

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ADMINISTRAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO - PPGA/UFRGS

Pablo Cristini Guedes

**Modelagem do Problema de Escalonamento
de Veículos com Múltiplas Garagens usando
Rede Tempo-Espaço: Grandes Instâncias e
Frota Heterogênea**

Porto Alegre, Fevereiro de 2014

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Guedes, Pablo Cristini

Modelagem do Problema de Escalonamento de Veículos com Múltiplas Garagens usando Rede Tempo-Espaço: Grandes Instâncias e Frota Heterogênea / Pablo Cristini Guedes. - 2014.

82 f.

Orientador: Denis Borenstein

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Administração - PPGA/UFRGS, Porto Alegre, BR-RS, 2014.

Palavras-chave: MDVSP, rede tempo-espaço, geração de colunas, MDVTSP.

Pablo Cristini Guedes

**Modelagem do Problema de Escalonamento
de Veículos com Múltiplas Garagens usando
Rede Tempo-Espaço: Grandes Instâncias e
Frota Heterogênea**

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

Orientador: Prof. Dr. Denis Borenstein

Porto Alegre, Fevereiro de 2014

- “restate my assumptions:*
- 1. Mathematics is the language of nature.*
 - 2. Everything around us can be represented and understood through numbers.*
 - 3. If you graph these numbers, patterns emerge.”*
- MAX COHEN. (PI, 1998)

AGRADECIMENTOS

É difícil acreditar que estou findando esta jornada que iniciei lá em março de 2012. Naquela época me encontrava formado em Ciência da Computação e trabalhando com programação em uma empresa, porém insatisfeito, pois meu sonho e minha vocação é ensinar. Quando lá no distante Pré-Cálculo de 2008, quando a Professora Luisa Doering me convidou (agradeço muitíssimo a ela por isso e pelas cartas de recomendação, este trabalho tem sua colaboração), ela despertou em mim essa certeza de que nasci para lecionar, pois fui, desde jovem, sedento por conhecimento e saber, mas nunca egoísta com ele. O conhecimento pode ser a moeda do século XXI, mas ele é por si só um bem da humanidade que deve ser compartilhado e não monopolizado, até porque quando distribuído, ele se multiplica e cria-se novos conhecimentos. Hoje termino com sucesso o que há 2 anos era um risco, trocar de área, sair da computação e ir para a administração, mas tenho certeza que consegui alcançar meu objetivo com essa decisão. Fui bem recebido por meus colegas e professores, que sempre me deram a energia e palavras que precisei.

Portanto agradeço, primeiramente, à minha família, especialmente ao meus pais e minha irmã, que sempre me apoiaram e me deram suporte para conseguir chegar até aqui.

Agradeço à Deus por me dar um outro presente durante o mestrado que foi a minha namorada Marina, que foi bastante importante na parte final deste trabalho. Dedico a ela também esse trabalho.

Quero agradecer, especialmente ao Professor Denis, pois além de meu orientador, ele foi meu amigo, sempre com seu jeito honesto, contribuiu não somente para minha formação profissional de Mestre em Administração, mas também na minha constante evolução como ser. Agradeço ao Denis pelo desafio que é esse trabalho e pela paciência

necessária que ele teve comigo.

Gostaria de agradecer também aos membros da melhor turma de LDM que o PPGA já viu, na minha opinião: Leonardo, Guilherme, Gabriel e Alberto. Obrigado pessoal pela companhia, pelos trabalhos que realizamos juntos. Pelo apoio desprendido e pelos ensinamentos passados nessa jornada.

Ao Camilo, Leonardo e Guilherme, agradeço por tudo, pois além de colegas vocês se tornaram amigos, são portanto um presente que o mestrado me deu.

Agradeço também ao grupo de *scheduling* em transporte: aos colegas Monize e William, que compartilharam uma enorme parte deste trabalho, sempre me auxiliando e ajudando a desmistificar os artigos. Extendo um abraço aos colegas Everton e Juan, que por não ter os tópicos tão próximos não pude conviver tanto, mas são excelentes pessoas.

Ao professor Olinto, que auxiliou o grupo e suas contribuições, de alguma forma, estão presentes nesse trabalho.

Gostaria de agradecer à professora Luciana Buriol, pois sempre foi uma professora que me ajudou muito, desde os tempos de computação.

Um agradecimento a todos os professores do PPGA de Métodos Quantitativos e Sistemas de Informação pelos ensinamentos, em especial aos membros da banca de proposta professor Becker e professora Denise cujas contribuições fizeram com que este trabalho evoluísse.

Agradeço ao Prof. Gustavo Peixoto Silva que contribuiu com a melhoria deste trabalho através de suas contribuições na banca.

Ao CNPq pela ajuda financeira, pois sem a bolsa este trabalho não chegaria a este nível de excelência.

A secretária do PPGA Dejanira, por sempre estar solícita e atender minhas demandas com um sorriso no rosto.

Enfim agradeço a todos que de alguma forma contribuíram para esse trabalho.

Um Abraço.

SUMÁRIO

1 INTRODUÇÃO	17
1.1 OBJETIVO GERAL	19
1.2 OBJETIVOS ESPECÍFICOS	19
1.3 ORGANIZAÇÃO DA DISSERTAÇÃO	19
2 REVISÃO DA LITERATURA	20
2.1 DEFINIÇÕES	20
2.2 ESCALONAMENTO DE VEÍCULOS COM MÚLTIPLAS GARAGENS	21
2.2.1 Formulação do MDVSP	21
2.2.2 Revisitando o MDVSP	23
2.2.3 Formulando o MDVTSP	28
2.2.4 Conhecendo um pouco o MDVTSP	29
2.3 REDE TEMPO-ESPAÇO	33
2.3.1 Formulação da Rede Tempo-Espaço	33
2.4 MÉTODO DE GERAÇÃO DE COLUNAS	35
3 MÉTODO	39
3.1 ETAPA 1: DEFINIÇÃO DO PROBLEMA	39
3.2 ETAPA 2: MODELAGEM	40
3.2.1 Modelo de Rede Tempo-Espaço	40
3.2.2 Formulação do MDVSP	41
3.2.3 Formulação do MDVTSP	41
3.3 ETAPA 3: MÉTODO DE SOLUÇÃO	42
3.4 ETAPA 4: VERIFICAÇÃO DOS MODELOS	44

3.5 ETAPA 5: EXPERIMENTAÇÃO	44
4 RESOLVENDO O MDVSP	45
4.1 GERAÇÃO E REDUÇÃO DA REDE TEMPO-ESPAÇO	45
4.2 CONVERSOR DE TSN PARA CONEXÃO	48
4.3 GERAÇÃO DE COLUNAS TRUNCADA MODIFICADA	49
5 RESOLVENDO O MDVTSP	53
5.1 FORMULAÇÃO DE PARTIÇÃO DE CONJUNTOS PROPOSTA PARA O MDVTSP	53
5.2 GERAÇÃO E REDUÇÃO DA REDE TEMPO-ESPAÇO	54
5.3 CONVERSOR DE TSN PARA CONEXÃO	54
5.4 GERAÇÃO DE COLUNAS MODIFICADA PARA O MDVTSP	56
6 EXPERIMENTAÇÃO COMPUTACIONAL	60
6.1 GERADOR DE INSTÂNCIAS	60
6.2 CALIBRAGEM DOS PARÂMETROS	62
7 SOLUCIONANDO O MDVSP	67
7.1 SOLUÇÃO ÓTIMA COM TSN	67
7.2 MÉTODO DE GERAÇÃO DE COLUNAS TRUNCADO MODIFICADO	69
8 SOLUCIONANDO O MDVTSP	70
8.1 MÉTODO DE GERAÇÃO DE COLUNAS TRUNCADO MODIFICADO	71
9 CONCLUSÕES	74
9.1 SÍNTESE DAS CONTRIBUIÇÕES	75
9.2 TRABALHOS FUTUROS	75
REFERÊNCIAS	77

LISTA DE FIGURAS

2.1	Exemplo de Rede de Conexão Instanciada.	23
2.2	Exemplo de Rede Tempo-Espaço	35
3.1	Fluxograma Simplificado do Método de Solução.	43
4.1	Estágios de Agregação dos Arcos Expressos	46
4.2	Estágios de Agregação dos Vértices de Arcos <i>Pull In-Out</i>	47
5.1	Gráfico da relação dos custos com os fatores.	56
6.1	Distribuição de Demandas.	61
6.2	Calibrando os Parâmetros: Z_{min} igual a zero.	64
6.3	Calibrando os Parâmetros: Ω_{min} igual a 0,8.	65

LISTA DE TABELAS

2.1	Tabela de Revisão Literária para o MDVSP	24
2.2	Tabela de Revisão Literária para o MDVTSP	30
3.1	Exemplo de Tabela de Horários	39
4.1	Parâmetros do Método de Geração de Colunas	50
4.2	Tempo Gastos na Resolução dos Subproblemas	52
5.1	Exemplo de Geração de Fatores de Custos	56
5.2	Análise sobre o Uso do Relaxamento na Restrição (3.8)	57
6.1	Limites dos Parâmetros do Método de Geração de Colunas	62
6.2	Calibragem dos Parâmetros Usados na Geração de Colunas - Parte 1. . .	63
6.3	Calibragem dos Parâmetros Usados na Geração de Colunas - Parte 2. . .	64
6.4	Calibragem dos Parâmetros Usados na Geração de Colunas - Parte 3. . .	66
7.1	Solução Ótima Utilizando a Rede Tempo-Espaço	68
7.2	Resultados Usando Geração de Colunas	69
8.1	Custo de cada Tipo de Veículo	70
8.2	Tipos de Veículos	71
8.3	Frota Heterogênea : 3 Tipos de Veículos	71
8.4	Frota Heterogênea : 5 Tipos de Veículos	72
8.5	Frota Heterogênea : 8 Tipos de Veículos	72

LISTA DE ALGORITMOS

1	Conversor de TSN para Conexão	48
2	Método de Geração de Colunas Truncado Modificado	51
3	Conversor de TSN para Conexão com Frota Heterogênea	55
4	Método de Geração de Colunas Truncado Modificado	59
5	Gerador de Viagens	62

