas, contendo várias garagens e, cada veículo saindo e chegando de uma garagem, respeitando suas capacidades, buscando assim, otimizar a eficiência do transporte (BOFFEY, 1984; KLIEWER; MELLOULI; SUHL, 2005).

Contudo, o MDVSP pode ser considerado um problema de alta complexidade que, se aplicados em certas situações do mundo real, os tempos computacionais são considerados impraticáveis ou, até mesmo, inviáveis, (LAURENT; HAO, 2009; GINTNER; KLIEWER; SUHL, 2005), sendo classificado como um problema do tipo NP-*Hard*, ao qual, para sua solução ou otimização, normalmente é necessária a utilização de métodos conhecidos como heurísticas (BERTOSSO; CARRARRESI; GALLO, 1987). Embora já existam vários métodos de solução na literatura para o MDVSP (PEPIN; DESAULNIERS; HERTZ, 2008), instâncias com um elevado número de garagens e viagens são ainda resolvidos com grande dificuldade, dificultando a utilização prática por empresas de transporte e logísticas.

Portanto, o objetivo geral deste trabalho é solucionar o MDVSP através do desenvolvimento de uma heurística, de forma a auxiliar a decisão das organizações na estruturação e planejamento de suas redes de transporte. O algoritmo combina técnicas para a redução de viagens, reduzindo arcos na rede de conexão adjacente ao problema, e um algoritmo modificado de geração de colunas para oferecer muito boas soluções em tempos aceitáveis.

Como objetivos específicos da presente dissertação, tem-se:

- Apresentar métodos encontrados na literatura para a solução do MDVSP;
- Aplicar e adaptar soluções para o MDVSP, utilizando técnicas de redução de espaço de estados e de geração de colunas;
- Avaliar a solução encontrada, comparando resultados com os estudos de PEPIN; DESAULNIERS; HERTZ (2008); e

- Avaliar o desempenho da heurística para instâncias de tamanho superior.

Este trabalho está organizado como se segue: No capítulo 2, é apresentada a revisão da literatura, abordando sobre o problema de escalonamento de veículos ou *Vehicle Scheduling problem* (VSP), sobre o MDVSP, sobre as heurísticas de redução de espaço de estados e geração de colunas truncado. No capítulo 3 é abordado sobre os procedimentos metodológicos. Em seguida, no capítulo 4, é apresentado sobre o método de solução desenvolvido, no capítulo 5 são apresentados experimentos computacionais executados com a heurística desenvolvida e, por último, no capítulo 6 são apresentadas as conclusões da presente dissertação.

## 2 REVISÃO DA LITERATURA

Este capítulo aborda a revisão da literatura acerca do tema discutido neste estudo, sendo organizado da seguinte forma: Na seção 2.1, é apresentado o problema de escalonamento de veículos ou *Vehicle Scheduling Problem* (VSP), a seguir, na seção 2.2 é abordado o problema de escalonamento de veículos com múltiplas garagens ou *Multi depot Vehicle Scheduling Problem* (MDVSP) e, por último, na seção 2.3 são abordadas as soluções propostas para a resolução do MDVSP.

### 2.1 O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS (VSP)

O propósito do VSP é minimizar o custo de veículos e de diferentes custos operacionais, designando um conjunto de rotas previamente agendadas para um determinado conjunto de veículos, de modo que, cada viagem é associada a um veículo.

Considerado como um problema clássico de otimização de transporte (BAITA *et al.*, 2000), tem o objetivo de minimizar os custos de transportes envolvendo um conjunto de viagens com tempos, locais de partida e chegada pré-definidos, bem como os tempos de viagem entre todos os pontos de partida e chegada. O VSP busca encontrar uma designação de viagens, devendo respeitar os seguintes itens (BUNTE; KLIEWER, 2010; FRELING; WAGELMANS; PAIXÃO, 2001):

- Cada viagem deve ser coberta apenas uma vez; e
- Cada veículo execute uma sequência viável de viagens;

A formulação para a solução do problema deve obedecer a uma série de restrições como: capacidade individual dos veículos, distância percorrida de cada veículo e janelas de tempo específicas de cada viagem. Várias abordagens para o problema



foram desenvolvidas, comprovando que o problema pode ser resolvido em tempo polinomial (BUNTE; KLIEWER, 2010; AUGUSTINE; SEIDEN, 2004).

As designações práticas para este tipo de problema devem abordar a inclusão de um conjunto de maior de critérios, como no caso de aplicar o problema envolvendo carros elétricos, com períodos para reabastecimento das baterias, portanto, para permitir que este tipo de problema atenda às demandas de situações reais, deve-se modificar o modelo para que possa atender à variáveis - a exemplo - de tempos e custos de abastecimento (KANG; LEE; LEE, 2005; BAITA *et al.*, 2000).

### 2.1.1 Formulação matemática para o VSP

Por ter sua solução resolvida em tempo polinomial, existem várias formulações matemáticas para a solução do VSP (BUNTE; KLIEWER, 2010), sendo que, as formulações mais comuns encontradas na literatura são:

- (a) Formulação de decomposição mínima (SAHA, 1972);
- (b) Formulação de designação (ORLOFF, 1972);
- (c) Formulação de transporte (GAVISH; SHLIFER, 1979); e
- (d) Formulação de fluxo de rede (BODIN *et al.*, 1983).

A formulação de fluxo de rede está relacionada com o presente trabalho, ela é abordada a seguir. Para as outras formulações, sugere-se consultar as referências citadas. A Tabela 1 indica os parâmetros utilizados na estruturação da formulação VSP para fluxo de rede, apresetando o parâmetro (coluna 1) e um descritivo do que o parâmetro representa no problema (coluna 2).

Podemos indicar que dois pontos  $i$  e  $j$  são compatíveis quando um mesmo veículo tem a possibilidade de passar pelos dois pontos em sequência. Portanto, considera-se uma sequência de viagens viável aquela que, cada par consecutivo de viagens seja compatível.

Considera-se também que os nós  $s$  e  $t$  representam, em ambos, a garagem localizada em  $d$ . Assim, pode-se definir uma rede do tipo VSP como  $G = \{V, A\}$  como uma rede dirigida acíclica com nós  $V = N \cup \{s, t\}$  e arcos  $A = E \cup (s \times N) \cup (N \times t)$ .

Um caminho de  $s$  para  $t$  na rede apresenta um escalonamento viável de um veículo e a possibilidade de um escalonamento completo é um grupo de caminhos

Tabela 1 – Parâmetros envolvidos no VSP

Parâmetro	Descrição
$N = \{1, 2, \dots, n\}$	Conjunto de viagens
$b_i$	Local de saída da viagem $i$
$e_i$	Local de chegada à viagem $i$
$bt_i$	Horário de saída da viagem $i$
$et_i$	Horário de chegada à viagem $i$
$n$	Número de viagens
$d$	Garagem
$s$	Pontos de saída das garagem $d$
$t$	Pontos de chegada das garagem $d$
$c_{ij}$	Custos de <i>deadheading</i> da viagem $i$ , para a viagem $j$

disjuntos de  $s$  para  $t$ , tal que, cada nó em  $N$  seja coberto pela solução. Também, consideramos a utilização da variável binária,  $x_{ij} \in \{0, 1\}$ , indicando se um veículo executou ou não uma viagem entre os pontos  $i$  e  $j$ .

Partindo dos pressupostos anteriormente citados, o VSP pode ser formulado matematicamente, conforme descrito a seguir:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (1)$$

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

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

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (4)$$

A função objetivo (1) busca minimizar os custos operacionais das viagens expressas, enquanto as restrições (2) e (3) garantem que cada viagem será atribuída a exatamente uma viagem predecessora e sucessora - garantindo que a rede será particionada em um conjunto de caminhos disjuntos de  $s$  a  $t$  - e, por último, a restrição (4) condiciona as variáveis da formulação como binárias.

Para atender os objetivos do presente estudo, é necessário utilizar uma abordagem que envolva múltiplas garagens, evoluindo a formulação do VSP para a formulação do MDVSP, portanto, a seguir, na seção 2.2, é apresentado sobre o Problema de Escalonamento de Veículos para Múltiplas Garagens.

## 2.2 O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS PARA MÚLTIPLAS GARAGENS (MDVSP)

O MDVSP consiste em buscar a minimização dos custos de transporte, de modo a cobrir um conjunto de viagens com agendas pré-definidas, envolvendo vários veículos e várias garagens (RIBEIRO; SOUMIS, 1994; KLIEWER; MELLOULI; SUHL, 2005).

Para atender ao escalonamento de veículos com Múltiplas garagens, o problema deve obedecer a alguns requisitos (BERTOSSO; CARRARRESI; GALLO, 1987; DELL'AMICO; FISCHETTI; TOTH, 1993), conforme citados a seguir:

- Cada viagem é coberta por exatamente um veículo;
- Cada veículo utilizado na solução cubra um caminho viável; e
- O número de veículos estacionado nas garagens usadas na solução, não excedam a capacidade de veículos que estejam estacionados na mesma garagem.

Obedecendo às premissas listadas acima, sua formulação pode ser utilizada em várias aplicações para auxiliar gestores a buscar uma melhor tomada de decisões. A Tabela 2 apresenta os parâmetros envolvidos no problema MDVSP, indicando o parâmetro (coluna 1) e um descritivo do parâmetro (coluna 2).

**Tabela 2 – Parâmetros envolvidos no MDVSP**

<b>Parâmetro</b>	<b>Descrição</b>
$k$	Número de garagens
$K = \{1, 2, \dots, k\}$	Conjunto de garagens de tamanho $k$
$s_k$	Pontos de saída da garagem $k$
$t_k$	Pontos de chegada das garagem $k$

Formalmente, o problema pode ser definido como se segue. Dado um conjunto  $N$  de agendas de tarefas a serem cumpridas de uma frota de veículos estacionados em um conjunto de garagens  $K$ , é feita a busca pelas agendas de menor custo para os veículos, ao qual, cada tarefa é designada para apenas um veículo, sendo que, o número de veículos disponíveis  $v_k$  em cada garagem  $k \in K$  não pode ser excedido.

Para cada tarefa é definida por um ponto inicial  $s_i$ , e um ponto final  $t_i$ . Também, uma agenda precisa iniciar e finalizar na mesma garagem, devendo ser composta por uma sequência de tarefas ordenadas.

Como o MDVSP contém muitas otimizações combinatórias, o problema é considerado de natureza NP-*Hard*, quando o número de garagens for maior ou igual a 2, por isso, os modelos exatos não conseguem resolver grandes instâncias encontradas no dia-a-dia, justamente pelas numerosas possibilidades de designação de veículos para cada rota (BERTOSSO; CARRARRESI; GALLO, 1987). Caso a instância contenha um número de garagens igual a 1, o modelo deve ser considerado como um problema de fluxo de rede de custo mínimo, sendo resolvido em tempo polinomial. Também, no caso de a função objetivo buscar a minimização do número total de veículos usados para executar  $n$  viagens, o problema também é solucionável em tempo polinomial (RIBEIRO; SOUMIS, 1994).

As formulações para o MDVSP mais comumente encontradas na literatura são:

- (a) Formulação de *commodity* único (CARPANETO *et al.*, 1989; MESQUITA; PAIXÃO, 1992);
- (b) Formulação de *multicommodity* (BERTOSSO; CARRARRESI; GALLO, 1987);  
e
- (c) Formulação de conjunto de partições (RIBEIRO; SOUMIS, 1994).

A seguir, são sucintamente descritas cada uma das formulações, iniciando pela formulação de *commodity* único (seção 2.2.1), tendo destaque para as formulações de *multicommodity* (seção 2.2.2) e de conjunto de partições (seção 2.2.3), por serem foco da presente proposta.

### 2.2.1 Formulação matemática do tipo *commodity* único para o MDVSP

A formulação de *commodity* único propõe uma estrutura de rede similar ao VSP, com uma característica de que cada veículo disponível por garagem é considerado como um nó e, para cada nó criado provindo de um veículo disponível, deve-se ser replicado outro nó com seus respectivos custos, para cada uma das viagens possíveis para o respectivo veículo (CARPANETO *et al.*, 1989).

Para o problema de *commodity* único, podemos definir um conjunto de nós  $V \in N$ , que contém todos os nodos de viagens e um conjunto  $A$  com todos os arcos da rede. Considerando também, para cada caminho  $p$ , dentro do conjunto de possíveis

viagens  $\Omega$ . Partindo destes pressupostos, a formulação de *commodity* único para o MDVSP pode ser ilustrada conforme a seguir (BUNTE; KLIEWER, 2010).

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (5)$$

$$\sum_{j:(i,j) \in A} x_{ij} = 1 \quad \forall i \in V, \quad (6)$$

$$\sum_{i:(i,j) \in A} x_{ij} = 1 \quad \forall i \in V, \quad (7)$$

$$\sum_{(i,j) \in P} x_{ij} \leq |p| - 1 \quad \forall p \in \Omega \quad (8)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (9)$$

$$(10)$$

A função objetivo (6) indica a minimização dos custos totais, enquanto que a restrição (7) indica que cada ponto de saída deve ser visitado apenas uma vez, enquanto que a restrição 8) indica que cada ponto de chegada deve ser visitado apenas uma vez. Já a restrição 9 indica que cada caminho  $p$  dentro do conjunto  $\Omega$  não pode conter mais que uma garagem e, por último na restrição (10), a indicação da variável binária  $x$ .

Devido a inserção de nós para cada veículo e suas possíveis viagens, o número de elementos presente no problema é extremamente alto, aumentando drasticamente o número de elementos a cada adição de garagens, resultando em tempos de resposta impraticáveis para a resolução do problema, se aplicados em problemas reais (BUNTE; KLIEWER, 2010).

### 2.2.2 Formulação matemática do tipo *multicommodity* para o MDVSP

A formulação do tipo *multicommodity* para o MDVSP pode ser definida a partir das seguintes premissas. Considerando uma rede  $G^k = (V^k, A^k)$  para cada garagem  $k \in K$ , ao qual,  $V^k$  e  $A^k$  denota seu nodo e os conjuntos de arcos de cada garagem, respectivamente.

O conjunto  $V^k$  contém um nó para cada tarefa  $i \in T$  e um par de nós,  $o(k)$  e  $d(k)$ , representando o início e o fim de uma agenda para um veículo associada a uma garagem  $k$ , respectivamente. Portanto,  $V^k = \{o(k), d(k)\} \cup T$ .