LISTA DE ABREVIATURAS E SIGLAS

MDVSP	Multi-Depot Vehicle Scheduling Problem
MDVTSP	Multi-Depot Vehicle-Type Scheduling Problem
RMP	Restricted Master Problem
SDVSP	Single-Depot Vehicle Scheduling Problem
TSN	Time-Space Network

LISTA DE SÍMBOLOS

T	Conjunto que denota o tempo
S	Conjunto que denota o espaço
K	Conjunto de Garagens
F	Conjunto de Tipos de Veículos
$Trips$	Conjunto das viagens de serviço
N	Conjunto de Vértices
A	Conjunto de Arestas
x_{ij}^k	Variável de decisão binária da formulação <i>multicommodity</i> do MDVSP
c_{ij}	Custo do arco (i, j) na formulação <i>multicommodity</i> do MDVSP
Ω^k	Conjunto de todos os caminhos da garagem k
Ω^{kf}	Conjunto de todos os caminhos com o veículo f da garagem k
θ_p	Variável de decisão binária da formulação de particionamento de conjuntos
c_p	Custo do caminho p na formulação de particionamento de conjuntos
a_{ip}	Coefficiente tecnológico que indica que a viagem i está no caminho p
v_k	Capacidade da garagem k
v_{kf}	Número de veículos do tipo f da garagem k
$o(k)$	Nó de inicio de um bloco de veículo de uma garagem k

$d(k)$ Nó final de um bloco de veículo de uma garagem k

q_f Capacidade do veículo f

D_j Demanda da viagem j

RESUMO

O problema de escalonamento de veículos com múltiplas garagens (MDVSP, do inglês Multi-Depot Vehicle Scheduling Problem) é um problema clássico de logística e transportes. O MDVSP também é a base para a solução de vários problemas correlatos, tais como o problema de escalonamento de veículos em tempo-real e soluções integradas com o escalonamento de veículos, tais como o escalonamento da tripulação e otimização da tabela de horários. Desta forma, aprimorar a solução deste problema pode ser considerado de grande relevância, a qual permitirá resolver grandes instâncias reais de forma eficiente, bem como permitir a solução de problemas correlatos. O objetivo desta dissertação é verificar a aplicabilidade da utilização da rede tempo-espaço e do método de geração de colunas modificado proposto, para a solução deste problema, e de sua variante com frota heterogênea, considerando grandes instâncias. Diversos testes foram realizados utilizando o gerador de instâncias aleatórias com base na distribuição de demandas proposto. Grandes instâncias, envolvendo milhares de viagens (entre 500-10.000) e dezenas de garagens (4-128) são resolvidas em tempos razoáveis.

Palavras-chave: MDVSP, rede tempo-espaço, geração de colunas, MDVTSP.

ABSTRACT

The multiple-depot vehicle-scheduling problem (MDVSP) is a classic logistic and transportation problem. The MDVSP is also a subproblem for solving various related problems, such as the real time vehicle scheduling problem, disruption management; and integrated problems such as the vehicle and crew scheduling problems. Although several mathematical and solution method have been developed in the literature, large instances (involving thousands of trips and several depots) are still difficult to solve in a reasonable time. The objective of this research work is to verify the applicability of the use of the space-time network towards obtaining good solutions for large instances in short time. Time-space network was suggested by Kliewer et al (2006), and it is positioned with respect to two-dimensional axes, one representing time and the other one space or stations. The arcs represent deadheading movements; and waiting periods in the same station. Solution methods for the MDVS combining time space with integer linear programming solvers and column generation were developed. Extensive testing was carried out using random generated instances, based on demands distribution. Large instances, involving thousands of trips (between 1,000-10,000) and dozen (4-64) depots, are solved in reasonable times.

Keywords: MDVSP, time-space network, column generation, MDVTSP.

1 INTRODUÇÃO

Com o crescimento populacional, o transporte público é mais demandado, criando problemas de mobilidade urbana como poluição do ar, acidentes e congestionamentos (Farahani *et al.*, 2013). É de conhecimento público que empresas de transportes operam sob fortes pressões de custos. Desta forma, as empresas devem focar no uso eficiente de todos os seus recursos, inclusive o uso dos veículos. Embora a quantidade de estudos torne o escalonamento de veículos um problema clássico, poucos são aqueles que apresentem bons resultados (soluções otimizantes em um curto tempo) para grandes instâncias, principalmente quando adiciona-se o uso de múltiplas garagens ou frota heterogênea.

O problema de escalonamento de veículos consiste em atribuir um veículo a uma viagem com horário anteriormente definido. Cada viagem pode ser atendida por um veículo pertencente a uma determinada frota de veículos. Cada veículo tem de começar e terminar o seu dia de trabalho em uma das garagens. Depois de atender uma viagem (carregado), cada veículo pode atender uma das viagens posteriores a partir do terminal de onde o veículo está parado, ou ele pode mudar sua localização, movendo-se vazio para um outro terminal, a fim de atender a próxima viagem.

O escalonamento de veículos com múltiplas garagens (MDVSP, do inglês *Multi-Depot Vehicle Scheduling Problem*) consiste na atribuição de um conjunto de viagens escalonadas a um conjunto de veículos (homogêneo ou não) que são oriundos de duas ou mais garagens, com o objetivo de minimização de custos. O MDVSP também é a base de solução de vários problemas correlatos, tais como: (i) o problema de escalonamento de veículos em tempo real (Li *et al.*, 2009; Yang *et al.*, 2004; Powell and Carvalho, 1998;

Chen *et al.*, 2011); (ii) *disruption management* (Huisman and Wagelmans, 2006; Sato and Fukumura, 2012; Jozefowicz *et al.*, 2013) e (iii) soluções integradas de problemas tais como veículos e tripulação (Freling *et al.*, 2001; Haase *et al.*, 2001; Huisman *et al.*, 2005; Goel, 2009; Steinzen *et al.*, 2010).

Destá forma melhorar a solução deste problema pode ser considerada uma importante tarefa que permitirá resolver instâncias maiores e condizentes com os problemas do mundo real, bem como permitir a solução de novos problemas correlatos. Uma importante variação do MDVSP é o escalonamento de veículos com múltiplas garagens e frota heterogênea (de sigla em inglês, MDVTSP) . O MDVTSP pode ser descrito da mesma forma que o MDVSP, porém considerando diferentes tipos de veículos com capacidades e características distintas.

Bunte and Kliwer (2009) mostram que diversos tipos de formulações matemáticas foram propostas para o MDVSP: (i) modelos *single-commodity* (Carpaneto *et al.*, 1989; Mesquita and Paixão, 1992), (ii) modelos *multi-commodity* (Forbes *et al.*, 1994; Löbel, 1998; Haghani *et al.*, 2003; Gintner *et al.*, 2005; Kliwer *et al.*, 2006) e (iii) modelos de partição de conjuntos (Bianco *et al.*, 1994; Ribeiro and Soumis, 1994; Hadjar *et al.*, 2006). Os métodos utilizados propostos incluem algoritmos exatos ou *matheuristics* (Carpaneto *et al.*, 1989; Forbes *et al.*, 1994; Ribeiro and Soumis, 1994; Löbel, 1998; Hadjar *et al.*, 2006) e o uso de heurísticas (Ball *et al.*, 1983; Bodin *et al.*, 1983; Bianco *et al.*, 1994; Kliwer *et al.*, 2006; Rohde, 2008; Pepin *et al.*, 2009). Em relação à modelagem de rede, duas alternativas foram propostas: rede de conexão (Carpaneto *et al.*, 1989; Forbes *et al.*, 1994; Löbel, 1998; Ribeiro and Soumis, 1994; Pepin *et al.*, 2009) e a rede tempo-espaço (Kliwer *et al.*, 2002; Gintner *et al.*, 2005; Kliwer *et al.*, 2006; Hadjar *et al.*, 2006).

Como o problema é NP-Difícil (Bertossi *et al.*, 1987), várias instâncias não foram resolvidas na otimalidade, principalmente para instâncias geradas aleatoriamente. As maiores instâncias não-ótimas obtidas computacionalmente são de 1500 viagens e 8 garagens para o MDVSP (Pepin *et al.*, 2009) e para o MDVTSP de 1862 viagens, 5 tipos de veículos e 3 garagens levando 5 horas para encontrar uma solução heurística (van den Heuvel e van Kooten, 2008).

1.1 OBJETIVO GERAL

Desenvolver um método de solução, baseado na integração da rede tempo-espaço e da geração de colunas para a solução de grandes instâncias não estruturadas do MDVSP e do MDVTSP, envolvendo dezenas de garagens e milhares de viagens.

1.2 OBJETIVOS ESPECÍFICOS

1. Desenvolver um método de integração da rede tempo-espaço e o método de geração de colunas para a resolução do MDVSP.
2. Desenvolver um procedimento baseado na geração de colunas para a resolução do escalonamento de veículos com múltiplas garagens e frota heterogênea.
3. Desenvolver um gerador de instâncias compatível com a rede tempo-espaço para a realização de experimentos computacionais.
4. Desenvolver um construtor e redutor da rede tempo-espaço a partir de uma tabela de horários dada e as localizações dos terminais e garagens.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Essa dissertação se organiza da seguinte forma: No capítulo 2 será feita uma revisão da literatura existente sobre os problemas abordados neste trabalho, bem como sobre a modelagem de rede tempo-espaço e do método de geração de colunas. No capítulo 3 é descrito cada item do método científico utilizado para a resolução dos problemas propostos. O capítulo 4 é destinado a explicação em detalhes do método de solução do escalonamento de veículos com múltiplas garagens, que de mesma forma, foi destinado o capítulo 5 para a descrição do método de solução de sua variante com frota heterogênea. O capítulo 6 apresenta os experimentos computacionais realizados. O capítulo 7 apresenta resultados do método de solução apresentado no capítulo 4 para o MDVSP e o capítulo 8 faz o mesmo em relação ao MDVTSP. Por fim, são apresentadas algumas conclusões no capítulo 9.

2 REVISÃO DA LITERATURA

O objetivo deste capítulo é apresentar uma revisão da literatura sobre o MDVSP e o MDVTSP, analisando de forma crítica os modelos e métodos de solução desenvolvidos para resolvê-los.

Considerando que esta área apresenta muitos conceitos, faz-se necessário apresentar em uma seção os que serão utilizados no desenvolvimento deste trabalho, quais sejam:

2.1 DEFINIÇÕES

Cobertura (*Covering*): Atendimento à demanda por parte de um recurso.

Garagem (*Depot*): Facilidade que acolhe a frota de veículos de uma determinada empresa.

Terminais/Estações (*Relief point/Stations*): Local onde os veículos podem iniciar ou concluir uma viagem.

Tabela de Horários (*Timetable*): Estrutura de dados que contém as viagens a serem realizadas por um sistema de transporte público.

Bloco (*Vehicle Block*): É a sequência de viagens iniciadas e terminadas em uma garagem. Constitui, portanto, um conjunto de tarefas.

Viagem (*Trip*): Deslocamento de um veículo, com origem e destino determinados, e com horários estabelecidos para início e fim.

Viagem vazia ou viagem morta (*Deadhead*): Viagem realizada por um veículo entre dois terminais, em que não é efetuada a condução de passageiros.

Intervalo entre viagens (*Headway*): A distância no tempo ou no espaço que separa os dois veículos que viajam na mesma rota.

Estagnação: No contexto do método de geração de colunas, estagnação representa a situação onde o método encontra-se em um valor (que pode representar um mínimo local) por diversas iterações e a geração de novas colunas não consegue reduzir o custo.

Instância não Estruturada: Instância obtida através de um processo de geração aleatório de viagens, em que as viagens são independentes entre si.

Instância Estruturada: Instância, geralmente, originária de tabelas de horários vigentes, e, portanto, já foram feitas diversas adequações para compatibilizar viagens e economizar deslocamentos vazios de veículos.

2.2 ESCALONAMENTO DE VEÍCULOS COM MÚLTIPLAS GARAGENS

Para a apresentação da revisão literária, esse capítulo discute os seguintes pontos: Escalonamento de Veículos com Múltiplas Garagens (MDVSP), Escalonamento de Veículos com Múltiplas Garagens e Frota Heterogênea (MDVTSP), modelagem da rede tempo-espaço e o método de geração de colunas.

2.2.1 Formulação do MDVSP

O MDVSP pode ser definido através de uma rede de conexão, como se segue, seja $Trips$ um conjunto de viagens programadas e uma frota de veículos alojados em um conjunto K de garagens. Deve-se achar o custo mínimo de cobrir todas as viagens, na qual cada viagem seja atendida exatamente uma vez por um veículo, garantindo que o número v_k de veículos disponíveis em cada garagem $k \in K$ não seja excedido. Cada viagem $i \in Trips$ é definida por um local de início s_i e um local de destino e_i , um tempo de início a_i e um tempo de fim b_i . Um veículo deve iniciar e terminar na mesma garagem

compondo um bloco.

Desta forma, o MDVSP pode ser matematicamente formulado baseado em Pepin *et al.* (2009), que por sua vez se baseou em Bodin *et al.* (1983) e em Ribeiro e Soumis (1994). Define-se uma rede de conexão $G^k = (N^k, A^k)$ correspondente a uma garagem k , que é um grafo direcionado acíclico com vértices N e arestas A . Define-se um par de nós $o(k)$ e $d(k)$ representando o início e o fim de um bloco de um veículo associado a uma garagem k , respectivamente.

Considere c_{ij} como o custo do veículo ao utilizar o arco $(i, j) \in A^k$. Definindo a variável de decisão binária x_{ij}^k que assume valor igual a um se um veículo da garagem k atende a viagem j após realizar a viagem i , e zero caso contrário, o MDVSP pode ser formulado como segue:

$$\min \sum_{k \in K} \sum_{(i,j) \in A^k} c_{ij} x_{ij}^k \quad (2.1)$$

s.t.

$$\sum_{k \in K} \sum_{j:(i,j) \in A^k} x_{ij}^k = 1 \quad \forall i \in Trips \quad (2.2)$$

$$\sum_{j:(o(k),j) \in A^k} x_{ij}^k \leq v_k \quad \forall k \in K, \forall i \in Trips \quad (2.3)$$

$$\sum_{j:(j,i) \in A^k} x_{ji}^k - \sum_{j:(i,j) \in A^k} x_{ij}^k = 0 \quad \forall i \in (N^k - \{o(k), d(k)\}), \forall k \in K \quad (2.4)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in A^k, k \in K \quad (2.5)$$

A função objetivo (2.1) minimiza os custos totais. A restrição (2.2) garante que cada viagem é executada exatamente uma vez e por um veículo. A restrição (2.3) limita o número de veículos que podem ser utilizados a partir de cada garagem, enquanto que a restrição (2.4) é a restrição de conservação de fluxo. Finalmente, o requisito de integralidade das variáveis é fornecido por (2.5).

Uma rede de conexão, como descrita em Carpaneto *et al.* (1989) e aprimorada em Mesquita and Paixão (1992), é uma rede onde viagens e garagens são representadas por nós e as conexões possíveis entre as viagens são dadas pelos arcos. Os arcos na rede de conexão são compostos pelos arcos de espera, arcos *deadheading* e arcos de entrada e

saída da garagem. A Figura 2.1 apresenta um exemplo de rede de conexão com 4 viagens e 2 garagens.

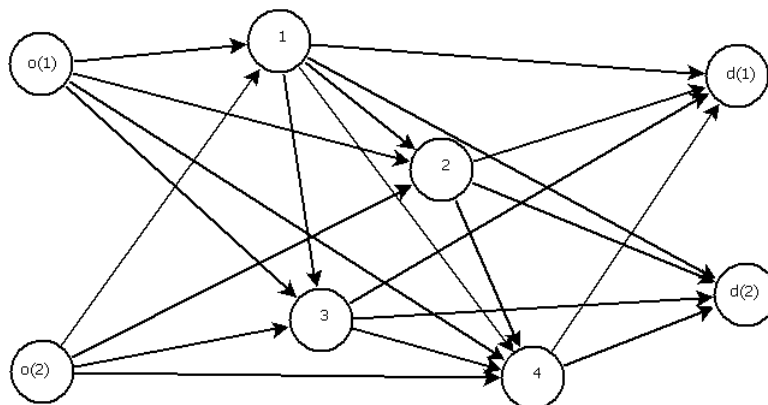


Figura 2.1: Exemplo de Rede de Conexão.

2.2.2 Revisitando o MDVSP

A Tabela 2.1 apresenta, de forma sucinta, um panorama da evolução na modelagem do MDVSP no que tange às contribuições de cada um dos trabalhos sejam essas através de proposições ou melhorias de métodos, maiores instâncias resolvidas ou resoluções ótimas ou não. A Tabela 2.2 se restringe a contribuições científicas que significativamente alteraram a representação ou a formulação do problema, uma vez que sendo o MDVSP um problema clássico, muitas são as propostas que tentaram solucioná-lo ou evoluir o estado-da-arte do problema.

Ano	Autor(es)	Rede	Garagens	Viagens	Método		Contribuição
1987	Bertossi <i>et al.</i> (1987)	Conexão	3	50	Relaxamento Lagrangeano	La-	Prova que o problema é NP-Difícil
1989	Carpaneto <i>et al.</i> (1989)	Conexão	3	100	<i>branch-and-bound</i>		Proposta de Formulação para o Problema Gerador de instâncias não estruturadas. Primeira solução ótima proposta
1992	Mesquita e Paixão (1992)	Conexão	3	200	Relaxamento Lagrangeano	La-	Nova formulação onde agrega-se os nós das garagens em um.
1994	Ribeiro e Soumis (1994)	Conexão	6	300	Geração de Colunas		Propõe o método de geração de colunas para o MDVSP
1994	Forbes <i>et al.</i> (1994)	Conexão	3	600	<i>branch-and-bound</i>		Aumenta o tamanho da maior instância ótima conhecida
1998	Löbel (1998)	Conexão	44	25000	Relaxamento Lagrangeano e Geração de Colunas	La-	Soluciona a maior instância estruturada vista na literatura, porém de forma não ótima
2002	Kliwer <i>et al.</i> (2002)	TSN	3	3776	Modelagem TSN e reduções na Rede		Propõe a rede tempo-espaço
2008	Pepin <i>et al.</i> (2009)	Conexão	8	1500	Diversas Heurísticas		Teste de diversas heurísticas e novo limite do tamanho de instâncias não estruturadas
2013	Lopes (2013)	Conexão	8	1500	Redução nos Espaços de Estados e Geração de Colunas		Propõe reduções no espaço de estados, apresenta uma implementação melhorada do método de geração de colunas

Tabela 2.1: Tabela de Revisão Literária para o MDVSP

Os artigos de Bertossi *et al.* (1987) e Carpaneto *et al.* (1989) são considerados os marcos iniciais do MDVSP. Bertossi *et al.* (1987) provam que o MDVSP é NP-Difícil e propõem uma solução por relaxamento lagrangeano da restrição 2.2. Carpaneto *et al.* (1989) apresentam uma formulação de transporte para a modelagem *single-commodity* com uma restrição de sub-ciclos, onde são representados as viagens e veículos como nós.

Carpaneto *et al.* (1989) são os primeiros a apresentar uma solução ótima para o MDVSP. Outra contribuição importante de Carpaneto *et al.* (1989) foi a definição de uma rotina para geração de instâncias aleatórias.

Mesquita e Paixão (1992) propõem outra abordagem *single-commodity*, onde uma estrutura de rede mais abrangente é usada. Os nós dos veículos, conforme descrito acima são agregados e combinados para um nó por garagem. Os autores resolvem pelo relaxamento lagrangeano da restrição 2.3.

Forbes *et al.* (1994) propõem outra abordagem (*multi-commodity*) e utilizam um relaxamento lagrangeano para executar um algoritmo dual simplex para se obter uma solução do problema linear. Forbes *et al.* (1994) também observam que a solução potencialmente fracionária é na maioria dos casos inteira, ou quase inteira, para os casos reais. Devido a este fato, a solução inteira foi obtida por um algoritmo *branch-and-bound* padrão. Forbes *et al.* (1994) solucionam o problema de forma ótima com 600 viagens e 3 garagens.