O conjunto  $A^k$  contém três tipos de arcos, sendo eles, arcos de saída, arcos de entrada e arcos de conexão. Para cada arco do tipo de saída  $(o(k), i)$  para dada tarefa do nó  $i \in N$ , simetricamente, há um arco de entrada  $(i, d(k))$  para cada nó da tarefa  $i \in N$  e, um arco de conexão  $(i, j)$  para cada par de nós das tarefas  $i$  e  $j$  em  $N$ , ao qual,  $et_i + trav(e_i, b_j) \leq bt_j$ , onde  $trav(e_i, b_j)$  é o tempo de *deadheading* do local  $e_i$  para o local  $b_j$ . O custo de um arco  $(i, j) \in A^k$ , denominado  $c_{ij}$ , é igual aos custos de viagem e de espera associados. Se  $i = o(k)$ , então, existe um custo fixo para veículo a ser utilizado.

A formulação proposta envolve as variáveis binárias  $x_{ij}^k$ ,  $(i, j) \in A^k$  e  $k \in K$ .

Com esta notação, a formulação matemática do tipo *multicommodity* para o MDVSP pode ser descrita conforme a seguir (BERTOSSO; CARRARRESI; GALLO, 1987):

$$\min \sum_{k \in K} \sum_{(i,j) \in A^k} c_{ij} x_{ij}^k \quad (11)$$

$$\sum_{k \in K} \sum_{j: (i,j) \in A^k} x_{ij}^k = 1 \quad \forall i \in T \quad (12)$$

$$\sum_{j: (o(k),j) \in A^k} x_{o(k),j}^k \leq v_k \quad \forall k \in K \quad (13)$$

$$\sum_{j: (j,i) \in A^k} x_{ji}^k - \sum_{j: (i,j) \in A^k} x_{ij}^k = 0 \quad \forall i \in V^k \setminus \{o(k), d(k)\}, k \in K \quad (14)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in A^k, k \in K \quad (15)$$

A função objetivo (11) minimiza os custos totais, enquanto que, a restrição (12) garante que cada tarefa seja executada por, exatamente, uma vez e por apenas um veículo. A restrição (13) limita o número de veículos que pode ser utilizado por cada garagem. A restrição (14) define uma estrutura de múltiplos caminhos para cada garagem e, por último, a restrição (15) define as variáveis como binárias.

A formulação *multicommodity* é largamente utilizada como base para aplicação de heurísticas do tipo *branch-and-cut* e relaxamento lagrangeano (RIBEIRO; SOUMIS, 1994; PEPIN; DESAULNIERS; HERTZ, 2008), considerando que é possível resolver problemas de tamanhos maiores que a formulação de *commodity* único.

A seguir, na seção 2.2.3 é apresentada a modelagem matemática do tipo de

conjunto de partições para o MDVSP.

### 2.2.3 Formulação matemática do tipo conjunto de partições para o MDVSP

A formulação matemática do tipo de conjunto de partições para o MDVSP foi criada por RIBEIRO; SOUMIS (1994) e sua equivalência matemática comprovada a partir da formulação de *multicommodity* (formulação descrita na seção 2.2.2).

Para compreender a formulação do tipo conjunto de partições, deve-se considerar a estruturação a seguir.

Com os circuitos dos grafos  $G^k = (V^k, A^k)$  definidos anteriormente, consideramos  $\Omega^k$  como o conjunto das agendas de veículos consideradas viáveis da garagem  $k \in K$ . Para cada agenda  $p \in \Omega^k$ , definimos os seguintes parâmetros: o custo  $c_p$  e, para cada tarefa  $i \in N$ , um parâmetro binário  $a_{ip}$  igual à 1 caso a agenda  $p$  inclua a tarefa  $i$  e, 0, caso contrário. Sendo que, com cada agenda  $p \in \Omega^k$ , definimos uma variável binária  $\theta_p$ , ao qual terá o valor 1 se  $p$  está dentro da solução e, 0, caso contrário (RIBEIRO; SOUMIS, 1994; PEPIN; DESAULNIERS; HERTZ, 2008).

Com estas especificações, a formulação do tipo conjunto de partições pode ser descrito conforme abaixo.

$$\min \sum_{k \in K} \sum_{p \in \Omega^k} c_p \theta_p \quad (16)$$

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip} \theta_p = 1 \quad \forall i \in N \quad (17)$$

$$\sum_{p \in \Omega^k} \theta_p \leq v_k \quad \forall k \in K \quad (18)$$

$$\theta_p \in \{0, 1\} \quad \forall p \in \Omega^k, k \in K \quad (19)$$

A função objetivo (16) busca a minimização dos custos totais. As restrições de conjunto de partições (17) impõem que cada tarefa seja executada exatamente por um veículo. Quando houver diferenças, a restrição (18) expressa a disponibilidade das garagens e, por último, a restrição (19) permite utilizar os elementos como restrições binárias.

A formulação matemática do tipo conjunto de partições foi concebida para servir de base para a aplicação da heurística de geração de colunas (ROHDE, 2008; PEPIN;

DESAULNIERS; HERTZ, 2008), formulada a partir da ideia da decomposição de *Dantzig-Wolfe* (DANTZIG; WOLFE, 1961), podendo, também ser adaptada para ser utilizada com outros modelos de decomposição.

Cada partição gerada deve ser considerado como um subproblema. Cada subproblema deve ser composto pela circulação de uma unidade de fluxo (ou seja, de um veículo) que, obrigatoriamente, deverá sair de uma garagem, passando por um ou mais pontos e, retornando à mesma garagem. Conforme RIBEIRO; SOUMIS (1994), o subproblema é equivalente a um problema de caminho mais curto. Para mais detalhes, favor consultar a seção 2.3.2.

A seguir, na seção 2.3, é apresentado um conjunto de trabalhos anteriores pesquisados durante o desenvolvimento do presente trabalho.

### 2.3 SOLUÇÕES PROPOSTAS

Esta seção tem como objetivo apresentar uma revisão de trabalhos encontrados na literatura acerca dos modelos propostos para solucionar o MDVSP.

Para o levantamento das soluções propostas foram pesquisados artigos envolvendo o problema MDVSP e sua resolução, buscando demonstrar os esforços feitos como tentativas de resolução e otimização do problema.

Para ilustrar o revisão de trabalhos encontrados, a Tabela 3 apresenta uma lista de trabalhos envolvendo o MDVSP estudados durante o levantamento da revisão bibliográfica, ordenado pelo ano da publicação, indicando a referência do autor (coluna 1), o tipo de formulação utilizada (coluna 2) e a(s) heurísticas(s) utilizada(s) para a solução (coluna 3).

Os primeiros trabalhos envolvendo o MDVSP, recorriam ao método de solução em duas fases, primeiramente, resolvendo o problema para uma garagem e, após, designando veículos através do tradicional problema de transportes (BODIN *et al.*, 1983).

A formulação de *commodity* único foi combinada com várias heurísticas: como: aplicação da busca dos caminhos mais curtos para várias garagens (KIM; KIM, 1999); da busca local iterada (LAURENT; HAO, 2009); de caminhos mais curtos combinada com algoritmo exato (KANG; LEE; LEE, 2005).

ROHDE (2008), desenvolveu uma técnica de duas fases, inicialmente, reduzindo



**Tabela 3 – Relação de trabalhos envolvendo o MDVSP e seus métodos de solução**

<b>Publicação</b>	<b>Formulação</b>	<b>Método de solução</b>
BODIN <i>et al.</i> (1983)	<i>Commodity</i> único	SDVSP e Problema de Transporte
BERTOSSI; CARRARRESI; GALLO (1987)	<i>Multicommodity</i>	Relaxamento lagrangeano
CARPANETO <i>et al.</i> (1989)	<i>Multicommodity</i>	<i>Branch-and-bound</i>
LAMATSCH (1990)	<i>Multicommodity</i>	Relaxamento lagrangeano
DELL'AMICO; FISCHETTI; TOTH (1993)	<i>Multicommodity</i>	Caminhos mais curtos
MESQUITA; PAIXÃO (1992)	<i>Multicommodity</i>	Relaxamento lagrangeano
BIANCO; MINGOZZI; RICCIARDELLI (1994)	Conj. de partições	Simplex Dual
RIBEIRO; SOUMIS (1994)	Conj. de partições	Geração de Colunas e <i>Branch-and-cut</i>
DESROSIERS <i>et al.</i> (1995)	Conj. de partições	Geração de colunas
BARNHART <i>et al.</i> (1996)	Conj. de partições	Geração de colunas
DESAULNIERS; LAVIGNE; SOUMIS (1998)	Conj. de partições	Geração de colunas
LÖBEL (1998)	<i>Multicommodity</i>	Geração de colunas / Relaxamento lagrangeano
KIM; KIM (1999)	<i>Commodity</i> único	Caminhos mais curtos
GINTNER; KLIEWER; SUHL (2005)	<i>Multicommodity</i>	Corrigir-e-otimizar
KANG; LEE; LEE (2005)	<i>Commodity</i> único	Caminho mais curto/Algoritmo exato
OUKIL <i>et al.</i> (2006)	Conj. de partições	Geração de colunas
ROHDE (2008)	<i>Commodity</i> único	Redução do espaço de estados
PEPIN; DESAULNIERS; HERTZ (2008)	<i>Multicommodity</i> /Conj. de partições	<i>Branch-and-cut</i> /Geração de colunas/ Relaxamento lagrangeano/ <i>Tabu Search</i> / Procura do vizinho mais longe
LAURENT; HAO (2009)	<i>Commodity</i> único	Busca local iterada
HADJAR; SOUMIS (2009)	<i>Multicommodity</i>	<i>Branch-and-bound</i> Geração de colunas

o espaço de estados (possibilidades de viagens) através de três procedimentos de exclusão de nodos considerados com grandes chances de não pertencer à solução ótima, após, com o conjunto de espaço de estados reduzido, verificava qual era a melhor solução. A redução do espaço de estados diminuiu drasticamente a quantidade de variáveis analisadas, assim, diminuindo a complexidade do problema e, conseqüentemente, otimizando o tempo de resposta computacional.

A criação da formulação de *multicommodity* permitiu diferentes explorações nas estratégias de otimização do problema, sendo mais utilizado com a heurística de relaxamento lagrangeano (BERTOSS; CARRARRESI; GALLO, 1987; LAMATSCH, 1990; LÖBEL, 1998). Para tanto, foi aplicada utilizando as seguintes

heurísticas: *branch-and-bound* (CARPANETO *et al.*, 1989); caminhos mais curtos DELL'AMICO; FISCHETTI; TOTH (1993); *fix-and-optimize* ou, corrigir-e-otimizar (GINTNER; KLIEWER; SUHL, 2005); e *branch-and-bound* e geração de colunas (HADJAR; SOUMIS, 2009). Também, para a formulação *multicommodity*, LÖBEL (1998), desenvolveu dois modelos, uma utilizando a geração de colunas e outra utilizando *branch-and-bound*, sendo que, a aplicação da técnica *branch-and-bound* obteve resultados expressivamente melhores em relação à geração de colunas.

A formulação de conjunto de partições, desenvolvida por RIBEIRO; SOUMIS (1994), foi combinada com a técnica de geração de colunas, trazendo resultados expressivamente rápidos, desencadeando uma série de estudos posteriores com modelos similares, a exemplo de (DESROSIERS *et al.*, 1995), (BARNHART *et al.*, 1996), (DESAULNIERS; LAVIGNE; SOUMIS, 1998) e (OUKIL *et al.*, 2006) e, também, aplicada em combinação com a técnica *simplex dual* por BIANCO; MINGOZZI; RICCIARDELLI (1994).

Com um estudo extenso nas otimizações voltadas para o MDVSP, PEPIN; DESAULNIERS; HERTZ (2008) analisou resultados para cinco heurísticas diferentes. Entre as heurísticas empregadas está o método *branch-and-cut* truncado, o relaxamento lagrangeano com subgradientes, a geração de colunas truncadas, a busca do vizinho mais longe e o *tabu search*. PEPIN; DESAULNIERS; HERTZ (2008) fornece um conjunto de resultados e comparativos envolvendo várias técnicas, demonstrando que, os melhores resultados, tanto em tempo de execução quanto em resultados foi obtida, justamente, pela técnica que abordava a utilização da geração de colunas. Fica explícito nesta comparação, que as atuais heurísticas possuem muita dificuldade em encontrar boas soluções, com baixo *gap*, em um tempo razoável para grandes instâncias. Dentro do limite de uma hora estabelecido pelos autores, instâncias com no máximo 1500 viagens e 8 garagens foram resolvidas, sem garantia de otimalidade. Estes valores são reduzidos se comparados aos problemas de grandes cidades de alguns países como Estados Unidos, e de nações européias. Nesta direção, o desenvolvimento de novos métodos de solução é justificada.

Como o presente trabalho propõe a aplicação de técnicas para a resolução do MDVSP, utilizando-se das técnicas de redução de espaço de estados e de geração de colunas para o MDVSP, a seguir, nas seções 2.3.1 e 2.3.2, as duas heurísticas são

abordadas, respectivamente. Para mais detalhes das outras heurísticas, sugere-se consultar a bibliografia do presente trabalho.

### 2.3.1 Redução de espaço de estados

Esta seção tem como objetivo apresentar a técnica de redução de espaço de estados, demonstrando as três etapas envolvidas no processo.

O espaço de estados é a representação de todas as possibilidades de resposta de um problema, que, em grandes instâncias, pode tomar proporções exponenciais devido ao grande número de possibilidades de viagens. Se considerarmos um problema MDVSP, com uma matriz de tamanho  $n$ , o número de viagens escolhidas para um MDVSP será de  $n$ , enquanto a matriz de espaço de estados poderá conter até  $(n^2) - 1$  viagens, portanto, compreende-se que esta quantidade de variáveis é custosa demais para a resolução do problema MDVSP.

Levando em consideração também que, apenas uma ou algumas das respostas obtidas dentro deste espaço de estados podem ser considerados como soluções ótimas, e pelo número de viagens escolhidas, a retirada de variáveis que, potencialmente, não fariam parte da solução ótima, beneficiaria a resolução do problema, devido à simplificação do problema, através da redução de  $m$  variáveis por linha e coluna.

A redução do espaço de estados, justamente, visa a diminuição das possibilidades do problema, retirando as alternativas que dificilmente fariam parte de uma resposta considerada como solução ótima, para tanto, a escolha de um dimensionamento correto para a redução é muito importante. Reduções dimensionadas adequadamente permitem que problemas de alta complexidade possam ser resolvidos utilizando-se de menor tempo computacional em relação ao espaço de estados original, buscando não comprometer a solução ótima ou, pelo menos, garantindo valores próximos à solução ótima.