Ribeiro e Soumis (1994) propõem um método de geração de colunas para uma formulação de partição de conjuntos. Essa formulação pode ser obtida aplicando uma decomposição de Dantzig e Wolfe (1960) na modelagem *multi-commodity* como demonstram Hadjar *et al.* (2006). O método de geração de colunas fará parte da solução proposta nesse projeto e portanto será melhor descrito nas seções 2.4 e 3.3.

Löbel (1998) soluciona grandes instâncias oriundas de três grandes companhias de transporte urbano da Alemanha, através de um técnica de *branch-and-cut* com geração de colunas. Löbel (1998) utiliza a formulação proposta por Carpaneto *et al.* (1989). Ele propõe uma técnica denominada de *pricing* lagrangeano baseada em relaxações lagrangeanas do modelo de fluxo *multicommodity*. O *pricing* lagrangeano fornece os *lower bounds* e através de um conjunto de heurísticas (ND - atribui a cada viagem a garagem mais perto;

enquanto SCR - efetua o escalonamento antes, através de um fluxo de custo mínimo, e depois vincula cada viagem com uma garagem), o autor obtém os *upper bounds*. Esses dados são utilizados para inicializar a geração de colunas. Silva (2010) também utiliza essas heurísticas entretanto agregando restrições reais ao problema, tais como que o veículo deve permanecer, no mínimo, trinta minutos na garagem e se ele tiver um período de espera maior que duas horas em um terminal, ele deve obrigatoriamente retornar à garagem. Pode-se verificar que em Silva (2010) há um cuidado extra em relação ao veículo, ou seja cada veículo é um elemento diferenciado na modelagem. Silva (2010) conclui que a melhor solução é obtida quando é efetuado primeiramente o escalonamento seguido pelos agrupamentos.

Löbel (1998) resolve o RMP (do inglês, *Restricted Master Problem*), eliminando as colunas pelo critério de custo reduzido e gerando colunas pelo mesmo critério. Efetua, então, uma verificação se o problema global é ótimo ou próximo do ótimo. Em caso afirmativo efetua um *branch-and-cut* com o problema e novos *lower* e *upper bounds* são obtidos, caso contrário o procedimento é repetido. Suas maiores instâncias resolvidas na otimalidade são com 12 garagens e 8563 viagens e com 19 garagens e 5457 viagens, embora apresente uma solução para 25000 viagens com 49 garagens, mas esta não é ótima. Löbel (1998) não apresenta o *gap* para a solução inteira na maioria dos seus casos, bem como o valor da solução ótima é na maioria das vezes obtida através de um arredondamento do valor ótimo da relaxação linear (Hadjar *et al.*, 2006). Suas maiores instâncias além de serem estruturadas (diferente das abordadas no nosso trabalho) utilizaram até 16 horas de processamento computacional em uma máquina dedicada.

Todos as propostas de solução para o MDVSP até 2002 utilizavam redes de conexão. Kliwer *et al.* (2002), baseados em um conceito do escalonamento aéreo (*timeline*), propuseram a rede tempo-espaco para o MDVSP. Como essa modelagem fará parte dessa proposta, será melhor descrita na seção 2.3.

Hadjar *et al.* (2006) aprimoram Löbel (1998) propondo novas desigualdades válidas para a formulação do problema de partição de conjuntos. Hadjar *et al.* (2006) propõem um algoritmo de *branch-and-bound* para resolver o MDVSP, que combina geração de

colunas, ajuste na variável e planos de corte. Hadjar *et al.* (2006) provam que as desigualdades apresentadas representam, sob certas condições, as faces de um polítopo subjacente. Hadjar *et al.* (2006) afirmam que o máximo de viagens que se pode resolver para instâncias não estruturadas é cerca de 800 viagens. Contudo, para instâncias estruturadas, já foi possível resolver problemas com até 7000 viagens, conforme mostram os resultados atingidos por Löbel (1998) e Klierer *et al.* (2006).

Pepin *et al.* (2009) comparam o desempenho de cinco diferentes abordagens heurísticas para esse problema, entre elas a geração de colunas. Resultados computacionais em instâncias geradas aleatoriamente mostraram que a geração de colunas tem o melhor desempenho computacional. Pepin *et al.* (2009) solucionam instâncias de até 1500 viagens com 8 garagens.

Lopes (2013) optou por tratar o problema através de uma abordagem baseada na redução do espaço de estados e na utilização da técnica de geração de colunas truncada. Dois procedimentos de redução do espaço de estados foram utilizados a partir do trabalho de Rohde (2008). De acordo com Rohde (2008) é possível reduzir em até 98% o número de variáveis nesses problemas sem comprometer uma solução satisfatória ou ótima. Lopes (2013) solucionou o problema para uma rede de conexão.

Pela análise da Tabela 2.1, pode-se notar que a área encontra-se carente de métodos de solução que permitam a solução de grandes instâncias (com dezenas de garagens e milhares de viagens) em um tempo reduzido. Embora Löbel (1998) tenha resolvido instâncias com 25000 viagens e 49 garagens, o algoritmo demora cerca de 16 horas para encontrar uma solução de boa qualidade. Esse desempenho é incompatível com as exigências dos problemas correlatos ao MDVSP, o MDVSP deve ser resolvido muito rápido em situações de tempo-real, ou resolvido várias vezes como um subproblema do escalonamento de tripulação (*Crew Scheduling*) (Huisman *et al.*, 2005). Esta dissertação tem como principal contribuição integrar a rede TSN com o método de geração de colunas. Acredita-se que a integração desta representação e este conhecido método de solução de problemas de programação linear inteira possibilite o desenvolvimento de um método eficiente para grandes instâncias, envolvendo dezenas de garagens e milhares de viagens.

2.2.3 Formulando o MDVTSP

Escalonamento de veículos com frota heterogênea não tem recebido a mesma atenção na literatura que o problema com frota homogênea. Isto pode ser considerando uma surpresa, pois a maioria dos problemas de transporte urbano ou aéreo envolvem vários tipos de veículos (por exemplo, o transporte urbano de passageiros em São Paulo possui mais de 8 tipos diferentes de veículos). A quantidade de artigos publicados que trabalham diretamente com o MDVTSP é inócua. Muito disso se deve à grande complexidade de resolver esse problema.

Escalonamento de veículos com frota heterogênea compreende o uso de diversos veículos, com atributos, significativamente, diferentes para o atendimento de um conjunto específico de viagens que, neste trabalho, são originadas de uma tabela de horários obtidas a partir do gerador de instâncias proposto neste trabalho.

O MDVTSP pode ser definido como se segue, seja $Trips$ um conjunto de viagens programadas e uma frota de veículos F distribuídos em um conjunto K de garagens. O objetivo consiste em encontrar o custo mínimo de cobrir todas as viagens, na qual cada viagem seja atendida exatamente uma vez por um veículo, garantindo que o número v_{kf} de veículos disponíveis em cada garagem $k \in K$ do tipo $f \in F$ não seja excedido. Cada viagem $i \in Trips$ é definida por um local de início s_i e um local de destino e_i , um tempo de início a_i e um tempo de fim b_i . Um veículo deve iniciar e terminar na mesma garagem compondo um bloco.

Desta forma, o MDVTSP pode ser matematicamente formulado como segue. Definindo-se a rede $G^k = (N^k, A^k)$ correspondente a uma garagem k , que é um grafo direcionado acíclico com vértices N e arestas A . Define-se um par de nós $o(k)$ e $d(k)$ representando o início e o fim de um bloco de um veículo associado a uma garagem k , respectivamente. Considere c_{ij}^f como o custo do veículo ao utilizar o arco $(i, j) \in A^k$ utilizando o veículo $f \in F$, D_i a demanda da viagem i e q_f a capacidade do veículo f . Definindo a variável de decisão binária x_{ij}^{kf} que assume valor igual a um se um veículo da garagem k do tipo f atende a viagem j após realizar a viagem i , e zero caso contrário, o MDVTSP pode ser formulado como segue:

$$\min \sum_{k \in K} \sum_{f \in F} \sum_{(i,j) \in A^k} c_{ij}^f x_{ij}^{kf} \quad (2.6)$$

s.t.

$$\sum_{k \in K} \sum_{f \in F} \sum_{j \in (i,j) \in A^k} q_f x_{ij}^{kf} \geq D_i \quad \forall i \in Trips \quad (2.7)$$

$$\sum_{j: (o(k),j) \in A^k} x_{ij}^{kf} \leq v_f^k \quad \forall k \in K, f \in F \quad (2.8)$$

$$\sum_{k \in K} \sum_{f \in F} \sum_{j \in (i,j) \in A^k} x_{ij}^{kf} = 1 \quad \forall i \in Trips \quad (2.9)$$

$$\sum_{j: (j,i) \in A^k} x_{ji}^{kf} - \sum_{j: (i,j) \in A^k} x_{ij}^{kf} = 0 \quad \forall i \in (N^k - \{o(k), d(k)\}), \forall k \in K, f \in F \quad (2.10)$$

$$x_{ij}^{kf} \in \{0, 1\} \quad \forall (i,j) \in A^k, k \in K, f \in F. \quad (2.11)$$

A função objetivo (2.6) minimiza os custos totais. A restrição (2.7) garante que toda demanda de cada viagem seja atendida. A restrição (2.8) limita o número de veículos que podem ser utilizados a partir de cada garagem, enquanto que a restrição (2.9) garante que cada viagem é executada exatamente uma vez e por um veículo. A restrição (2.10) é a restrição de conservação de fluxo. Finalmente, o requisito de integralidade das variáveis é fornecido por (2.11).

2.2.4 Conhecendo um pouco o MDVTSP

A maioria dos problemas propostos na literatura sobre o escalonamento de veículos no transporte público considera frota homogênea ao invés de diferentes tipos de veículos (Ceder, 2011b); entretanto, na prática, mais de um tipo é utilizado. Um dos motivos para a falta de estudos sobre o tema é que diferentes tipos de veículos introduzem complexidade ao problema de escalonamento de veículos. Tal situação representa uma simplificação de contexto, que não se sustenta em situações práticas, principalmente no contexto do transporte público.