Considerando as dificuldades para a resolução do problema MDVSP, justamente, por se tratar de problema de natureza NP-*hard*, ROHDE (2008), desenvolveu uma técnica específica para a redução do espaço de estados, provendo-se de três etapas, explanadas a seguir:

**Etapas 1 - Redução pelo critério estatístico:** A redução pelo critério estatístico busca identificar viagens que podem ser consideradas com baixas chances de

pertencer à solução ótima.

Partindo do pressuposto de que o escalonamento de veículos contenha viagens que tenham incompatibilidades geográficas ou de horários, o custo de deslocamento destas viagens torna-se elevado o suficiente para que a opção de viagem possa ser descartada como uma opção viável na resolução do problema.

A primeira etapa justamente busca identificar e descartar as variáveis com baixas chances de pertencer à solução ótima - com exceção das viagens de saída e chegada das garagens - buscando assim, acelerar a busca pela resposta do MDVSP. Para identificar quais são as melhores variáveis a manter, o procedimento identifica um grupo de  $m$  variáveis por linha e coluna, pelo critério de menor custo, criando um conjunto reduzido de possibilidades de viagens em relação ao modelo original, conseqüentemente, todas as demais variáveis por linhas e colunas não selecionadas são descartadas, sobrando até  $m$  variáveis por linha e coluna.

Um valor baixo para  $m$  pode limitar muito o espaço de estados, retirando viagens candidatas à solução ótima, podendo assim, afetar o resultado final. Um valor de  $m$  muito alto não traria significância no tempo de resposta da solução, por não diminuir significativamente a complexidade do problema. Portanto, para um bom desempenho do primeiro procedimento é necessário uma escolha bem dimensionada para o valor de  $m$ .

Para ROHDE (2008), o critério estatístico utilizado foi a construção de histogramas, de forma a buscar o melhor valor para  $m$ , tratando cada linha e cada coluna como uma distribuição amostral. Fundamentado através de definição de classes, o melhor valor considerado por ROHDE (2008) é de  $\sqrt{(n)}$ .

Por meio de análises, comparando resultados de VSPs para cada garagem em relação ao problema principal e ao problema reduzido, ROHDE (2008) comprovou que a redução da do número de variáveis de até  $n * n$  para até  $m * m$  foi aplicada com sucesso, sem afetar os resultados finais.

**Etapa 2 - Definição das viagens iniciais e finais:** A segunda etapa analisa apenas as viagens de entrada e saída das garagens, para tanto, a etapa dois busca identificar quais são as viagens que devem continuar no espaço de estados.

Para a escolha das viagens iniciais e finais, o modelo MDVSP é transformado em vários VSPs, mais especificamente, um VSP para cada garagem, sendo que, cada

VSP é solucionado separadamente e os resultados das viagens de entradas e saídas das garagens são marcadas para serem mantidas no espaço de estados.

Ao analisar individualmente cada problema VSP, é possível identificar quais são as viagens que foram selecionadas para a entrada e saída das garagens nas soluções, assim, obtendo prováveis candidatas a serem utilizadas como alternativas de resposta para a redução do espaço de estados.

**Etapa 3 - Exclusão de variáveis nunca selecionadas:** Para a terceira etapa, os resultados dos VSPs da segunda etapa são utilizados, de forma a identificar variáveis que nunca foram selecionadas no VSP, exceto as viagens de entradas e saídas das garagens, reduzidas previamente na Etapa 2.

Com base no pressuposto que, se uma variável não foi selecionada no VSP, provavelmente, a mesma variável não será selecionada no MDVSP, portanto, são descartadas todas as viagens que não fizeram parte em nenhum resultado VSP para as viagens.

A seguir, na seção 2.3.2 é apresentada a geração de colunas truncada para o MDVSP.

### 2.3.2 Geração de colunas truncado para o MDVSP

Esta seção tem como objetivo apresentar a heurística de geração de colunas truncado tendo como base a técnica desenvolvida por PEPIN; DESAULNIERS; HERTZ (2008).

A técnica de geração de colunas, primariamente, foi desenvolvida para resolver formulações de relaxamento linear por meio da utilização de um método iterativo, buscando assim, evitar a enumeração de todas as variáveis encontradas (DANTZIG; WOLFE, 1961), com isso, a técnica objetiva otimizar o tempo de resposta computacional sem afetar o resultado final da resolução do problema (PEPIN; DESAULNIERS; HERTZ, 2008).

A geração de colunas decompõe o problema em duas partes, sendo que, a primeira parte é considerado o Problema mestre restritivo ou *Restrictive Master Problem* (RMP) e, a segunda parte é um ou mais subproblemas (DANTZIG; WOLFE, 1961).

No programa de maximização linear, qualquer coluna com custos positivos reduzidos é candidata a entrar como novo valor para a geração de colunas, de forma

a encontrar a coluna que contenha a maior redução de custos, dependendo do algoritmo de aproximação para produzir uma coluna com custos reduzidos positivos objetivando ser adicionada ao problema mestre (RIBEIRO; SOUMIS, 1994).

Se o algoritmo de aproximação falhar na produção de uma ou mais colunas com custos positivos reduzidos, ou seja, não encontrou valores de redução para o problema principal, um algoritmo de otimização do problema de lucro é chamado para tentar provar a otimalidade ou, produzir uma coluna com custos positivos reduzidos (PEPIN; DESAULNIERS; HERTZ, 2008).

Durante o processo de geração de colunas, o RMP continua crescendo pela adição de novas variáveis, sendo possível adicionar mais que uma coluna por iteração, considerando que, como consequência do aumento de tamanho do RMP, o número de variáveis de decisões aumenta, portanto, o tempo para a solução do RMP também aumenta conforme o número de iterações. Por outro lado, o número de iterações é limitado, justamente, pelas restrições que, a cada iteração reduzem o número de possibilidades de solução pela convergência à solução ótima (BARNHART *et al.*, 1996).

O RMP é um subconjunto de variáveis  $\theta_p$ , ou seja, para cada caminho  $p$ , é criado um subconjunto  $\Omega_n^k \subseteq \Omega^k, k \in K$  dos caminhos gerados pelos subproblemas, sendo que, para provar que a solução primal é a solução ótima do problema mestre restritivo, verifica-se os custos de caminhos ainda não gerados não são negativos ( $p \in \cup_{k \in K} (\Omega^k \setminus \Omega_n^k) > 0$ ), para assim, verificar se podem contribuir com o problema mestre, propondo uma ou um conjunto de variáveis (colunas) com um custo negativo reduzido (PEPIN; DESAULNIERS; HERTZ, 2008).

Para cada iteração  $n$ , o subproblema da garagem  $k \in K$  é um problema do caminho mais curto de  $o(k)$  para  $d(k)$  na rede de conexões  $G^k$  com a estrutura de custos modificada. As variáveis  $\pi_n^i, i \in N$  e  $\beta_n^k, k \in K$  são consideradas como os valores das variáveis duais associadas com as restrições (17) e (18) do RMP das iterações, fornecendo assim, valores de custos modificados do arco  $(i, j) \in A^k$ , por meio da perturgação dada em  $c_{ij} - \pi_n^i$ , se  $i \in T$  e  $c_{ij} = \beta_n^k$  se  $i = o(k)$ , portanto, a formulação do subproblema utilizado pela geração de colunas com base na formulação de conjunto de partições pode ser descrita conforme a seguir (PEPIN; DESAULNIERS; HERTZ, 2008):

$$\min \beta_n^k + \sum_{(i,j) \in A^k} (c_{ij} - \pi_n^i) x_{ij}^k \quad (20)$$

$$\sum_{j \in V^k} x_{ij}^k - \sum_{i \in V^k} x_{ji}^k = 0 \quad \forall i \in N \quad (21)$$

$$\sum_{j \in T} x_{n+k,i}^k = 1 \quad \forall k \in K \quad (22)$$

$$\sum_{j \in T} x_{i,n+k}^k = 1 \quad (23)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall A^k \quad (24)$$

O subproblema correspondente para a geração da coluna para o conjunto  $\Omega^k$ , consiste, basicamente, em encontrar o menor caminho saindo e retornando à uma garagem visitando, ao menos, um nodo e retornando à mesma garagem de saída sendo que, a solução do subproblema sempre será um raio convexo (RIBEIRO; SOUMIS, 1994), seguindo os valores da combinação linear dos raios extremos de cada iteração do algoritmo de DANTZIG; WOLFE (1961). Após a obtenção dos custos alterados, o custo de um caminho  $p$  em  $G^k$  corresponde ao custo reduzido da variável  $\theta_p$ , portanto, resolvendo este subproblema na iteração  $n$ , a agenda  $p \in \Omega^k$  com o menor custo reduzido pode ser identificada.

A convergência do método de geração de colunas é maior nas iterações iniciais, ao qual, a cada nova iteração esse comportamento de convergência diminui gradativamente, fazendo com que, em certo ponto, a convergência seja quase nula ou zero. Para evitar a estagnação no processo de convergência da geração de colunas, controla-se o limite de iterações através de um valor mínimo esperado de redução ( $Z_{min}$ ) em um número máximo de iterações ( $I$ ), pré-definidos antes de o método ser executado. A solução é considerada encontrada quando o resultado das variáveis de decisão do conjunto de partições for inteira, ou seja  $|\Omega| \in \{0, 1\}$ .

Com a notação dada anteriormente, o algoritmo de geração de colunas truncado para o MDVSP pode ser descrito conforme a seguir (PEPIN; DESAULNIERS; HERTZ, 2008):

**Passo 0: Inicialização:** São escolhidos os parâmetros  $Z_{min}$  (valor mínimo de diminuição para o truncamento),  $I$  (número mínimo de iterações para o trunca-

mento) e  $\theta_{min}$  (valor mínimo da otimização resultante), iniciando também, as variáveis  $A_i, i \in N$  adicionadas às restrições (17) no RMP, de forma a gerar um coeficiente do tipo *big-m* para a função objetivo (16);

**Passo 1: *Restricted Master Problem*:** O RMP corrente é resolvido usando um algoritmo de programação linear, com um conjunto de caminhos  $\Omega_n^k, k \in K$ , de forma a obter uma solução primal  $(\theta_n, \Lambda_n)$  e uma solução dual  $(\pi_n, \beta_n)$  de custos  $z_n^{RMP}$ ;

**Passo 2: Teste de terminação antecipada da geração de colunas** Se o processo de geração de colunas necessita ser truncado, por não estar mais contribuindo com a redução de valores, evitando assim, a estagnação dos valores (Se  $\theta_n = 0, n - no > I$  e  $z_{n-1}^{RMP} - z_n^{RMP} < Z_{min}$ ), então pula para o Passo 6;

**Passo 3: Resolução do subproblema** Os subproblemas dos caminhos mais curtos são resolvidos utilizando um algoritmo especialista, sendo que para todas as garagens  $k \in K$ , atualiza-se os custos dos arcos em  $A^k$ , aplicando as modificações dos valores de  $(\pi_n, \beta_n)$ , resolvendo o problema de caminho mais curto em  $G^k$ , denotando o melhor valor em  $z_n^{SUB,k}$ ;

**Passo 4: Teste da otimalidade da geração de colunas** Se o valor de cada subproblema não for negativo, o processo de geração de colunas para na solução do RMP corrente, considerando como ótimo o relaxamento linear atual (Se  $z_n^{SUB,k} \geq 0, \forall k \in K$ ), então, pula para o Passo 6;

**Passo 5: Atualização do problema mestre** Atualiza os conjuntos de problemas  $S_n^k$  das agendas calculadas, ao qual tiveram contribuição de custos negativos reduzidos ( $\Omega_{n+1}^k \leftarrow \Omega_n^k \cup S_n^k$  e  $n \leftarrow n + 1$ ), após, retornando ao Passo 1;

**Passo 6: Teste de viabilidade** Executa o teste de viabilidade, PARANDO o processo, caso indique que não foi encontrada uma solução viável (Se  $\Lambda_n \neq 0$  então pare por não existir solução viável);

**Passo 7: Teste de integrabilidade** Executa o teste de integrabilidade, PARANDO o processo, caso considerado que haja uma solução viável (Se  $\theta_n$  é inteiro então pare por ter encontrado uma solução viável);



**Passo 8: Arredondamento** Executa o processo de arredondamento, sendo que todas as viagens consideradas viáveis, são marcadas como viagens viáveis. Caso não haja uma viagem viável, seleciona-se a que mais se aproxima do valor considerado como viável (Para todos os valores fracionados de  $\theta_p$ , ao qual  $\theta_{p,n} \geq \theta_{min}$  então  $\theta_p \leftarrow 1$ . Caso nenhuma for selecionada, então  $\theta_p \leftarrow 1$  quando  $\theta_p$  for a variável com o maior valor fracionário de  $\theta_n$  e, atualiza  $n \leftarrow n + 1$  e  $no \leftarrow n$ ), retornando ao Passo 1.

A técnica contém oito passos, seguindo as premissas da geração de colunas de BARNHART *et al.* (1996), adicionando um controle para valores fracionados, considerado como um modelo para geração de colunas truncado para o MDVSP (PEPIN; DESAULNIERS; HERTZ, 2008). Seguindo os oito passos, a geração de coluna rodará até que as iterações indiquem que a solução ótima foi encontrada ou que não há mais colunas com custos reduzidos. Algumas calibrações podem melhorar ou piorar o resultado ou o desempenho do algoritmo. Como exemplo de calibragem, podemos citar as variáveis  $Z_{min}$  e  $I$ , utilizadas no passo 1, ao qual, indicam o momento de analisar a estagnação da geração de colunas, influenciando no tempo e no valor objetivo.

A seguir, no capítulo 3 são apresentados os procedimentos metodológicos utilizados no presente trabalho.

### 3 PROCEDIMENTOS METODOLÓGICOS

Para o desenvolvimento da presente pesquisa utilizou-se a metodologia de Pesquisa Operacional baseada nas premissas de ARENALES *et al.* (2007), adotando algumas de suas etapas. Cada uma das etapas foram utilizadas para atender aos objetivos deste estudo, sendo descritas a seguir:

**Etapa I - Definição do problema:** Esta pesquisa tem como objetivo solucionar o clássico MDVSP, conforme descrito por BERTOSSI; CARRARRESI; GALLO (1987). Como o MDVSP é NP-*Hard*, foi definido o desenvolvimento uma heurística, cuja principal habilidade seja a de encontrar um bom compromisso entre qualidade da solução e o tempo computacional requerido para encontrá-la.

Os resultados e os tempos de execução relatados por PEPIN; DESAULNIERS; HERTZ (2008), para solucionar instâncias randomicamente geradas, foram considerados como *benchmarking*.