A Tabela 2.2 apresenta, de forma sucinta, um panorama da evolução na modelagem do MDVTSP no que tange às contribuições de cada um dos trabalhos sejam essas através de proposições ou melhorias de métodos, maiores instâncias resolvidas ou resoluções ótimas ou não.

Ano	Autor(es)	Método	Contribuição
2005	Gintner <i>et al.</i> (2005)	Heurística de duas fases que fixa cadeias estáveis	Apresenta uma heurística de solução que utiliza a rede tempo-espço <i>fix-and-optimize</i>
2006	Kliwer <i>et al.</i> (2006)	vários SDVSPs com rotações	Apresenta uma heurística de solução que utiliza a rede tempo-espço
2006	Hadjar <i>et al.</i> (2006)	Branch-and-Cut com Geração de Colunas	Aprimora o tamanho de uma solução ótima conhecida
2008	van den Heuvel and van Kooten (2008)	Busca Local	Propõe reduções e melhorias a modelagem da rede tempo-espço
2008	Laurent and Hao (2008)	GRASP	Integram ao escalonamento de veículos com frota heterogênea em uma única garagem, o problema de escalonamento da tripulação
2009	Oughalime <i>et al.</i> (2009)	programação inteira e goal programming	Esforços teóricos para solução integrada do escalonamento de veículos e tripulação.
2011	Ramos <i>et al.</i> (2011)	Colônia de Formigas	Apresenta uma modelagem em forma do problema do caixeiro viajante assimétrico para o MDVTSP
2011	Ceder (2011b)	Teoria de Déficit de Função	Propõe uma integração do Escalonamento de veículos de frota heterogênea com a otimização da tabela de horários

Tabela 2.2: Tabela de Revisão Literária para o MDVTSP

Com objetivo de solucionar o escalonamento de veículos com múltiplas garagens e frota heterogênea, Gintner *et al.* (2005) apresentam uma heurística de duas fases, que fixa algumas conexões *a priori*. A idéia básica da heurística é primeiro resolver um número simplificado de modelos, como um SDVSP (do inglês, *Single Depot Vehicle Scheduling Problem*) para cada garagem, e então procurar por cadeias de viagens comuns em cada uma das soluções. Se a mesma sequência de viagens é incluída em cada solução, os autores classificam esta como uma cadeia estável e assumem que pode ocorrer na solução ótima global. As cadeias estáveis atuam como viagens na otimização exata do modelo, reduzindo-o significativamente. Esta técnica, que produz soluções muito próximas da otimalidade, é chamada de "*fix-and-optimize*", pois primeiro fixa algumas variáveis e depois utiliza a técnica de otimização.

Kliwer *et al.* (2006) afirma ter encontrado uma solução para o MDVTSP, porém simplificações significativas indicam relaxamentos que descaracterizam o problema como um MDVTSP. Em seu método de solução foi designada uma garagem específica para cada tipo de veículo, relaxando a questão da frota heterogênea a uma situação irreal. Essa simplificação permitiu resolver um SDVSP para cada tipo de frota. A solução global é

encontrada através de uma heurística que combina as soluções dos SDVSPs. Com as simplificações apresentadas os autores resolveram instâncias estruturadas com até 7068 viagens e 5 garagens, para tal gastaram 3 horas de processamento computacional.

Van den Heuvel e van Kooten (2008) ampliam a formulação de Gintner *et al.* (2005), adicionando o tipo de veículo na formulação, bem como a demanda de passageiros e possibilitando que mais de um veículo realize uma mesma viagem; entretanto os autores consideram somente uma garagem. Assim, são propostas duas novas formulações: na primeira, mais de um tipo de veículo para atender a uma viagem, mas que se torna muito difícil de resolver; e a segunda, que visa minimizar em partes a complexidade da primeira, permitindo que vários veículos do mesmo tipo atendam a mesma viagem, sendo que os veículos são distribuídos uniformemente no tempo. Se, por exemplo, há uma viagem a cada hora, e serão necessários dois veículos de um determinado tipo, um veículo sairia do terminal a cada meia-hora. Isso é obtido através da inclusão de viagens de serviço adicionais na rede tempo-espaco. Esse problema, ao flexibilizar a frota, aumenta significativamente a complexidade de resolução computacional se comparado ao VSP para um único tipo de veículo.

Laurent and Hao (2008) integram ao escalonamento de veículos com frota heterogênea em uma única garagem, o problema de escalonamento da tripulação. Para resolver o problema, eles aplicam uma heurística GRASP (do inglês, *Greedy Randomized Adaptive Search Procedure*). A construção da solução inicial é baseada na técnica de programação por restrições e a fase de busca local usa um mecanismo de exploração da vizinhança. Os experimentos computacionais foram conduzidos com sete diferentes instâncias, advindas de dados reais, considerando dois tipos de veículos.

Em trabalho posterior, Laurent and Hao (2009) consideram a frota heterogênea no contexto de um MDVSP e o formulam como um problema de coloração de grafos. Eles testam sete instâncias reais, com oito categorias de veículos. Uma Busca Tabu iterativa é desenvolvida com o objetivo de minimizar o número total de veículos, indicando bons resultados em termos de qualidade de solução e baixo tempo computacional.

Oughalime *et al.* (2009) apresentam esforços teóricos para efetuar a integração do

escalonamento de veículos com frota heterogênea e o escalonamento de tripulação. Os autores propuseram uma modelagem sequencial que utiliza programação inteira para a solução do problema de escalonamento de veículos e *goal programming* para o escalonamento de tripulação. Embora o artigo trate de um exemplo real, nele só é apresentada a definição do problema e a modelagem proposta. Não há uma implementação, nem uma experimentação que valide o modelo.

Ramos *et al.* (2011) modela o MDVTSP como um problema do caixeiro viajante assimétrico. Para solucionar eles propõem o uso de um método de colônia de formigas. Ramos *et al.* (2011) solucionam instâncias reais de uma empresa portuguesa. Essa abordagem não apresenta-se satisfatória visto que o problema do caixeiro viajante é NP-Completo.

Ceder (2011b) integra a frota heterogênea e a tabela de horários. A partir da Teoria de Déficit da Função (DF), diferentes tipos de veículos, já relacionados com viagens específicas, são escalonados. Conforme definição de Ceder (2011b), o DF representa o número deficitário de veículos em um terminal particular em um sistema de transporte multi-terminal. A partir dessa premissa, o problema é formulado como um fluxo de rede de custo mínimo e objetiva designar a quantidade mínima de veículos a determinada tabela de horários. Uma heurística é desenvolvida para resolver o problema e são apresentados três exemplos fictícios para explicar seus princípios. O procedimento heurístico permite a inclusão de mudanças nos horários de partida das viagens, dentro de um limite de tolerância (adiantar ou atrasar), seguindo alguma regra contida na DF.

Ceder (2011a) trabalha com duas fases iniciais do planejamento do transporte público (planejamento da tabela de horários e escalonamento de veículos). Ceder (2011a) propõe a construção de tabelas de horários alternativas conforme a demanda de passageiros e define os tipos de veículos que podem atendê-las, a partir de um procedimento heurístico. Tendo como ponto de partida as tabelas de horários previamente definidas, o escalonamento dos veículos é realizado heurísticamente, incorporando componentes da teoria de déficit de função (*deficit-function theory*), já aplicada por Ceder (2011b), que tem como base o número de veículos necessários em um terminal. De modo bem simplificado, a

heurística consiste em ordenar os veículos em ordem decrescente de custo, para após resolver o problema de escalonamento de veículos de diferentes tipos usando a teoria de déficit de função. A introdução de viagens expressas ao escalonamento dos veículos também é definida através da heurística, possibilitando a redução do tamanho da frota. Esse procedimento é testado com dados reais, originários de Auckland, Nova Zelândia.

Como pode-se verificar pela revisão da literatura acima detalhada, o estado-da-arte em modelagem e solução do MDVTSP é incipiente. O desenvolvimento de uma abordagem que permita uma solução de qualidade é um tema de grande relevância na área.

2.3 REDE TEMPO-ESPAÇO

2.3.1 Formulação da Rede Tempo-Espaço

A rede tempo-espaço (TSN, do inglês Time-Space Network) é a base para a formulação matemática apresentada nesta dissertação. Em uma TSN, os nós representam um local específico no tempo e espaço, e cada arco corresponde a uma transição no tempo e, possivelmente, no espaço. A rede tempo-espaço foi primeiramente proposta para problemas de roteamento em escalonamento aéreo (Hane *et al.*, 1995), devido a sua facilidade na modelagem de possíveis conexões entre vôos.

A definição formal da TSN necessita certa nomenclatura. Seja S o conjunto de terminais; K o conjunto de garagens e T o tempo. Definindo N como o conjunto de vértices da rede caracterizados por uma tripla ordenada com os conjuntos S (espaço), T (tempo), e $Trips$ (viagens de serviço). Uma viagem $i \in Trips$ pode ser denotada como um par de nós i_s e i_e ou j_s e j_e , representando o ponto de início e de fim da viagem, respectivamente. Definindo também A como o conjunto de arestas que em uma rede tempo-espaço é subdividido em 6 diferentes conjuntos (cinco tipos nativos mais um artificial), assim definidos:

O conjunto A^s denota o conjunto de arcos que representam as viagens i a serem executadas:

$$A^s = \{(n_1, n_2) \in N \times N \mid n_1 = (s_1, t_1, i_s) \wedge n_2 = (s_2, t_2, i_e)\};$$

O conjunto A^{wait} denota o conjunto de arcos de espera, que representam as transições somente no tempo que simbolizam que o veículo se encontra parado em um terminal, sendo:

$$A^{wait} = \{(n_1, n_2) \in N \times N \mid n_1 = (s, t_1, i_e) \wedge n_2 = (s, t_2, j_s)\};$$

O conjunto A^{dh} denota o conjunto de arcos que representam uma transição no tempo e espaço sem passageiros, sendo:

$$A^{dh} = \{(n_1, n_2) \in N \times N \mid n_1 = (s_1, t_1, i_e) \wedge n_2 = (s_2, t_2, j_s)\};$$

O conjunto A^{pin} denota o conjunto de arcos que representam uma viagem da garagem para um terminal com o propósito de iniciar um bloco de viagens, sendo:

$$A^{pin} = \{(n_1, n_2) \in \{o(k)\} \times N \mid n_1 = (o(k), t_1, \emptyset) \wedge n_2 = (s_2, t_2, i_s) \wedge k \in K\}$$

O conjunto A^{pout} denota o conjunto de arcos que representam um deslocamento para uma garagem a partir de um terminal com o significado de finalizar um bloco de viagens, sendo:

$$A^{pout} = \{(n_1, n_2) \in N \times \{d(k)\} \mid n_1 = (s_1, t_1, i_e) \wedge n_2 = (d(k), t_2, \emptyset) \wedge k \in K\};$$

É possível definir, sem perda de generalização, O conjunto A^c , subconjunto de A , que denota o conjunto de arcos de circulação, formado por arcos que ligam cada veículo no ponto final de sua jornada ao ponto de início da mesma com propósito de permitir a minimização do número de veículos utilizados. Esse arco é artificial, e serve ao propósito de facilitar a resolução.

A figura 2.2 é um exemplo de uma TSN com 5 viagens e 3 terminais.

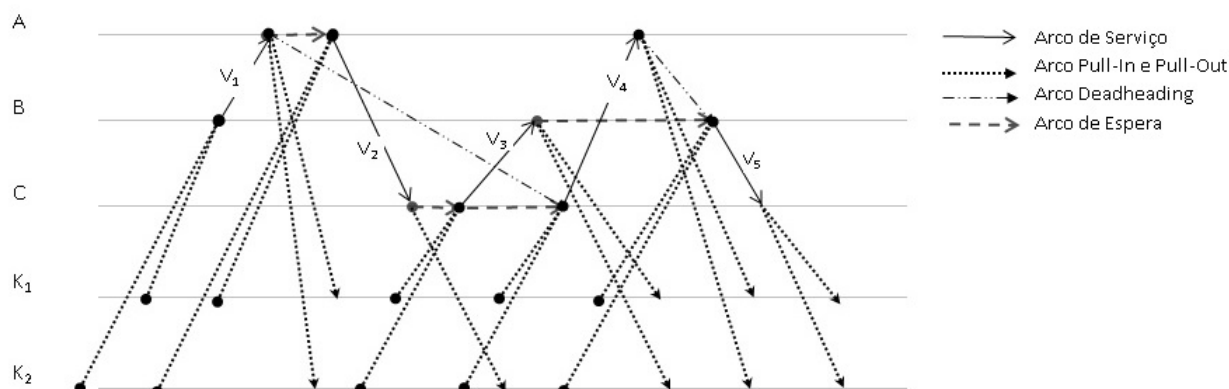


Figura 2.2: Exemplo de Rede Tempo-Espaço.

A principal vantagem da estrutura de TSN é a redução do número de variáveis e restrições, comparada com outras abordagens na literatura (Steinzen *et al.*, 2010). Se o problema contém m estações e n viagens, então o número de arcos *deadheading* em uma TSN é $O(mn)$ em oposição a $O(n^2)$ da rede de conexão, sendo $n \gg m$. Steinzen *et al.* (2010) mostram um comparativo entre o número de arcos vazios gerados em uma TSN e em uma rede à base de conexão. A TSN é especialmente relevante quando o número de estações envolvidas no problema é baixo quando comparado com o número de viagens.

2.4 MÉTODO DE GERAÇÃO DE COLUNAS

A geração de colunas é uma forma de resolver um problema de programação linear que adiciona colunas (correspondentes às variáveis) durante a fase de *pricing* do método simplex (Desaulniers *et al.*, 2005). Gerar uma coluna na formulação primal simplex de um problema de programação linear corresponde à adição de uma restrição, na formulação dual do problema; desta forma, é possível interpretar que a técnica de geração de colunas se assemelha a um método de planos de corte.

Geração de colunas se baseia no fato de que, no método simplex, o método de solução não necessita de acesso a todas as variáveis do problema simultaneamente. Na verdade, um método de solução pode começar a trabalhar com apenas a base e então usar os custos reduzidos para decidir que outras variáveis acessar, quando necessário.

A idéia geral do método introduzido por Ford e Fulkerson (1958) e aprimorado por

Dantzig e Wolfe (1960), consiste em resolver um conjunto de subproblemas que gerarão colunas para o problema mestre reduzido, obtidas através das variáveis duais. O algoritmo proposto será descrito na seção 4.3 para o MDVSP e na seção 5.4 para o MDVTSP.

O algoritmo de geração de colunas converge quando a solução ideal com base no conjunto atual de colunas não pode ser melhorada pela adição de mais colunas. Então, a solução ótima do problema reduzido é a solução ideal para o problema original. Neste trabalho será referenciado o problema reduzido como o problema mestre e para o problema da geração de um novo conjunto de colunas como subproblemas ou *pricing*.

Seja o seguinte problema de programação linear (P'), com um grande número de variáveis $|J|$:

$$(P'): \min \sum_{j \in J} c_j x_j \quad (2.12)$$

st

$$\sum_{j \in J} a_{ij} x_j = b_i, \quad \forall i \in N \quad (2.13)$$

$$x_j \geq 0, \quad \forall j \in J. \quad (2.14)$$

Caso aplique-se geração de colunas para resolver (P'), iniciaria-se com um conjunto inicial de colunas $J_0 \subseteq J$ de tal modo que estas colunas contêm uma solução viável para o problema reduzido. Resolve-se este problema e calcula-se os valores ótimos das variáveis duais. Estas variáveis duais são necessárias nos subproblemas para a seleção de novas colunas com custo reduzido negativo, que serão adicionadas ao conjunto J_0 . O custo reduzido de uma coluna é definida como $c_j - \sum_{i \in N} a_{ij} \mu_i$, onde μ_i é o valor ótimo da variável dual correspondente a cada restrição em (2.12). Se não existem colunas restantes com custos reduzidos negativos, pode ser facilmente visto que a solução ótima do problema reduzido é também a solução ótima de (P'). Observe que, se $\min_{j \in J} c_j - \sum_{i \in N} a_{ij} \mu_i > 0$, todas as colunas têm custos reduzidos positivos. Devido a esta observação trivial, não é necessário que todas as colunas sejam explicitamente conhecidas. O problema de *pricing* pode ser visto como um problema de otimização, por si só, e colunas podem ser geradas implicitamente, resolvendo este problema de otimização. Em aplicações de transporte, estes problemas são mais frequentemente problemas de caminho mais curto

(caminho mais curto simples ou caminho mais curto com restrições de recursos, onde o leitor deve observar que o primeiro pode ser resolvido em tempo polinomial, enquanto o segundo é, em geral, NP-Difícil). Se (algumas das) as variáveis de decisão precisam ser inteiras, algoritmos *branch-and-bound* são mais frequentemente utilizados para resolver o problema de programação inteira (mista) resultante. Geração de colunas podem ser incorporada na árvore do *branch-and-bound*, resolvendo os relaxamentos em cada nó com um algoritmo de geração de colunas. Esta abordagem é chamada de solução *branch-and-price*.

Geração de colunas vem sendo aplicada em diversas classes de problemas, Ford e Fulkerson (1958), por exemplo, sugeriram a técnica de geração de colunas no contexto de um problema de fluxo de rede *multi-commodity*. Gilmore e Gomory (1961), em seguida, demonstraram a sua eficácia em um problema de corte. Mais recentemente, problemas de cobertura (Alfandari *et al.*, 2013), roteamento e escalonamento de veículos (Jin *et al.*, 2013; Archetti *et al.*, 2013), problema de cortes (Furini e Malaguti, 2013), e outros problemas de programação inteira têm motivado mais pesquisas sobre geração de colunas.

No contexto do MDVSP, Ribeiro e Soumis (1994) foram os primeiros a aplicar o método de geração de colunas, apresentando uma reformulação do MDVSP para um problema de partição de conjuntos. Ribeiro e Soumis (1994) propõem a geração de colunas para o MDVSP descrito da seguinte forma: Seja Ω^k o conjunto de todos os blocos de veículos de uma garagem k . Define-se θ_p a variável de decisão que indica se o bloco de veículo p pertencerá a solução ou não, c_p como a soma de todos os custos dos arcos em θ_p e a_{ip} como uma constante binária que possui valor um se o *vehicle block* p inclui a viagem i . O MDVSP pode ser formulado como:

$$\min \sum_{k \in K} \sum_{p \in \Omega^k} c_p \theta_p \quad (2.15)$$

st

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip} \theta_p = 1, \quad \forall i \in Trips \quad (2.16)$$

$$\sum_{p \in \Omega} \theta_p \leq v_k, \quad \forall k \in K \quad (2.17)$$

$$\theta_p \in \{0, 1\}, \quad \forall p \in \Omega^k. \quad (2.18)$$

A função objetivo (2.15) visa minimizar os custos totais. A restrição de particionamento de conjuntos (2.16) impõe que cada tarefa seja realizada por exatamente um veículo, enquanto que a restrição (2.17) expressa a restrição de capacidade por garagem. Finalmente, o requisito de integralidade das variáveis é fornecido por (2.18).

Para cada iteração n , o subproblema da garagem $k \in K$ é o problema do caminho mais curto de $o(k)$ para $d(k)$ na rede de conexões G^k com a estrutura de custos modificada, sendo que a solução do subproblema é sempre um raio convexo (Ribeiro e Soumis, 1994), seguindo os valores da combinação linear dos raios extremos de cada iteração do algoritmo de Dantzig e Wolfe (1960). Considera-se também, as variáveis π_n^i , com $i \in T$ e β_n^k , com $k \in K$ como os valores das variáveis duais associadas com as restrições (2.7) e (2.8) do RMP na iteração n , o custo modificado do arco $(i, j) \in A^k$ é dado por $c_{ij} - \pi_n^i$, se $i \in T$ e $c_{ij} = \beta_n^k$ se $i = o(k)$ (Pepin *et al.*, 2009).