**Etapa II - Modelagem:** O MDVSP foi formulado como um problema de programação linear. A heurística desenvolvida tem como base as técnicas de redução de espaço de estados, proposta por ROHDE (2008), e a técnica de geração de colunas, conforme descrito por PEPIN; DESAULNIERS; HERTZ (2008).

Ambas as técnicas foram analisadas e modificadas com o intuito de otimizar os resultados. Para mais detalhes sobre o método de solução desenvolvido, consultar o capítulo 4.

**Etapa III - Verificação & Validação da solução:** Para a verificação e validação da heurística desenvolvida foram utilizadas as mesmas instâncias utilizadas por PEPIN; DESAULNIERS; HERTZ (2008), disponibilizada pelo mesmo no site <http://people.few.eur.nl/huisman/instances.htm>. Estas instâncias foram geradas aleatoriamente de acordo com os procedimentos descritos em CARPANETO *et*

*al.* (1989).

Os resultados da heurística desenvolvida, tais como número de veículos utilizados e o valor da função-objetivo, bem como o tempo de execução foram comparados com os resultados apresentados por PEPIN; DESAULNIERS; HERTZ (2008) para diferentes métodos de solução, tais como programação linear, geração de colunas truncada, tabu search, etc.

Em um primeiro momento, validou-se a implementação em geração de colunas modificada, comparando resultados em relação à implementação em PEPIN; DESAULNIERS; HERTZ (2008). Com a validação da geração de colunas modificada, as reduções de espaço de estados foram validadas, tendo como base os resultados da geração de colunas modificadas.

**Etapa IV - Experimentos computacionais:** Experimentos computacionais adicionais foram executados, envolvendo instâncias com dimensões superiores às fornecidas por PEPIN; DESAULNIERS; HERTZ (2008). O objetivo do experimento é de demonstrar a potencialidade da heurística desenvolvida para resolver instâncias de maiores dimensões que as já disponibilizadas na literatura durante o desenvolvimento do presente trabalho.

A seguir, no capítulo 4 é apresentado o método de solução desenvolvido.

## 4 MÉTODO DE SOLUÇÃO DESENVOLVIDO

Pelo MDVSP ser um problema do tipo NP-*Hard*, conforme demonstrado na seção 2.2, e como este tipo de problema deve ser resolvido rapidamente para auxiliar tomadas de decisões, optou-se por desenvolver uma heurística. Uma visão geral da heurística pode ser descrita a seguir:

**Passo 1:** Redução do espaço de estados:

**Passo 1.1:** Redução por escolha de melhores caminhos pelos VSPs (R1);

**Passo 1.2:** Redução pelo critério estatístico (R2);

**Passo 2:** Geração de colunas truncado modificado.

O Passo 1 tem como objetivo a redução das possibilidades de viagens, retirando as viagens que não foram consideradas dentro dos critérios de melhores caminhos e do critério estatístico e o Passo 2 tem como objetivo resolver o MDVSP através da matriz reduzida, resultante do passo 1, por meio da técnica de geração de colunas truncado.

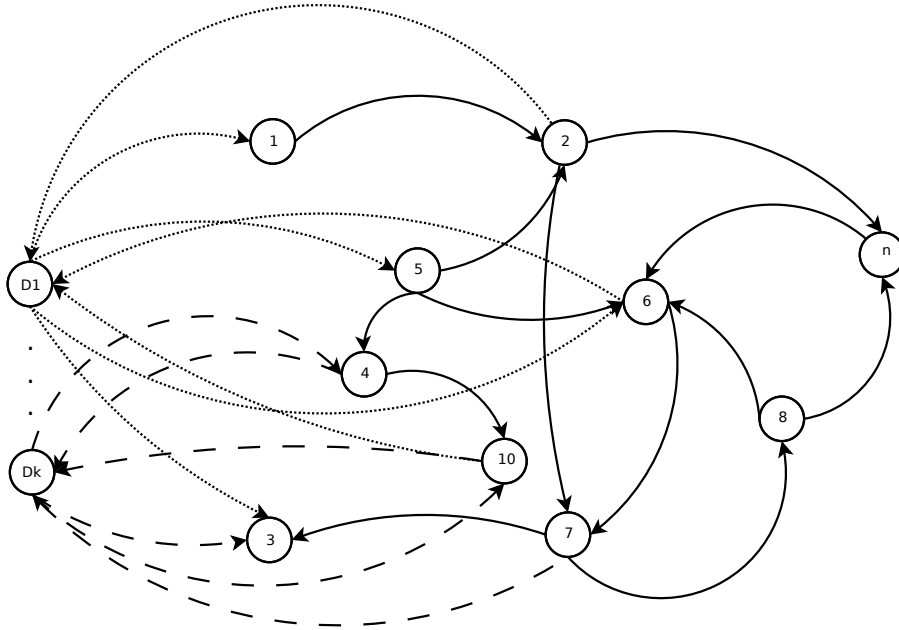
Ambas as técnicas sofreram modificações com o objetivo de melhoria dos resultados, sendo apresentadas conforme se segue: Na seção 4.1 é abordada a definição da rede, na seção 4.2, a redução do espaço de estados e, na seção 4.3 a geração de colunas truncada.

### 4.1 DEFINIÇÃO DA REDE

A presente heurística foi concebida para trabalhar com arcos do tipo de conexão, similar aos conceitos dos modelos de CARPANETO *et al.* (1989). A Figura 1 ilustra uma rede MDVSP de arcos de conexão, sendo que os círculos indicam um local e os

arcos (linhas) indicam as possibilidades de viagens. Os círculos  $D_1$  e  $D_k$  indicam as garagens do problema, enquanto que os outros pontos são os locais que devem ser visitados. As linhas pontilhadas e tracejadas indicam as possibilidades de viagens de *pull-in* e *pull-out* das garagens, enquanto que, as linhas contínuas representam as viagens de *deadheading*, sendo que, as setas indicam o destino de cada arco.

**Figura 1 – Exemplo de rede de arcos de conexão**



O formato de arquivo utilizado contém a seguinte estrutura: a primeira linha, informa o número de garagens  $K$ ; a segunda linha, o número de viagens  $n$ ; a terceira linha, o número máximo de veículos a serem utilizados por cada garagem; da quarta linha em diante, é fornecida uma matriz  $((K + n) * (K + n))$ , sendo que o número da linha  $i$  e na coluna  $j$  indica que:

- Se  $i \leq K$  e  $j > K$ : custo de saída da garagem  $i$  para a viagem  $j - K$ ;
- Se  $i > K$  e  $j \leq K$ : custo de retorno à garagem  $i$  pela viagem  $j - K$ ;
- Se  $i > K$  e  $j > K$ : custo da viagem até o ponto  $j - K$  saindo de  $i - K$ ;

A seguir é ilustrada uma matriz com custos  $d_{ij}$ , indicando o custo de uma viagem que sai do ponto  $i$  e vai até o ponto  $j$ , correspondendo assim, aos custos das viagens de entradas ou saídas de garagens (*pull-in* e *pull-out*, respectivamente) e  $t_{ij}$ , correspondendo aos custos de uma viagem saindo do ponto  $i$  e, posteriormente,

chegando ao ponto  $j$ , (ou seja, viagens de *deadheading*). Pode-se observar que os custos de *pull-in* e *pull-out* vão de 1 até  $k$ , enquanto que os custos de *deadheading* vão de  $k + 1$  até  $n + k$  na matriz  $MDVSP$ .

$$MDVSP = \begin{bmatrix} -1 & \cdots & -1 & d_{11} & d_{12} & d_{13} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & \cdots & -1 & d_{k1} & d_{k2} & d_{k3} & \cdots & d_{kn} \\ d_{11} & \cdots & d_{k1} & t_{11} & t_{12} & t_{13} & \cdots & t_{1n} \\ d_{21} & \cdots & d_{k2} & t_{21} & t_{22} & t_{23} & \cdots & t_{2n} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ d_{n1} & \cdots & d_{kn} & t_{n1} & t_{n2} & t_{n3} & \cdots & t_{nn} \end{bmatrix}$$

Caso as viagens de  $t_{ij}$  são incompatíveis, os valores de custos, tanto de  $d_{ij}$  e  $t_{ij}$  são configurados com o valor  $-1$ , bem como as viagens que não podem ser executadas, a exemplo das viagens que corresponderiam aos deslocamentos entre garagens.

## 4.2 REDUÇÃO DO ESPAÇO DE ESTADOS

Para a técnica de redução de espaço de estados, adotou-se as mesmas ideias do trabalho de ROHDE (2008), sendo que, alterações visando a otimização dos resultados foram aplicadas.

Primeiramente, a redução de escolha dos melhores caminhos pelo VSP é aplicada (Passo 1.1), baseado no princípio de que a primeira redução contém um critério mais apurado em relação à redução por critério estatístico (Passo 1.2).

A figura 2 indica o processo de interpretação da redução de espaço de estados, ilustrando a forma como são processados os subpassos e coletadas as informações para aplicar a redução na matriz original, ao qual, alimentarão uma matriz de igual tamanho, indicando quais são os caminhos a serem mantidos:

O algoritmo para redução de espaço de estados, inicialmente, cria uma matriz com o mesmo tamanho da matriz original, alimentando todas as suas posições com uma marcação que considera o respectivo ponto da matriz original como inválido. Após, o algoritmo verifica quais reduções devem ser aplicadas e as executa. Cada redução, ao final de sua execução, alimenta a matriz de redução de espaço de estados, marcando os caminhos válidos em suas respectivas posições.

**Figura 2 – Processo de interpretação da redução do espaço de estados**

```

Cria matriz de redução do espaço de estados, marcando todos os caminhos
como inválidos;
if Configurado para aplicar a redução 1 then
|   Processa redução do espaço de estados por escolha de melhor caminho nos
|   VSPs, marcando na matriz de redução de espaço de estados, os caminhos
|   a serem mantidos;
end
if Configurado para aplicar a redução 2 then
|   Processa redução do espaço de estados por critério estatístico, marcando
|   na matriz de redução de espaço de estados, os caminhos a serem mantidos;
end
Atualiza espaço de estados original, marcando como inviáveis as viagens não
marcadas na(s) redução(ões);

```

Após a aplicação das reduções, percorre-se a matriz de redução e, para cada caminho inválido, substitui-se o valor na posição equivalente na matriz de estados original, diminuindo assim, as possibilidades da matriz de estados original retirando-se os caminhos através de alguma ou das duas técnicas.

A seguir, nas seções (4.2.1 e 4.2.2) são apresentados os procedimentos de redução por escolha de melhores caminhos pelo VSP e de redução pelo critério estatístico, respectivamente.

#### 4.2.1 Redução por escolha de melhores caminhos pelo VSP

Para a redução por escolhas de melhores caminhos pelo VSP, ROHDE (2008) processava os resultados por meio de programação linear, resolvendo um VSP para cada uma das garagens, criando uma matriz com os *deadheading* originais do problema e, para cada garagem, adicionando os caminhos de *pull-in* e *pull-out* correspondentes à garagem a ser analisada.

ROHDE (2008) executava dois procedimentos com base nos resultados dos VSPs, sendo: retirada das viagens não selecionadas nos VSPs e retirada das viagens de entrada e saída das garagens não selecionadas nos VSPs. Ambas as retiradas envolviam as marcações das variáveis que nunca foram selecionadas em algum dos VSPs. Como o procedimento utiliza-se dos mesmos resultados dos VSPs, o segundo e terceiro procedimentos de ROHDE (2008) foram unificados na presente técnica.

Em substituição à programação linear para a resolução dos VSPs, optou-se por

utilizar um algoritmo especialista, devido ao desempenho dos resultados. O algoritmo escolhido foi o JV (JONKER; VOLGENANT, 1987). Para tanto, a estrutura de dados apresentada na seção 4.2 é convertida para vários VSPs, sendo criada uma rede equivalente por garagem.

A estrutura da matriz VSP utilizada para o JV, criada para cada garagem, é do tipo *quasi-assignment*, permitindo solucionar o problema por meio de um algoritmo de resolução do problema de *assignment* (PINTO; PAIXAO, 1987). A estruturação da matriz *quasi-assignment* pode ser ilustrada conforme a seguir.

$$VSP^k = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{1n} & d_{1n+1} & \cdots & d_{12n} \\ t_{21} & t_{22} & t_{23} & \cdots & t_{2n} & d_{2n+1} & \cdots & d_{22n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \ddots & \ddots \\ t_{n1} & t_{n2} & t_{n3} & \cdots & t_{nn} & -1 & -1 & -1 \\ d_{n+11} & d_{n+12} & d_{n+13} & \cdots & d_{n+1n} & -1 & -1 & -1 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \ddots & \ddots & \ddots \\ d_{2n1} & d_{2n2} & d_{2n3} & \cdots & d_{2nn} & -1 & -1 & -1 \end{bmatrix}$$

Com visto anteriormente, a matriz  $VSP^k$  difere da matriz  $MDVSP$ , contendo as viagens de *deadheading* de 1 até  $n$  e, após as viagens *deadheading*, as viagens de *pull-in* e *pull-out* são localizadas ao final da matriz, sendo repetidas de  $n + 1$  até  $2n$ , assim, duplicando o tamanho da matriz (PINTO; PAIXAO, 1987).

Criado para resolver problemas de *assignment*, o algoritmo JV permitiu, através de uma modificação da estrutura de conexões do MDVSP, resolver os problemas VSPs de forma rápida, pois o algoritmo JV tem uma complexidade  $\Theta(n^2 \cdot \log n)$ , sendo necessárias iterações para que o algoritmo seja rodado várias vezes, de forma a buscar o melhor valor, utilizando o mínimo de veículos, portanto, para cada iteração adiciona-se ou retira-se uma linha e coluna correspondentes às viagens de *pull-in* e *pull-out* de um veículo, até que seja encontrado o melhor valor com um mínimo de veículos. Na figura 3, é ilustrado o primeiro passo da técnica de redução de espaços, ou seja, a redução de melhores caminhos pelos VSPs.