Após obter os custos alterados, o custo de um caminho p em G^k corresponde ao custo reduzido da variável θ_p , portanto, resolvendo este subproblema na iteração n , o escalonamento $p \in \Omega^k$ com o menor custo reduzido pode ser identificado.

3 MÉTODO

Este capítulo tem como objetivo apresentar a metodologia utilizada no desenvolvimento do presente estudo. Para o desenvolvimento desta dissertação, adotou-se a metodologia de Pesquisa Operacional baseada nas premissas de Arenales *et al.* (2011).

3.1 ETAPA 1: DEFINIÇÃO DO PROBLEMA

Essa dissertação trabalha com grandes instâncias dos problemas de escalonamento de veículos com múltiplas garagens com frota homogênea (MDVSP) e frota heterogênea (MDVTSP). A seguir, descrever-se-á em detalhes ambos problemas.

Nº	Horário de Partida	Horário de Chegada	Origem	Destino	Demanda
1	7:12	8:17	A	E	70
2	8:44	9:16	B	C	50
3	8:49	9:42	A	C	40
4	9:12	10:17	D	B	30
5	10:44	11:22	B	C	45
6	11:49	12:17	A	D	55
7	12:12	13:17	B	E	60
8	13:44	14:22	B	D	45
9	14:49	15:37	D	C	25
10	15:54	16:15	D	E	20

Tabela 3.1: Exemplo de Tabela de Horários

Para o MDVSP tem-se: Dada uma Tabela de horários representando um conjunto de viagens a serem executadas (vide Tabela 3.1, exceto pela coluna de demanda) e um conjunto de garagens com suas localizações e capacidades, busca-se oferecer ao tomador de decisão um escalonamento dos veículos de custo ótimo ou quase ótimo com o menor

número de veículos utilizados. Nesta dissertação acrescenta-se à descrição o fato de solucionar grandes instâncias não-estruturadas em um tempo razoável computacionalmente.

O escalonamento de veículos com múltiplas garagens e frota heterogênea se assemelha na definição ao MDVSP. Sua diferença está no fato de que as viagens podem ser alocadas a diferentes tipos de veículos.

A partir de uma tabela de horários representando um conjunto de viagens a serem executadas (vide Tabela 3.1) com suas demandas estimadas, um conjunto de garagens com suas localizações e capacidades, tem-se ainda a informação de quantos tipos de veículos, quais as suas localizações (a que garagem pertencem) e suas capacidades para então buscar oferecer ao tomador de decisão um escalonamento dos veículos de custo ótimo ou quase ótimo com o menor número de veículos utilizados.

3.2 ETAPA 2: MODELAGEM

Para atingir o propósito desta dissertação o problema apresentado na seção 3.1 foi modelado como uma rede tempo-espaço, conforme Kliwer *et al.* (2002). Cada viagem ocorre em uma rede de transporte determinada por um conjunto de nós e arestas. Cada nó representa uma posição no tempo e no espaço e está vinculado a uma viagem. Já cada arco representa um deslocamento nessa rede de transporte seja no tempo ou no espaço.

3.2.1 Modelo de Rede Tempo-Espaço

Com base na formulação da rede apresentada na seção 2.3.1, pode-se então modelar nosso problema em uma rede tempo-espaço, como segue:

1. As garagens $k \in K$ e cada terminal $s \in S$ serão representados por uma linha do tempo.
2. As viagens de serviços (A^s), as viagens sem passageiros entre terminais (A^{dh}), o intervalo em que um veículo encontra-se estacionado em um terminal (A^{wait}) e as viagens de início e fim de bloco de veículo serão representadas por arcos.
3. Cada evento (início ou fim de um deslocamento no tempo ou no espaço) será modelado via um nó na rede.

3.2.2 Formulação do MDVSP

A formulação apresentada por Pepin *et al.* (2009) para o MDVSP, usando a rede TSN, será utilizada, qual seja:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \quad (3.1)$$

s.t.

$$\sum_{k \in K} \sum_{j:(i,j) \in A^s} x_{ij}^k = 1, \quad \forall i \in A^s \quad (3.2)$$

$$\sum_{j:(o(k),j) \in A^{b\text{out}}} x_{o(k),j}^k \leq v_k, \quad \forall k \in K \quad (3.3)$$

$$\sum_{j:(j,i) \in A} x_{ji}^k - \sum_{j:(i,j) \in A} x_{ij}^k = 0, \quad \forall i \in V^k / \{o(d), f(d)\}, k \in K \quad (3.4)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in A, k \in K. \quad (3.5)$$

A variável de decisão x_{ij}^k denota a garagem que atende o arco do ponto i até o ponto j , sendo i e j nós na rede tempo-espço. A função objetivo (3.1) minimiza os custos totais. A restrição (3.2) garante que cada tarefa é executada exatamente uma vez por um veículo. A restrição (3.3) limita o número de veículos que podem ser utilizados a partir de cada depósito, enquanto que a restrição (3.4) é a restrição de conservação de fluxo. Finalmente, o requisito de integralidade das variáveis é fornecido por (3.5).

3.2.3 Formulação do MDVTSP

Baseado na formulação da seção 3.2.2, considerando que c_{ij}^f como o custo do veículo do tipo f ao utilizar o arco $(i, j) \in A$, q_f a capacidade do veículo f , d_j a demanda da viagem j e v_{kf} como a quantidade de veículos do tipo f na garagem k . Definindo a variável de decisão binária x_{ij}^{kf} que assume valor igual a um se um veículo da garagem k e do tipo f percorrer o arco $(i, j) \in A$, e zero caso contrário. foi proposta a seguinte formulação para o problema de escalonamento de veículos com frota heterogênea:

$$\min \sum_{f \in F} \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^{kf} x_{ij}^{kf} \quad (3.6)$$

s.t.

$$\sum_{f \in F} \sum_{k \in K} \sum_{j:(i,j) \in A^s} q_f x_{ij}^{kf} \geq D_j, \quad \forall j \in A^s \quad (3.7)$$

$$\sum_{f \in F} \sum_{k \in K} \sum_{j:(i,j) \in A^s} x_{ij}^{kf} = 1, \quad \forall i \in A^s \quad (3.8)$$

$$\sum_{j:(o(k),j) \in A^{pout}} x_{o(k),j}^{kf} \leq v_{kf}, \quad \forall k \in K, f \in F \quad (3.9)$$

$$\sum_{j:(j,i) \in A} x_{ji}^{kf} - \sum_{j:(i,j) \in A} x_{ij}^{kf} = 0, \quad \forall i \in N^k / \{o(k), d(k)\}, k \in K, f \in F \quad (3.10)$$

$$x_{ij}^{kf} \in 0, 1 \quad \forall (i, j) \in A, k \in K, f \in F. \quad (3.11)$$

A variável de decisão x_{ij}^{kf} denota o tipo de veículo e a garagem que atende o arco do ponto i até o ponto j , sendo i e j nós na rede tempo-espço. A função objetivo (3.6) minimiza os custos totais. A restrição (3.7) garante que toda a demanda de cada viagem vai ser atendida. A restrição (3.8) garante que cada viagem é executada exatamente uma vez por um veículo. A restrição (3.9) limita o número de veículos de cada tipo que podem ser utilizados a partir de cada garagem, enquanto que a restrição (3.10) é a restrição de conservação de fluxo. Finalmente, o requisito de integralidade das variáveis é fornecido pela restrição (3.11). Em relação à formulação do MDVSP (seção 3.2.2) a modificação mais significativa está na adição de uma restrição de atendimento da demanda (restrição 3.7).

3.3 ETAPA 3: MÉTODO DE SOLUÇÃO

Para a solução dos problemas formulados em 3.2.2 e em 3.2.3 foi utilizada a rede tempo-espço (descrita em 3.2) e o método de geração de colunas truncado modificado (descrito em 4.3 para o MDVSP e em 5.4 para o MDVTSP). Em uma visão holística, o algoritmo pode ser visto na figura 3.1. Obviamente a solução de cada um dos problemas exige especificidades, que serão detalhadas no capítulo 4 para o MDVSP e no capítulo 5 para o MDVTSP. O método proposto baseia-se na rede tempo-espço, pois, segundo Steinzen *et al.* (2010), a rede tempo-espço apresentou melhores resultados do que a rede

de conexão. A escolha pelo método de geração de colunas ocorreu por se tratar de um método otimizador bastante utilizado na literatura e que de acordo com Pepin *et al.* (2009) apresentou bons resultados para o problema de escalonamento de veículos com múltiplas garagens. Uma vez feita a escolha pelo método de geração de colunas, foi necessário construir um algoritmo modificado e um conversor da rede tempo-espço para a rede de conexão equivalente (esse procedimento está descrito na seção 4.2 do para o MDVSP e na seção 5.3 para o MDVTSP). A necessidade de construção de um conversor é decorrente da decisão de não desenvolver um método totalmente diferenciado de geração de colunas especialista para o uso com a rede tempo-espço. Esta decisão foi tomada primeiramente para tentar manter uma compatibilidade com os estudos anteriormente desenvolvidos, segundo, por simplicidade de implementação e terceiro, porque como efeito colateral desenvolveu-se um algoritmo que permitirá uma melhor análise da rede tempo-espço versus a rede conexão em trabalhos futuros, uma vez que agora é possível a partir de uma chegar na outra.

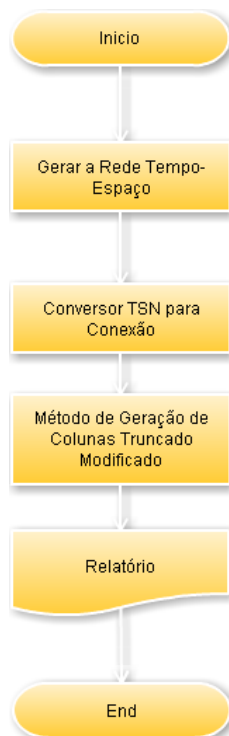


Figura 3.1: Fluxograma Simplificado do Método de Solução

3.4 ETAPA 4: VERIFICAÇÃO DOS MODELOS

Para a verificação do modelo foram geradas instâncias conforme descrito na seção 6.1. Isso ocorreu porque as instâncias não-estruturadas disponíveis na literatura tratavam somente de modelos de conexão (Carpaneto *et al.*, 1989). Embora outros autores já trabalhem com instâncias modeladas como rede tempo-espaço (Kliwer *et al.*, 2002; Löbel, 1998; Gintner *et al.*, 2005; Steinzen *et al.*, 2010), essas são instâncias estruturadas e privadas das empresas utilizadas nos estudos. Como objetiva-se solucionar instâncias não-estruturadas, foi necessário desenvolver um gerador direcionado a redes tempo-espaço, considerando não somente as distâncias e os tempos, mas a questão da demanda que é de extrema importância para a solução do MDVTSP.