Com o intuito de acelerar o processo, a constante  $JVCOEF$  fornece um número que indica uma quantidade inicial de veículos que deve ser utilizada para solucionar a resolução do VSP, diminuindo o número de iterações pela aproximação do valor



**Figura 3 – Redução por escolha de melhores caminhos do VSP**

```

for cada garagem  $k$  do
  Cria uma matriz  $VSP^k$  a partir da matriz  $MDVSP$ ;
  Veículos selecionados  $\leftarrow n * JVCOEF$ ;
  Modo de iteração  $\leftarrow$  Adiciona veículos;
  Iterações  $\leftarrow 0$ ;
  while Melhor resultado == falso do
    resultadoAnterior  $\leftarrow$  resultado;
    Atualiza Matriz  $VSP^k$ , adicionando ou retirando veículos, conforme
    modo de iteração;
    resultado  $\leftarrow JV(VSP^k)$ ;
    if resultadoAnterior == resultado && iterações < 2 then
      | Modo de iteração  $\leftarrow$  Retira veículos;
    end
    else
      | if resultadoAnterior == resultado then
        | Melhor resultado  $\leftarrow$  verdadeiro;
      end
    end
    Iterações  $\leftarrow$  Iterações + 1;
  end
  Registra caminhos na matriz de redução;
end

```

ideal de viagens, conseqüentemente, acelerando o processo de solução do VSP. Para tanto, a matriz de *quasi-assignment* terá um número de linhas e colunas de  $n$  até  $n * JVCOEF$ , indicando assim, que o número inicial de veículos a ser utilizado para a resolução é de  $(n * JVCOEF) - n$ .

Como a cada iteração é adicionado ou retirado um veículo, verifica-se se o valor obtido é o mesmo da iteração anterior. Se o valor for o mesmo, indica-se que o passo anterior encontrou o melhor valor, com um mínimo de veículos, sendo considerado o resultado do VSP. Caso o valor seja diferente, as iterações continuam até que se chegue a dois valores iguais.

Quando na segunda iteração é encontrado o melhor valor, presume-se que o fator  $JVCOEF$  indicou um número de veículos acima do ideal, trocando a iteração de adição de viagens, para retirada de viagens, diminuindo assim, o número de veículos utilizado. Quando a iteração indicar que o valor da iteração anterior for diferente da iteração atual, indica-se que a iteração anterior é o que obteve a melhor resposta, utilizando um número mínimo de veículos.

A seguir, na seção 4.2.2. é apresentada a redução por critério estatístico.

#### 4.2.2 Redução pelo critério estatístico

A redução do Passo 1.2. envolve a seleção de  $m$  melhores valores por linha e coluna, indicando assim, que as viagens com menores custos são as com maiores possibilidades de serem selecionadas na solução ótima.

Figura 4 – Redução por escolha de melhores valores

```

Cria duas listas ordenada  $m_i[1..m] \leftarrow 0$  e  $m_j[1..m] \leftarrow 0$ ;
MDVSP'[1..n][1..n]  $\leftarrow 0$ ;
for cada linha  $i$  do
  for cada coluna  $j$  do
    if Valor da coluna  $j$  na linha  $i$  for menor que os valores contidos em
     $m_i$  then
      Atualiza lista ordenada  $m_i$ , adicionando o valor da coluna  $j$  na
      linha  $i$ ;
    end
  end
  MDVSP'  $\leftarrow$  marcaCaminhosValidos( $m_i, i$ );
   $m_i[1..m] \leftarrow 0$ ;
end
for cada coluna  $j$  do
  for cada linha  $i$  do
    if Valor da linha  $j$  na coluna  $i$  for menor que os valores contidos em
     $m_j$  then
      Atualiza lista ordenada  $m_j$ , adicionando o valor da linha  $i$  na
      coluna  $j$ ;
    end
  end
  MDVSP'  $\leftarrow$  marcaCaminhosValidos( $m_j, j$ );
   $m_j[1..m] \leftarrow 0$ ;
end

```

O algoritmo de redução pelo critério estatístico segue os mesmos princípios do algoritmo de ROHDE (2008), conforme ilustrado na Figura 4. ROHDE (2008) utilizou um critério estatístico para fundamentar a seleção de melhores valores, considerando  $m$  melhores valores nas linhas e colunas, com base no critério estatístico definido por MARTINS (2002), o qual, considerou o valor  $m = \sqrt{n}$  como um valor equilibrado para a definição das classes do histograma de redução.

O algoritmo de redução por escolha de melhores valores contém complexidade  $\Theta(n^2)$ , por tratar-se de uma busca direta de  $m$  menores valores em cada coluna e

linha ROHDE (2008). A seguir, na seção 4.3 é apresentado sobre o método desenvolvido para a geração de colunas truncado.

### 4.3 GERAÇÃO DE COLUNAS TRUNCADO

Para a geração de colunas truncado, os princípios seguem a praticamnte mesma lógica desenvolvida por PEPIN; DESAULNIERS; HERTZ (2008), descrito na seção (2.3.2), tendo a ressalva de que algumas modificações foram feitas com o intuito de acelerar o processo de convergência de valores, considerado como um grave problema para a geração de colunas.

Conforme descrito em DESROCHERS; SOUMIS (1989), o modelo de conjunto de partições pode ser modificado, para obter sobrecarga ou não-utilização de viagens, de forma a oferecer uma solução contínua e válida para o problema, mesmo sobrecarregando alguns caminhos. Sendo assim, a modificação da restrição acelera a convergência de valores fornecendo um bom resultado de forma mais rápida que o modelo original de conjunto de partições.

Para a resolução da geração de colunas, a formulação de conjunto de partições foi utilizada, sendo que, da primeira iteração até a primeira estabilização (considerado apto para arredondar), a formulação da restrição (17) foi modificada para permitir um relaxamento nos valores duais, assim, acelerando o processo de convergência dos valores. A restrição modificada pode ser descrita conforme a seguir:

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip} \theta_p \geq 1 \quad \forall i \in T \quad (25)$$

No processo iterativo da geração de colunas, quando houver a primeira estabilização, ou seja, quando o processo de geração de colunas não teve, em um mínimo de  $I$  iterações, uma redução mínima no melhor valor de  $Z_{min}$ , modifica-se todas as colunas geradas para que atendam a apenas uma viagem, ou seja, a restrição (25) é alterada para seu estado original (17), sendo assim, após a modificação, o algoritmo segue os mesmos princípios da geração de colunas truncado de PEPIN; DESAULNIERS; HERTZ (2008).

Considerando a ideia de permitir a utilização de conjunto de partições modificadas, pelo princípio de DESROCHERS; SOUMIS (1989), após a primeira estabi-

lização, o problema é resolvido pela formulação original de conjunto de partições, utilizando todos os caminhos válidos gerados até o presente processo assim, acelerando o processo de convergência e, ao retornar à formulação original, a sobrecarga ou não utilização de viagens é eliminada, porém, por manter os valores já gerados, o ponto de convergência alcançado até o momento é mantido, sendo assim, obtém-se o melhor valor muito mais rápido do que a utilização da formulação original.

Para uma melhor compreensão do algoritmo, a Figura 5 ilustra o processo de geração de colunas modificado.

**Figura 5 – Algoritmo de geração de colunas truncado com modificações**

```

Considera problema relaxado como não estabilizado;
while Problema relaxado - função objetivo (16), com restrições (25), (18) e
(19) - considerado como não estabilizado do
    Executa processo de geração de colunas com a restrição relaxada;
    if Houve a primeira estabilização then
        Modifica problema mestre, modificando a restrição relaxada (25) para
        a formulação da restrição original (17);
        Considera o problema relaxado como estabilizado;
    end
end
Continua o processo de geração de colunas com a restrição original;

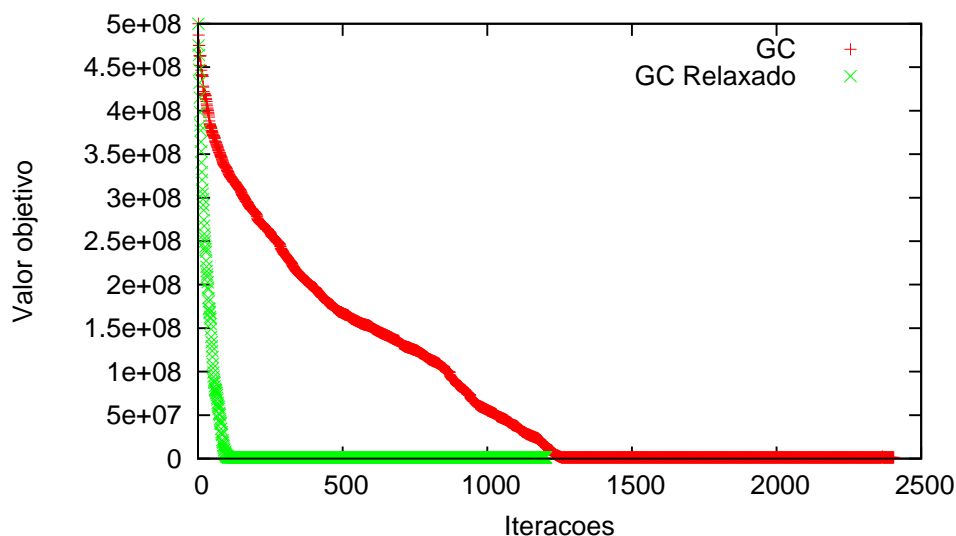
```

Com exceção da alteração na relaxamento da formulação da restrição dos caminhos, da primeira iteração até a primeira estabilização, o algoritmo de geração de colunas truncado segue as mesmas premissas citadas na seção 2.3.2.

A ideia da modificação da restrição é de criar uma maior convergência por meio do relaxamento da restrição, conseqüentemente, os valores de precificação fornecidos pelas duas do problema mestre aos subproblemas são mais altos. Sendo assim, a convexidade do poliedro é maior em relação à restrição não relaxada, fazendo com que, a quantidade de iterações necessárias para se chegar ao melhor valor seja menor, se comparado com as iterações da formulação original.

Como exemplo, a figura 6 ilustra um gráfico comparativo, com o número de iterações exigidas para obtenção do melhor valor, sendo aplicadas em uma instância com  $K = 4$  e  $N = 500$ , sendo que, a linha vermelha indica a geração de colunas truncado e a linha verde indica a geração de colunas com o relaxamento. Como pode-se notar, a geração de colunas com o relaxamento tem um comportamento

Figura 6 – Convergência entre GC truncado e GC com relaxamento



de convergência mais rápida em relação à geração de colunas sem relaxamento, sendo assim, o número de iterações exigido para resolver o problema é bem menor, reduzindo tempo de execução da solução. Nesta simulação, 1203 iterações foram necessárias para a geração de colunas com relaxamento para obtenção do resultado, enquanto que, a geração de colunas sem o relaxamento necessitou de 2405 iterações. Na iteração 1203, o melhor valor para a GC relaxada era de 1288800, enquanto que para a GC o melhor valor foi de 13145100. A GC obteve seu melhor valor, de 1289270 com 2405 iterações. Ambos os valores foram obtidos com as mesmas calibrações, com as mesmas instâncias e com o mesmo algoritmo, mudando apenas a estrutura da restrição.

Para a resolução dos subproblemas, dois algoritmos especializados foram testados. Em um primeiro momento, o algoritmo de (BELLMAN, 1958; FORD; LESTER, 1956) foi utilizado, permitindo assim, a convergência entre as iterações, porém, o tempo de resposta foi considerado como insatisfatório, comparado os tempos de resolução com trabalhos anteriores. Como segunda opção, o algoritmo SLF (*Small Label to the Front*) (BERTSEKAS, 1993) foi utilizado, buscando otimizar o tempo de resposta da resolução dos subproblemas, mantendo-se rápido e estável, sendo então, o algoritmo especialista escolhido para resolver os subproblemas.

A seguir, no capítulo 5, são apresentados os experimentos computacionais executados com o presente método de solução desenvolvido.

## 5 EXPERIMENTOS COMPUTACIONAIS

Este capítulo aborda os experimentos computacionais executados para o presente estudo. Os mesmos foram executados em uma máquina com processador Intel Core I5-3210M CPU @ 2.50GHz, com 6Gb de memória RAM, todos rodando na plataforma Ubuntu Linux 64bits. O algoritmo foi desenvolvido em linguagem C++. O CPLEX foi utilizado como biblioteca solucionar os problemas de programação linear. A geração de colunas foi desenvolvida sem utilização de frameworks específicos.

A seguir, são apresentadas as calibrações aplicadas na redução do espaço de estados (seção 5.1) e na geração de colunas truncado (seção 5.2), bem como, a verificação da geração de colunas truncado (seção 5.3), a validação da heurística (seção 5.4), em seguida, experimentos computacionais com problemas de maiores dimensões (seção 5.5) e, por último, discussões acerca dos trabalhos desenvolvidos (seção 5.6).

### 5.1 CALIBRAÇÕES DA REDUÇÃO DO ESPAÇO DE ESTADOS

Para a redução do espaço de estados, existem duas importantes variáveis a serem calibradas de forma a otimizar os resultados, conforme descrito abaixo.

Para o Passo 1.1, que envolve a redução por meio da seleção de melhores caminhos pelo VSP, o algoritmo necessita encontrar a quantidade ideal de veículos para encontrar o melhor valor. O fator  $JVCOEF$ , quanto mais se aproximar do valor ideal de veículos, menos iterações o algoritmo do Passo 1.1 precisará executar, consequentemente, fornecendo a melhor resposta mais rapidamente. Para o presente trabalho, utilizou-se valores entre 1.2 e 1.34, indicando um percentual inicial de veículos entre 20% e 34% em relação ao tamanho  $n$  da matriz original, dependendo muito da instância a ser resolvida. A seguir, na seção 5.2 são apresentadas

as calibrações efetuadas para a geração de colunas.

## 5.2 CALIBRAÇÕES DA GERAÇÃO DE COLUNAS TRUNCADO

Na geração de colunas truncado, três variáveis importantes tem efeitos consideráveis nos resultados, caso calibradas corretamente, conforme descrito abaixo.

Primeiramente, para a variável  $Z_{min}$ , os valores variaram entre 0 e 2000, com o intuito de buscar o melhor valor, como indicado no trabalho de PEPIN; DESAULNIERS; HERTZ (2008). Valores maiores podem aumentar o tempo de resposta, porém, a resposta obtida é de melhor qualidade, devido à gerações de um número maior de colunas. Para algumas instâncias, o valor de 2000 mostrou-se mais eficiente em relação ao valor 0, na questão de tempo de resolução, justamente, pela necessidade de um número menor de iterações.

Para a variável  $I$ , os valores variaram entre 2 a 10. Em certos momentos, aumentar o número de iterações melhorou o resultado final. Porém, para instâncias que já tinham um bom valor, com  $I = 2$ , comparados aos resultados de PEPIN; DESAULNIERS; HERTZ (2008), a tentativa de aumento do valor de  $I$  fez com que o resultado da função objetivo fosse pior em relação a um valor de  $I$  baixo.

Para o valor  $\Omega_{min}$ , os melhores resultados variam entre 0.7 e 0.85, sendo que, a diferença entre processar a instância com valores diferentes para  $\Omega_{min}$  melhoravam ou pioravam resultado final. O valor de 0.8, foi o valor que trouxe melhores resultados para a maioria das instâncias. As alterações no valor de  $\Omega_{min}$  não trouxeram mudanças significativas com relação aos tempos de execução.

Para a calibragem, utilizou-se de uma sequência de testes, ao qual testava-se valores de  $\Omega_{min}$  somando-se um intervalo de 0,25 para cada rodada. Em certos momentos, a geração de colunas retornava um valor em poucos segundos, ficando com uma diferença relativamente alta em relação ao valor ótimo encontrado por PEPIN; DESAULNIERS; HERTZ (2008), sendo assim, os valores de  $I$  eram aumentados, de forma a forçar a criação de mais colunas com raio de convexidade maior, resultando em resultados melhores, porém, os tempos de execução foram prejudicados, devido a um maior número de iterações. A seguir, na seção 5.3 é apresentada a verificação da implementação da geração de colunas.

### 5.3 VERIFICAÇÃO DA IMPLEMENTAÇÃO DA GERAÇÃO DE COLUNAS TRUNCADO

Esta seção apresenta a verificação da implementação da geração de colunas truncado, de forma a demonstrar a validade da implementação desenvolvida. As instâncias utilizadas foram as mesmas criadas randomicamente para o trabalho de PEPIN; DESAULNIERS; HERTZ (2008), sendo concebidas a partir dos conceitos dos modelos de CARPANETO *et al.* (1989). As instâncias utilizadas, tem tamanhos de viagens  $|T| \in \{500, 100, 1500\}$  e de garagens  $|K| \in \{4, 8\}$ . Há cinco instâncias para cada tamanho, ao qual foram resolvidas com várias calibrações, sendo as que tiveram melhor desempenho foram registradas para fins de comparação.

A forma de comparação foi feita através de diferenças de percentual, sendo calculadas da seguinte forma:  $GAP = \frac{100*(VB-VO)}{VB}$ , onde,  $VO$  é o valor obtido pelos resultados da heurística ou por parte da heurística desenvolvida e  $VB$  é o valor base, que, na validação da geração de colunas são os valores de PEPIN; DESAULNIERS; HERTZ (2008) e Huismann (site <http://people.few.eur.nl/huisman/instances.htm>) e, na validação da heurística, são os resultados da geração de colunas modificada. Valores de *gap* negativos, indicam valores maiores em relação ao valor base, valores de *gap* positivos, indicam valores menores em relação ao valor base, o valor 0 indica que não há diferenças entre os comparativos.

Com o intuito de validar os resultados, foram feitas comparações de tempo de execução (Tabela 4), número de garagens e valor da função objetivo (Tabela 5). A Tabela 4 apresenta os comparativos de médias de tempo de resolução, ilustrando a informação das instâncias (coluna 1); o tempo médio de PEPIN; DESAULNIERS; HERTZ (2008) (coluna 2); tempo médio da geração de colunas desenvolvida (coluna 3) e o percentual de diferença entre os tempos de resolução (*GAP*) (coluna 4). Para que fosse possível criar um comparativo de tempos, a Tabela 4, apresenta a média dos tempos de resolução para cada grupo de instâncias, tendo em comum o número de viagens e garagens. Os valores de tempo de PEPIN; DESAULNIERS; HERTZ (2008) foram arredondados, e portanto, o mesmo formato foi seguido para os valores apresentados.



Tabela 4 – Comparativo de média de tempo de resolução do MDVSP

K	Instância		Tempo (seg.)		
	T	PEPIN	GC	GAP (%)	
4	500	77	39	<b>49.35064</b>	
4	1000	651	551	<b>15.36098</b>	
4	1500	2203	2258	-2.49660	
8	500	119	43	<b>63.86555</b>	
8	1000	857	553	<b>35.47258</b>	
8	1500	3085	2117	<b>30.00846</b>	

Em questão de desempenho de tempo, nota-se um ganho esperado em relação aos tempos de PEPIN; DESAULNIERS; HERTZ (2008), principalmente, pela evolução de hardware em relação aos experimentos executados por estes autores. As diferenças de melhores valores são apenas apresentados para fins ilustrativos, pois não há sentido em uma comparação mais formal, considerando as diferenças tanto no software como no hardware empregados nas duas implementações.

Para as tabelas a seguir, os resultados são apresentados por instância, devido à disponibilidade das informações no site pessoal de Huismann, permitindo assim, comparar os resultados de forma individual para cada instância. Na Tabela 5 são apresentados os comparativos de melhor valor encontrados pelas resoluções, apresentando as seguintes informações: Garagens da instância (coluna 1); Viagens da instância (coluna 2); número de veículos encontrados por Huismann (coluna 3); número de veículos da GC (coluna 4); o melhor resultado de Huismann (coluna 4); melhor resultado da GC (coluna 5) e; *GAP* (coluna 6).

Dependendo das calibrações aplicadas na geração de colunas, os resultados eram obtidos em tempos menores, porém, o melhor valor tinha uma diferença muito alta em relação ao melhor valor de Huismann. Para tanto, considerou-se o equilíbrio entre valor e tempo de resolução, por meio da calibração dos valores da geração de colunas.

Para todas as instâncias rodadas, foi constatado que obteve-se um bom valor da função objetivo, sendo que, para a maioria das instâncias, os valores obtidos são melhores do que os resultados fornecidos pelo Huisman. O número de viagens para cada instância, em sua maioria é igual, indicando que houve uma escolha dos melhores caminhos para os casos em que a diferença é positiva em relação ao melhor valor. Em alguns casos, o número de veículos é bem diferente, indicando que a estruturação dos caminhos diferem bastante entre as duas rotinas. Dado

Tabela 5 – Comparativo de melhores valores para a geração de colunas

Instância		Veículos		Melhor valor		
K	T	PEPIN	CG	PEPIN	GC	GAP (%)
4	500	331	123	1289114	1288670	<b>0.03444</b>
4	500	118	118	1241618	1242620	-0.08070
4	500	125	123	1283811	1283350	<b>0.03591</b>
4	500	120	120	1258634	1258460	<b>0.01382</b>
4	500	126	126	1317077	1316440	<b>0.04836</b>
4	1000	241	241	2516247	2515500	<b>0.02969</b>
4	1000	229	229	2413393	2412620	<b>0.03203</b>
4	1000	233	233	2452905	2454400	-0.06095
4	1000	237	237	2490812	2489160	<b>0.06632</b>
4	1000	238	238	2519191	2517880	<b>0.05204</b>
4	1500	368	368	3830912	3828630	<b>0.05957</b>
4	1500	338	364	3559176	3560800	-0.04563
4	1500	350	350	3649757	3647470	<b>0.06266</b>
4	1500	326	326	3406815	3407790	-0.02862
4	1500	343	343	3567122	3566950	<b>0.00482</b>
8	500	124	124	1292411	1292030	<b>0.02948</b>
8	500	123	123	1276919	1276300	<b>0.04848</b>
8	500	126	128	1304251	1303900	<b>0.02691</b>
8	500	123	365	1277838	1277020	<b>0.06401</b>
8	500	123	123	1276010	1275210	<b>0.06270</b>
8	1000	232	232	2422112	2422100	<b>0.00050</b>
8	1000	244	247	2524293	2524530	-0.00939
8	1000	247	247	2556313	2556320	-0.00027
8	1000	237	237	2478393	2476960	<b>0.05782</b>
8	1000	240	240	2498388	2497560	<b>0.03314</b>
8	1500	337	366	3500160	3500160	<b>0.00000</b>
8	1500	366	366	3802650	3802180	<b>0.01236</b>
8	1500	349	349	3605094	3603240	<b>0.05143</b>
8	1500	338	338	3515802	3516560	-0.02156
8	1500	360	360	3704953	3705790	-0.02259

os comparativos de tempos de resolução, o número de viagens e resultado obtido, considera-se que o algoritmo mostrou-se capaz de resolver o MDVSP, validando assim, a geração de colunas truncada modificada.

#### 5.4 VALIDAÇÃO DA HEURÍSTICA

A validação da heurística seguiu os mesmos princípios e as mesmas instâncias da validação da geração de colunas modificada, sendo que, primeiramente, validou-se cada redução individualmente - R1 e R2 - e, em seguida, foi feita uma validação com o algoritmo completo R1+R2. Cada validação utilizava um ou mais processos de redução juntamente com a geração de colunas, ou seja, primeiro aplicava-se a(s) redução(ões) e, após, executava-se a geração de colunas modificada com a matriz reduzida.

Para a análise dos resultados da validação da heurística, os resultados da geração de colunas modificada serviu de *benchmarking*, visto que, foram validados através da comparação com os trabalhos de PEPIN; DESAULNIERS; HERTZ (2008). A análise dos resultados da validação permitiram não apenas utilizar os comparativos de valores da função objetivo, como também executar comparativos de tempo de resolução, por ter como base os mesmos recursos para o processamento dos resultados.

São apresentados nesta validação, os comparativos de valores da função objetivo entre a geração de colunas com e sem reduções (Tabela 6) e os comparativos de tempo entre a geração de colunas com e sem reduções (Tabela 7), bem como comparativos de tempos e valores entre a geração de colunas e a heurística completa (Tabela 8), de forma a oferecer informações que permitam ao analista escolher a combinação mais apropriada dadas as características de suas necessidades.

Na tabela 6, são apresentados os resultados da validação das reduções em comparação com o melhor valor, informando: Garagens (coluna 1); Viagens (coluna 2); Resultados da GC (coluna 3), Resultados com R1 (coluna 4); *GAP* da R1 (coluna 5); Valor da R2 (coluna 6); *GAP* da R2 (coluna 7).

Para a redução de melhores caminhos pelo VSP, utilizando somente a R1, os *gaps* de melhores valores ficaram mais aceitáveis em relação a R2, indicando valores mais aproximados em relação aos valores de somente utilizar a (GC). O *gap* é bem variável para R2, os resultados indicam que a qualidade da primeira redução é maior que a segunda redução, justamente, por R1 ter um critério mais apurado que R2.

Na Tabela 7 são comparados os tempos, sendo apresentados: Veículos (coluna 1); Viagens (coluna 2); Tempo da GC (coluna 3); Tempo com R1 (coluna 4); *GAP* com R1 (coluna 5); Tempo com R2 (coluna 6); *GAP* com R2 (coluna 7).

Os tempos indicam uma aceleração considerável no tempo de resolução para ambas as reduções. Como constatado, a redução por meio de melhores caminhos pelo VSP - redução R1 - ofereceu um bom desempenho na redução do tempo de execução, justamente, pela grande redução número de variáveis a serem analisadas pelo GC, por considerar apenas viagens relevantes, por isso, o tempo de resolução diminuiu, justamente pela diminuição da complexidade do problema.

Para a redução por critério estatístico - redução R2 - ainda há um ganho significa-

Tabela 6 – Comparativo de melhores valores entre geração de colunas com e sem reduções

Instância		Resultados				
K	T	GC	R1	GAP (%)	R2	GAP (%)
4	500	1288670	1295330	-0.51681	1495060	-16.01574
4	500	1242620	1248100	-0.44100	1348660	-8.53358
4	500	1283350	1299740	-1.27713	1402420	-9.27806
4	500	1258460	1267120	-0.68814	1398640	-11.13901
4	500	1316440	1323710	-0.55225	1405840	-6.79104
4	1000	2515500	2533270	-0.70642	2529770	-0.56728
4	1000	2412620	2445260	-1.35289	2543200	-5.41237
4	1000	2454400	2592310	-5.61889	2688810	-9.55060
4	1000	2489160	2499590	-0.41902	2692580	-8.17223
4	1000	2517880	2623820	-4.20751	2802840	-11.31746
4	1500	3828630	3939700	-2.90104	3869020	-1.05495
4	1500	3560800	3582130	-0.59902	3632340	-2.00910
4	1500	3647470	3801290	-4.21717	3651830	-0.11953
4	1500	3407790	3612470	-6.00624	3659690	-7.39189
4	1500	3566950	3854960	-8.07441	3686510	-3.35188
8	500	1292030	1318360	-2.03788	1448520	-12.11195
8	500	1276300	1326390	-3.92463	1413610	-10.75844
8	500	1303900	1310350	-0.49467	1314680	-0.82675
8	500	1277020	1285890	-0.69459	1451820	-13.68812
8	500	1275210	1283310	-0.63519	1392330	-9.18437
8	1000	2422100	2438550	-0.67916	2512870	-3.74757
8	1000	2524530	2537050	-0.49593	2525530	-0.03961
8	1000	2556320	2570300	-0.54688	2597830	-1.62382
8	1000	2476960	2488850	-0.48002	2855370	-15.27719
8	1000	2497560	2613440	-4.63973	2625750	-5.13261
8	1500	3500160	3710420	-6.00715	3666300	-4.74664
8	1500	3802180	4045590	-6.40185	3802160	<b>0.00053</b>
8	1500	3603240	3638100	-0.96746	3607810	-0.12683
8	1500	3516560	3536890	-0.57812	3663160	-4.16885
8	1500	3705790	3940010	-6.32038	3705030	<b>0.02051</b>

tivo em questão de tempo de resolução, não tão significativa quanto a R1, justamente por ser considerado um critério menos aprimorado que a primeira redução e pela maior quantidade de variáveis mantidas no espaço de estados.

Na Tabela 8 são apresentados os comparativos de valores entre a geração de colunas e a heurística (geração de colunas com R1+R2), apresentado: Garagens (coluna 1); Viagens (coluna 2); Tempo CG (coluna 3) Tempo Heurística (coluna 4); GAP de tempo (coluna 5); Valor CG (coluna 6); Valor Heurística (coluna 7) e; GAP do Valor (coluna 8). Considerando os resultados inferiores obtidos pela redução R2, em termos de desempenho e tempos de execução, em relação à R1 e R1+R2, esta solução não foi utilizada nas análises dos experimentos.

Para a validação da heurística, comparações feitas demonstram bons resultados

Tabela 7 – Comparativo de tempo entre geração de colunas com e sem reduções

Instância		Tempo (seg.)				
K	T	CG	R1	GAP (%)	R2	GAP (%)
4	500	34.71	3.54	<b>89.80121</b>	17.97	<b>48.22818</b>
4	500	43.75	3.84	<b>91.22286</b>	24.53	<b>43.93143</b>
4	500	36.81	3.63	<b>90.13855</b>	20.72	<b>43.71095</b>
4	500	46.05	3.89	<b>91.55266</b>	19.10	<b>58.52334</b>
4	500	33.08	4.17	<b>87.39420</b>	17.59	<b>46.82588</b>
4	1000	443.01	25.25	<b>94.3004</b>	248.47	<b>43.91322</b>
4	1000	760.54	23.73	<b>96.87988</b>	629.22	<b>17.26744</b>
4	1000	509.14	33.48	<b>93.42424</b>	418.21	<b>17.86001</b>
4	1000	525.05	41.93	<b>92.01423</b>	453.13	<b>13.69922</b>
4	1000	541.22	41.48	<b>92.33583</b>	380.15	<b>29.76054</b>
4	1500	1700.53	213.74	<b>87.43104</b>	1077.65	<b>36.62892</b>
4	1500	2542.24	99.86	<b>96.07197</b>	1373.5	<b>45.97286</b>
4	1500	1970.57	90.67	<b>95.39880</b>	1234.44	<b>37.35626</b>
4	1500	2938.56	68.85	<b>97.82690</b>	1359.89	<b>57.07811</b>
4	1500	2138.94	114.52	<b>94.64597</b>	1178.62	<b>44.89721</b>
8	500	50.04	8.24	<b>83.53317</b>	26.84	<b>46.36291</b>
8	500	46.93	6.85	<b>85.40379</b>	6.85	<b>85.40379</b>
8	500	39.42	9.06	<b>77.01674</b>	22.41	<b>43.15068</b>
8	500	37.72	9.75	<b>74.15164</b>	20.57	<b>45.46660</b>
8	500	41.77	8.9	<b>78.69284</b>	28.72	<b>31.24252</b>
8	1000	645.19	57.94	<b>91.01970</b>	310.37	<b>51.89479</b>
8	1000	534.21	85.14	<b>84.06245</b>	318.36	<b>40.40546</b>
8	1000	502.22	85.44	<b>82.98754</b>	268.88	<b>46.46171</b>
8	1000	547.73	95.52	<b>82.56075</b>	151.11	<b>72.41159</b>
8	1000	536.13	64.95	<b>87.88540</b>	17.22	<b>96.78809</b>
8	1500	2671.23	189.29	<b>92.91375</b>	1262.85	<b>52.72403</b>
8	1500	1822.27	309.15	<b>83.03490</b>	1465.22	<b>19.59369</b>
8	1500	1859.01	125.99	<b>93.22273</b>	1303.71	<b>29.87073</b>
8	1500	2598.23	240.99	<b>90.72484</b>	1433.81	<b>44.81589</b>
8	1500	1635.48	269.65	<b>83.51249</b>	1095.65	<b>33.00744</b>

Tabela 8 – Comparativo de valores de geração de colunas e heurística

Instância		Tempo (seg.)			Valor		
K	T	CG	Heurística	<i>GAP</i> (%)	CG	Heurística	<i>GAP</i> (%)
4	500	34.71	19.56	<b>43.64736</b>	1288670	1290260	-0.12338
4	500	43.75	19.11	<b>56.32000</b>	1242620	1253930	-0.91017
4	500	36.81	20.7	<b>43.76528</b>	1283350	1295900	-0.97791
4	500	46.05	24.96	<b>45.79805</b>	1258460	1260900	-0.19389
4	500	33.08	17.42	<b>47.33978</b>	1316440	1317680	-0.09419
4	1000	443.01	275.53	<b>37.80502</b>	2515500	2517270	-0.07036
4	1000	760.54	342.93	<b>54.91008</b>	2412620	2415490	-0.11896
4	1000	509.14	297.37	<b>41.59401</b>	2454400	2455340	-0.03830
4	1000	525.05	305.46	<b>41.82368</b>	2489160	2489940	-0.03134
4	1000	541.22	233.04	<b>56.94172</b>	2517880	2529250	-0.45157
4	1500	1700.53	1257.87	<b>26.03110</b>	3828630	3831090	-0.06425
4	1500	2542.24	1423.44	<b>44.00846</b>	3560800	3581350	-0.57712
4	1500	1970.57	1118.72	<b>43.22867</b>	3647470	3658740	-0.30898
4	1500	2938.56	1299.6	<b>55.77425</b>	3407790	3433010	-0.74007
4	1500	2138.94	1243.57	<b>41.86067</b>	3566950	3574050	-0.19905
8	500	50.04	33.81	<b>32.43405</b>	1292030	1293850	-0.14086
8	500	46.93	29.13	<b>37.92883</b>	1276300	1277760	-0.11439
8	500	39.42	32.04	<b>18.72146</b>	1303900	1304600	-0.05369
8	500	37.72	32.14	<b>14.79321</b>	1277020	1279020	-0.15661
8	500	41.77	28.62	<b>31.48192</b>	1275210	1278000	-0.21879
8	1000	645.19	373.60	<b>42.09458</b>	2422100	2422960	-0.03551
8	1000	534.21	322.71	<b>39.59117</b>	2524530	2537000	-0.49395
8	1000	502.22	331.88	<b>33.91741</b>	2556320	2556580	-0.01017
8	1000	547.73	373.31	<b>31.84416</b>	2476960	2488300	-0.45782
8	1000	536.13	286.66	<b>46.53162</b>	2497560	2520930	-0.93571
8	1500	2671.23	1461.66	<b>45.28139</b>	3500160	3512330	-0.34770
8	1500	1822.27	1331.71	<b>26.92027</b>	3802180	3799460	<b>0.07154</b>
8	1500	1859.01	1330.42	<b>28.43395</b>	3603240	3608140	-0.13599
8	1500	2598.23	1803.41	<b>30.59083</b>	3516560	3515970	<b>0.01678</b>
8	1500	1635.48	1386.524	<b>15.22220</b>	3705790	3706700	-0.02456

em termos de aceleração na busca do melhor resultado, aplicando as duas reduções e a geração de colunas modificada. Para todas as instâncias, a diferença ficou com um valor inferior a 1%. Sendo assim, o ganho obtido com as reduções demonstram que a heurística trabalha de forma eficiente em tempos de execução, sem perdas significativas no resultado ótimo. Comparando as informações, deduz-se que as reduções, trabalhando de forma conjunta, equilibram tempo e resultado, em relação à utilização das reduções de forma individual. Caso busca-se um desempenho de tempo, a redução R1 acelera o tempo de resposta com uma baixa perturbação no melhor valor.

A estrutura desenvolvida para heurística, que envolve a simplificação do problema por meio da redução do espaço de estados e pelas melhorias aplicadas na geração de colunas, fornece um embasamento para a resolução de instâncias com tamanhos maiores dos validados, abordado a seguir, na seção 5.5. Como colocado acima, a redução R2 não foi considerada nestes experimentos.

## 5.5 EXPERIMENTOS COMPUTACIONAIS

Considerando os resultados de otimizações apresentadas nas seções anteriores, experimentos computacionais buscando medir a capacidade para resolver problemas maiores foram executados, afim de aferir a capacidade de resolução de problemas MDVSP por parte da heurística.

Para tanto, foram utilizadas instâncias do tipo arco de conexões, envolvendo até  $|N| \in \{2000, 3000, 5000, 10000\}$  viagens e, em até  $|K| = \{4, 8, 16, 32\}$  garagens foram rodadas, as instâncias são parte do trabalho de GUEDES; BORENSTEIN (2013), com o intuito de verificar os resultados obtidos e validar o processamento para grandes instâncias. As instâncias são geradas de forma diferente das instâncias de CARPANETO *et al.* (1989). As frequências de viagens são definidas de acordo com demandas, utilizando uma função de distribuição gaussiana, Detalhes do processo de geração das instâncias é fornecido em GUEDES; BORENSTEIN (2013).

O objetivo destes experimentos é de avaliar o comportamento das heurísticas R1 e R1+R2 para instâncias superiores a de PEPIN; DESAULNIERS; HERTZ (2008). Os mesmos parâmetros utilizados na verificação foram empregados para esta avaliação, sendo que, R2 foi descartada por, sozinha, ser menos eficiente que a

heurística completa R1+R2 e, menos eficiente que R1. Buscou-se utilizar as mesmas calibrações realizadas para a verificação e validação, porém, não foram executadas várias rodadas com calibrações diferentes, pelo intuito de demonstrar a capacidade de resolução do problema.

A seguir, na tabela 9 são apresentados os resultados dos experimentos computacionais com a heurística. Apresentando as seguintes informações: Garagens (coluna 1); Viagens (coluna 2); Tempo de execução da heurística (coluna 3); Tempo de execução da R1 (coluna 4); *GAP* de tempo (coluna 5); melhor valor da heurística (coluna 6); melhor valor de R1; (coluna 7) e; *GAP* de valores (coluna 8).

**Tabela 9 – Experimentos computacionais com a heurística**

Instância		Tempo (seg)			Valor objetivo		
K	T	Heur.	R1	<i>GAP</i> (%)	Heur.	R1	<i>GAP</i> (%)
4	2000	274.38	213.2	<b>22.29754</b>	8929640	8575900	<b>3.96141</b>
8	2000	776.70	407.29	<b>47.56147</b>	7921620	8110400	-2.38309
16	2000	1147.01	623.59	<b>45.63342</b>	7771100	7885530	-1.47250
32	2000	894.74	790.92	<b>11.60337</b>	8858380	8042840	<b>9.20642</b>
4	3000	497.63	396.59	<b>20.30424</b>	13208600	12531200	<b>5.12847</b>
8	3000	1340.99	1204.52	<b>10.17680</b>	13342900	12469700	<b>6.54430</b>
16	3000	3557.17	1180.83	<b>66.80422</b>	11733200	12259500	-4.48556
4	5000	2032.47	1584.45	<b>22.04312</b>	22325800	21423500	<b>4.04151</b>
4	10000	17706.35	12434.10	<b>29.77604</b>	50305900	47512900	<b>5.55203</b>

Os resultados esperados na tabela 9 são de que as instâncias de tamanho de até 10000 conseguiram ser resolvidas pela heurística desenvolvida, indicando uma capacidade de resolução maior que a encontrada atualmente na literatura. Também, problemas com um número maior de garagens foi resolvido, indicando a possibilidade de resolução de problemas que podem ser considerados mais complexos, mesmo sem a adição de mais viagens.

Como pode ser visto, o comportamento da heurística, utilizando-se das duas reduções, consegue resolver o problema, porém, aplicando apenas a redução R1, o resultado é obtido de forma mais rápida, porém, com uma pequena diferença em relação ao valor da heurística completa, sendo que, em alguns casos, a diferença é até positiva em relação ao valor da heurística. Observa-se que, as duas resoluções utilizaram exatamente as mesmas calibrações para cada instância, portanto, os valores obtidos ainda podem ser melhorados, caso fossem aplicadas melhorias nos valores das calibrações, para cada instância em específico.



## 5.6 DISCUSSÕES

Esta seção aborda discussões acerca dos resultados obtidos, tanto nas validações quanto nos experimentos computacionais. Para a validação da geração de colunas truncada em relação aos valores de PEPIN; DESAULNIERS; HERTZ (2008), obteve-se uma média de tempo superior de 31.92% e uma média de melhores valores da função objetivo de 0.01855%, indicando que a geração de colunas truncado modificada consegue obter melhores valores. Não foram feitas análises de tempo, pois existe uma diferença de capacidade de processamento, por parte do *hardware* utilizado entre os trabalhos.

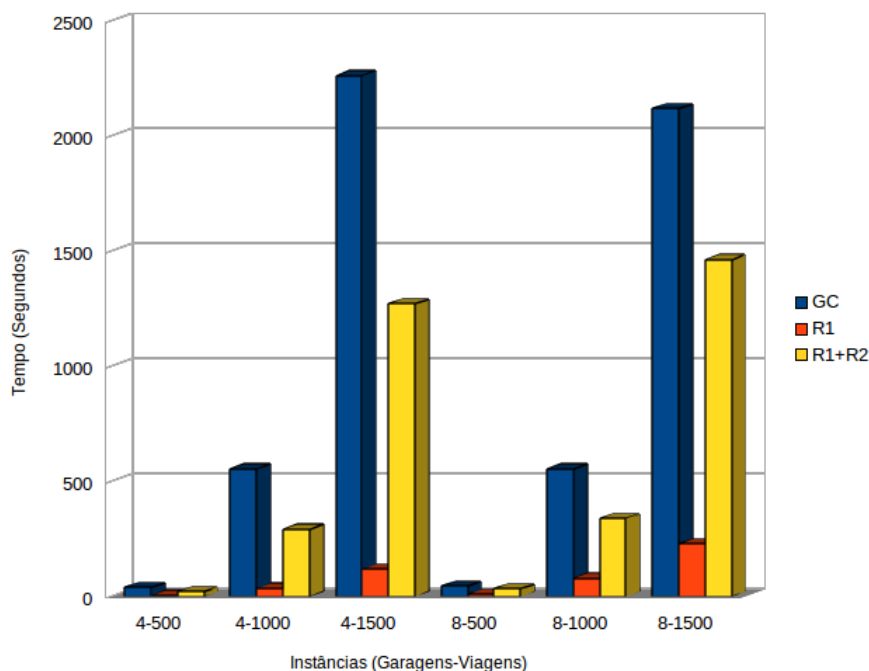
Para a validação das reduções, em comparação aos resultados validados da geração de colunas truncado modificado, para a redução R1 com a geração de colunas, houve uma média de ganhos de tempo considerável, sendo de 88.6967%. A diferença média do resultado foi de  $-2.41605\%$ , sendo que, a pior diferença obtida foi de  $-8.07441\%$ , portanto, para soluções que necessitam de tempo de resposta rápida, a redução R1 é recomendada a ser utilizada.

Para a redução R2 com a geração de colunas, também houve uma média de ganhos de tempo considerável, sendo de 44.06326%, porém, a diferença média foi de  $-6.07055\%$  e, tendo, em sua pior situação, uma diferença de  $-16.01574\%$ , portanto, a utilização desta redução sozinha não foi considerada como uma boa opção, portanto, esta redução de forma individual não analisada de forma mais profunda, somente de forma conjunta com R1.

Para a validação da heurística (as duas reduções com a geração de colunas), também foi analisado em comparação com a geração de colunas truncada modificada, resultando em uma média comparativa de tempo de 38.51041% e uma diferença de  $-0.26457\%$ , indicando um equilíbrio entre tempo e melhor valor. A aplicação das duas reduções em conjunto com a geração de colunas obtém um valor com diferenças extremamente pequenas e, com ganhos consideráveis de tempo, sendo que, na pior situação, a diferença foi de  $-0.97791\%$ . Esta situação indica que a heurística completa pode ser utilizada em problemas MDVSP mais complexos, onde um equilíbrio entre tempo de resolução e resultado seja importante.

Na figura 7 é ilustrado um gráfico comparando resultados e tempos de resolução entre a geração de colunas, as reduções e a heurística. Com isso, pode-se notar

Figura 7 – Comparativo de tempos entre as resoluções



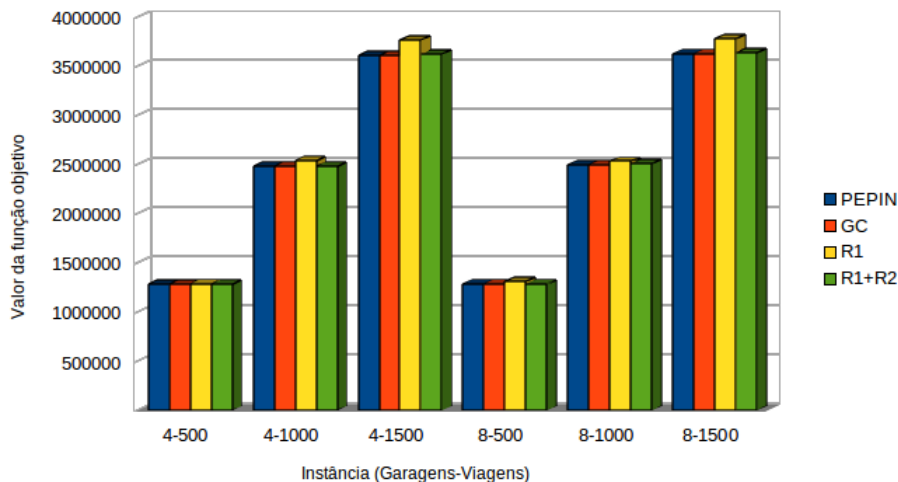
um ganho considerável de desempenho de tempo para as reduções, destacando-se o ganho de desempenho ao utilizar a primeira redução.

Na figura 8 é ilustrado graficamente a média dos melhores valores para cada conjunto de instâncias, demonstrando que, na média geral, a geração de colunas modificada forneceu melhores resultados em relação aos números fornecidos por Huisman. A redução R1 se comportou de forma a fornecer resultados bem próximos, mesmo que o tempo de resolução diminuiu consideravelmente, sendo que, a média de valores em comparação aos resultados de Huisman ficou em  $-0.23516\%$ . Considerando-se esta diferença diminuta e o baixo tempo de resolução, reforça-se a indicação de R1 como uma excelente opção para uma eficiente solução do MDVSP.

Observando-se as figuras 7 e 8, constata-se que a R1+R2 oferece resultados um pouco melhores que R1, porém com tempos bem superiores, em média de  $86.28449\%$ , portanto, sugere-se a utilização de R1+R2 quando se necessita uma maior precisão do resultado do melhor valor, pois R1 resolveu o problema, em média, 7.29 vezes mais rápido que a geração de colunas com o espaço de estados original.

Quanto aos experimentos computacionais, demonstrou-se a capacidade de resolução de problemas de tamanhos de 10000 viagens, enquanto que trabalhos anteriores

Figura 8 – Comparativo de melhores valores entre as resoluções



envolviam a resolução de 1500 viagens. PEPIN; DESAULNIERS; HERTZ (2008) resolveu problemas de 1500 viagens e 8 depots - em média - em 3085 seg., a heurística resolveu problemas de 5000 viagens e 4 garagens em 2033 seg. e, a redução R1 resolveu em 1585 seg., indicando uma boa capacidade de resolução de problemas maiores. Para a instância que contém 10000 viagens e 4 garagens, o tempo de resolução ficou em 17706.35 seg. para a heurística e, em 12434.10 seg. para R1, indicando assim, uma capacidade de resolução de grandes instâncias, podendo assim, serem utilizadas para resolver problemas de grandes malhas logísticas.

Em relação à heurística, notou-se que, ao adicionar mais garagens, o tempo de resolução tinha uma equivalência em relação ao total, demonstrando assim, a importância de um algoritmo especialista que seja rápido e eficiente, para resolver o problema do caminho mais curto, a exemplo do algoritmo GOR, desenvolvido por CHERKASSKY; GOLDBERG; RADZIK (1996).

## 6 CONCLUSÃO

Esta dissertação desenvolveu uma heurística simples e objetiva para a solução do *Multiple Depot Vehicle Scheduling Problem*. A heurística é baseada em duas fases: Na primeira fase, através de duas técnicas, retira-se as possibilidades de viagens que tem poucas ou nenhuma chance de pertencer à solução ótima, assim, reduzindo o espaço de estados; Na segunda fase, resolve-se o problema com a técnica de geração de colunas truncado modificado, de forma a acelerar o processo de escolha de melhores caminhos.

Para a resolução do MDVSP, a técnica de geração de colunas desenvolvida com base no trabalho de PEPIN; DESAULNIERS; HERTZ (2008), mais o estudo e busca do aprimoramento da convergência através de aplicação de um relaxamento, permitiu resolver o problema do MDVSP equilibrando resultados e tempos de resolução.

Avaliações foram feitas, demonstrando a validação da geração de colunas, por meio de comparações de resultados de instâncias consolidadas por pesquisas anteriores, os resultados permitiram verificar e aferir os resultados, sendo assim, indicado que a técnica de geração de colunas relaxado atende às expectativas de resolução do MDVSP.

A técnica de redução do espaço de estados também foi desenvolvida e melhorada com sucesso. A primeira redução - melhores caminhos dos VSPs - foram adaptados algoritmos consolidados para a redução de possibilidades de viagens, bem como, para a segunda redução - escolha de melhores valores por critério estatístico - por meio de uma seleção de melhores valores. Ambos permitiram acelerar o processo de resolução do MDVSP, inclusive, em alguns casos, fornecendo melhores valores.

Caso a exigência de tempo seja o item mais importante, a resolução do MDVSP por meio da redução por melhores caminhos fornece uma forma rápida de retorno de

resolução, penalizando pouco o melhor valor, se comparado ao ganho na velocidade de resolução do problema.

Caso haja necessidades de um melhor valor mais aprimorado, a utilização das duas reduções em conjunto ainda trazem uma melhoria no tempo de resolução em relação à geração de colunas sem reduções, fornecendo assim, um melhor valor muito próximo ao ideal. A utilização das duas reduções, portanto, equilibram entre a busca do melhor valor e o tempo de resolução. As avaliações demonstraram bons resultados para a aplicação da heurística completa, indicando um melhoria no tempo de resposta, oferecendo diferenças plenamente aceitáveis - abaixo de 1% - assim, fundamentando a validação da heurística.

A resolução de instâncias de tamanhos maiores aos que encontrados atualmente na literatura, foram corroboradas por meio de experimentos computacionais, indicando a capacidade de solucionar problemas maiores. Pelos aprimoramentos e pelo atendimento dos objetivos específicos, é concluso que o objetivo geral do trabalho foi plenamente atendido, buscando resolver o problema MDVSP em tempo aceitável. Para tanto, a heurística pode ser considerada como uma base para ser utilizada em outros problemas, no qual o relevante é uma solução de compromisso entre tempo e precisão, como exemplo, *disruption management*, *crew scheduling* e *real time assignment*.

Como sugestão de trabalhos futuros indica-se, o aprimoramento da redução do espaço de estados, como tentativas de perturbar os valores selecionados pelos VSPs, de forma a criar mais opções com grandes chances de pertencer à solução ótima. Também, sugere-se avanços em melhorias na técnica de geração de colunas, estudando formas de relaxamento para acelerar a convergência, bem como, desenvolver aprimoramentos para a estabilização de valores.

Também, como sugestão nota-se a possibilidade de utilização de outros tipos de redes, como a rede tempo-espaço, conforme abordado no trabalho de KLIEWER; MELLOULI; SUHL (2005). Também, na busca da otimização, a aplicação de um algoritmo especialista de caminhos mais curto.

## REFERÊNCIAS

- ANDRADE, E.L. de. **Introdução à Pesquisa Operacional: Métodos e Modelos para a Análise de Decisão**. 4<sup>a</sup>. ed., Rio de Janeiro: LTC, 2009.
- ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. **Pesquisa Operacional para Cursos de Engenharia**. Rio de Janeiro: Elsevier, 2007.
- AUGUSTINE, J. E.; SEIDEN, S. Linear Time Approximation Schemes for Vehicle Scheduling Problems, **Theoretical Computer Science**, 324, 147-160, 2004.
- BAITA F.; PESENTI, R.; UKOVICH, W; FAVARETTO, D. A Comparison of Different Solution Approaches to the Vehicle Scheduling Problem in a Practical Case **Computers & Operations Research**, 27, 1249-1269, 2000.
- BELLMAN, R. On a Routing Problem. **Quarterly of Applied Mathematics** 16, 87-90, 1958.
- BARNHART, C.; JOHNSON, E.L.; NEMHAUSER, G.L.; SAVELSBERGH, MARTIN W.P.; VANCE, P.H. Branch-and-Price: Column Generation for Solving Huge Integer Programs. **School of Industrial and Systems Engineering**, Georgia Institute of Technology, 1996.
- BERTOSSI, A.A.; CARRARESI, P.; GALLO, G. On Some Matching Problems Arising in Vehicle Scheduling Models. **Networks**, 17, 271-281, 1987.
- BERTSEKAS, D. P. A Simple and Fast Label Correcting Algorithm for Shortest Paths. **Networks**, 23, 703-709, 1993.
- BIANCO, L; MINGOZZI, A.; RICCIARDELLI, S. A Set Partitioning Approach

to the Multiple Depot Vehicle Scheduling Problem, **Optimization Methods and Software**, 3, 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 & Operations Research**, 10, 63-211, 1983.

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

BUNTE S.; KLIEWER, N. An Overview on vehicle scheduling problems. **Public Transportation**, 1, 299-317, 2010.

CARPANETO, G.; DELL'AMICO, M; TOTH, P. A Branch-and-bound Algorithm for the Multiple Depot Vehicle Scheduling Problem. **Networks.**, 19, 531-548, 1989.

CHERKASSKY, B.V.; GOLDBERG, A.V.; T. RADZIK Shortest Paths Algorithms: Theory and Experimental Evaluation. **Math Programming**, 73, 129-174, 1996.

DANTZIG, G. B.; WOLFE, P. The Decomposition Algorithm For Linear Programs. **Econometrica**, 29, 4, 1961.

DELL'AMICO, M.; FISCHETTI, M.; TOTH, P. Heuristic Algorithms for the Multiple Depot Vehicle Scheduling Problem **Management Science.**, 39 (1), 115-125, 1993.

DESAULNIERS, G.; LAVIGNE J.; SOUMIS F. Multi-depot Vehicle Scheduling Problems with Time Windows and Waiting Costs **European Journal of Operational Research.**, 11, 479-494, 1998.

DESROSIERS, J.; DUMAS, Y.; SOLOMON, M. M.; SOUMIS, F. Time Constrained Routing and Scheduling. Em: Ball, M.O., Magnanti, T.L., Monma, C.L. & Nemhauser, G.L. (Eds.) Handbooks in operations research and management science. 8, 35-139, 1995.

DESROCHERS, M.; SOUMIS F. A Column Generation Approach to the Urban Transit Crew Scheduling Problem. **Transportation Science**, 23 (1), 1-13, 1989.

- FORD, J.; LESTER R. **Network Flow Theory** Paper, 293, Santa Monica, California: RAND Corporation, 1956.
- FRELING, R.; WAGELMANS A.P.; PAIXÃO J.M. Models and Algorithms for Single-Depot Vehicle Scheduling. **Transportation Science**. 3(2), 165-180, 2001.
- GAITHER, N.; FRAZIER, G.; SANTOS, J. C. **Administração da Produção e Operações**. São Paulo: Pioneira, 2001.
- GAVISH, B.; SHLIFER, E. An Approach for Solving a Class of Transportation Scheduling Problems. **European Journal of Operations Research**, 72, 115-124, 1979.
- GINTNER, V.; KLIEWER N.; SUHL, L. Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice **OR Spectrum**, 2005.
- GUEDES. P.; BORENSTEIN. D. Otimização do Problema de Escalonamento de Veículos no Transporte Urbano de Passageiros com Múltiplas Garagens Utilizando a Rede Tempo-Espaço. **Anais da XLV Sobrapo**, Natal, Brasil, 2013.
- HADJAR, A.; SOUMIS, F. Dynamic Window Reduction for the multiple depot vehicle scheduling problem with time windows **Computer & Operations Research**., 36, 2160-2172, 2009.
- JONKER, R.; VOLGENANT, A. A Shortest Path Algorithm for Dense and Sparse Linear Assignment Problems. **Computing**, 38, 325-340, 1987.
- KANG, K.W.; LEE, Y.H.; LEE B.K. ICCSA **An Exact Algorithm for Multi Depot and Multi Period Vehicle Scheduling Problem**., 350-359, 2005.
- KIM, J.; KIM, Y. A decomposition approach to a multi-period vehicle scheduling problem **Omega, International Journal of Management Science**, 27, 421-430, 1999.
- KLIEWER, N.; MELLOULI, T.; SUHL, L. A time-space network based exact optimization model for multi-depot-bus scheduling **European Journal of Operations Research**., 175, 1616-1627, 2005.



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

LAURENT, B.; HAO, J. Iterated local search for the multiple depot vehicle scheduling problem. **Computers & Industrial Engineering**, 2009.

LÖBEL, A. Vehicle Scheduling in Public Transit and Lagrangean Pricing. **Management Science**, 44 (12), 1637-1649, 1998.

MARTINS, G A. **Estatística Geral e Aplicada**. São Paulo: Atlas, 2002.

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. **Computer-aided transit scheduling**, 386, 167-180, 1992.

ORLOFF, C.S. Route constrained fleet scheduling. **Transportation Science**, 10, 149-168. 1972.

OUKIL, A.; AMOR, H.B.; DESROSIERS, J.; GUEDDARI, H.E. Stabilized column generation for highly degenerate multiple-depot vehicle scheduling problems. **Computers & Operations Research**, 34, 817-834, 2006.

PEPIN, A.; DESAULNIERS, G.; HERTZ, A. A comparison of five heuristics for the multiple depot vehicle scheduling problem. **Journal Of Scheduling** 12, 17-30, 2008.

PINTO, J.M.; BRANCO, I.M. A quasi-assignment Algorithm for Bus Scheduling **Networks** 17, 249-269, 1987.

RIBEIRO, C.C.; SOUMIS, F. A column generation approach to the multiple-depot vehicle scheduling problem **Operations Research** 42 (1), 41-52, 1994.

ROHDE, L.R. **Desenvolvimento de Heurística para Solução do Problema de Escalonamento de Veículos com Múltiplas Garagens** 2008. 122p. Tese (Doutorado em Administração) — Escola de Administração, Universidade Federal do Rio Grande do Sul, Porto Alegre.

SAHA, J.L. An algorithm for bus scheduling problems. **Operations Research**, 21, 463-474, 1972.