3.5 ETAPA 5: EXPERIMENTAÇÃO

Experimentos computacionais foram desenvolvidos para analisar o impacto da calibragem dos parâmetros e capacidade computacional do método. Os experimentos foram feitos em uma máquina Intel® Core™ i7-3632QM com 2,2 GHz, 8 GB de memória rodando em uma plataforma Windows 8 64bits. Foi utilizado o solver Cplex 12.5 e a linguagem de desenvolvimento foi o C++. As instâncias aqui utilizadas foram geradas utilizando-se um gerador especialmente desenvolvido e estão disponíveis no site: sites.google.com/site/pablocguedes/disciplinas/instancias-mdvsp. Durante a experimentação verificou-se que o maior impacto computacional encontra-se na resolução dos problemas de caminho mais curto, que se trata dos subproblemas do método de geração de colunas. Portanto, foi feita uma breve análise de métodos computacionais e os resultados encontram-se na seção 4.3.

4 RESOLVENDO O MDVSP

Neste capítulo apresenta-se com detalhes como o escalonamento de veículos com múltiplas garagens foi resolvido. A seção 4.1 descreve como foram feitas a geração e as reduções da rede tempo-espço. Na seção 4.2 apresenta-se o conversor utilizado. Por fim, na seção 4.3 descreve-se em detalhes o método de geração de colunas aplicado a esse problema.

4.1 GERAÇÃO E REDUÇÃO DA REDE TEMPO-ESPAÇO

Com as definições da seção 3.2.1 é possível construir uma rede tempo-espço que representa computacionalmente uma instância do problema definido. Entretanto o número de arcos, ou seja, o número de variáveis em uma rede de médio porte torna proibitivo solucionar esse problema nos softwares especializados atuais. Portanto, baseado no trabalho de Kliwer *et al.* (2002) efetua-se reduções na rede. A redução dos arcos da rede tempo-espço é possível sobre os arcos expressos. A agregação desses arcos ocorre em dois estágios.

No Estágio 1, para cada viagem de chegada escalonada no terminal s' é determinada a primeira viagem compatível com a viagem i em cada estação s ($s' \neq s$, s' e s são terminais). Assim, são introduzidos os arcos vazios que correspondem à primeira partida (*first-match*). Percebe-se que todas as possíveis conexões anteriormente apresentadas permanecem viáveis. As demais viagens compatíveis com a viagem i , que tiveram seus arcos de espera agregados à primeira viagem, continuam compatíveis à mesma, devido aos arcos de espera traçados.

No Estágio 2, o número de arcos vazios é ainda mais reduzido. O conjunto de arcos da primeira partida é agrupado no último arco da primeira partida (*latest-first-match*). Os últimos arcos podem ser obtidos da seguinte forma: Seja S o conjunto de viagens de entrada i no terminal k' , tendo a mesma chegada (*first-match*) j no terminal k . Seja i_L a última viagem de entrada em S . Então, o último arco para cada elemento em S é igual a (i_L, k) . Ao manter todos os arcos correspondentes à última chegada da primeira partida, a rede é reduzida significativamente, e nenhuma conexão possível é perdida. A figura 4.1 mostra um exemplo de rede e sua correspondente após as reduções.

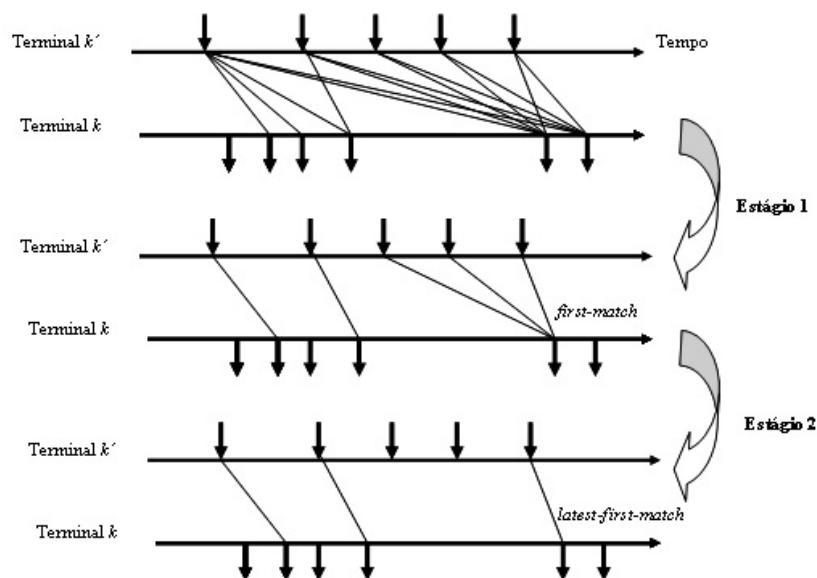


Figura 4.1: Estágios de Agregação dos Arcos Expressos. (Adaptada de Kliewer et. al., 2002)

A fim de reduzir ainda mais o tamanho da rede tempo-espacial, realizou-se um pré-processamento sobre os vértices que originam ou recebem os arcos *pull-in/out*. Se há vértices subsequentes na garagem apenas com arcos *pull-in*, ou seja, que partem da garagem em direção a um terminal, esses podem ser agrupados no primeiro vértice de saída (do primeiro arco *pull-in*), incluindo um arco de espera entre dois vértices da garagem. Se o segundo vértice é um arco *pull-out* isso seria impossível, pois esses arcos não podem ser movidos para um ponto anterior no tempo. No caso de haver vértices subsequentes com arcos *pull-out*, esses podem ser agrupados no último vértice, sem perdas na representação da rede. A figura 4.2 mostra a rede resultante após terem sido aplicadas as reduções de

pull-in/out.

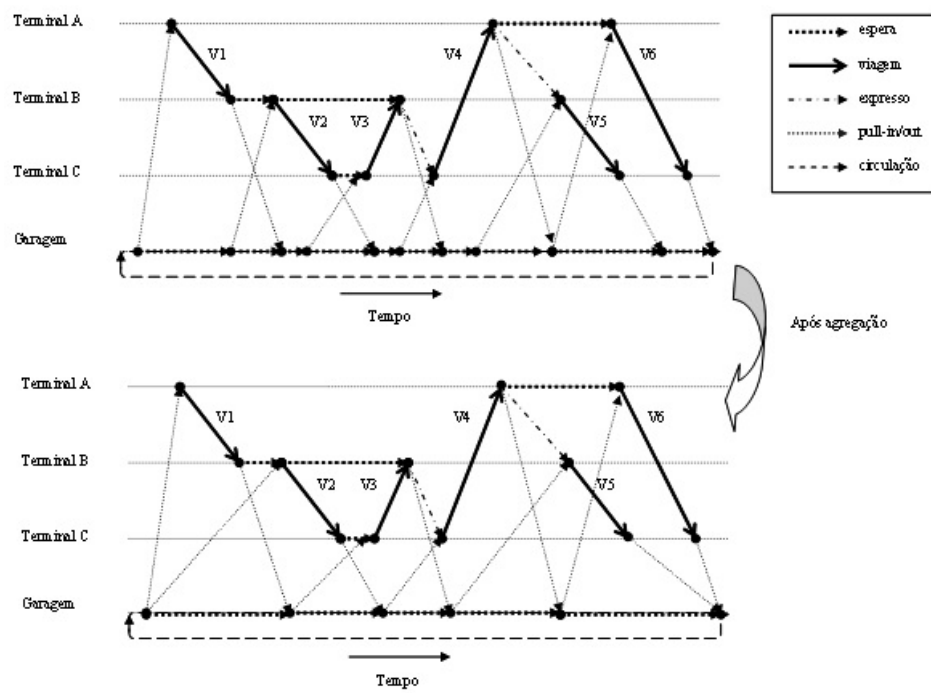


Figura 4.2: Estágios de Agregação dos Vértices de Arcos *Pull In-Out* (Adaptada de Kliwer *et al.* (2002)).

Complementarmente, aplicou-se uma nova forma de redução, até então não encontrada em outros trabalhos que utilizam a rede tempo-espço: agrupamento de arcos pull-in e pull-out em um mesmo nó. Quando vértices *pull-in* são subsequentes a um vértice *pull-out*, eles podem ser agrupados a este último. Ao realizar esta redução, tem-se uma diminuição ainda maior do que as mencionadas na literatura no número de restrições do modelo (Visentini *et al.*, 2013).

4.2 CONVERSOR DE TSN PARA CONEXÃO

Ao invés de desenvolver um método de geração de colunas especializado para o uso da rede tempo-espço, foi desenvolvida uma ferramenta de conversão que de posse de uma rede tempo-espço constrói uma matriz de custo (vide, matriz 4.1) representativa de uma rede de conexão. A decisão de implementar um conversor foi tomada, pois na literatura esse método somente foi implementado baseado nessa rede e os esforços para propor e implementar um método totalmente inovador seria escopo suficiente, por si só, de um trabalho a parte (como pode ser visto em Ribeiro e Soumis (1994)).

O algoritmo 1 descreve o procedimento desenvolvido.

Algoritmo 1 Conversor de TSN para Conexão

Para cada $(a \in A^s)$

Gere um novo nó $i \in V$ onde i representará um nó na rede de conexão respectiva (n , computacionalmente, será uma linha e uma coluna na matriz de custos).

Para cada $(a \in (A^{wait} \cup A^{dh}))$

Crie o arco na rede de conexão (i, j) onde i e j representam as viagens vinculadas aos nós de origem e destino do arco a na rede tempo-espço, respectivamente.

Para cada $(a \in (A^{pout} \cup A^{pin}))$

Crie o arco na rede de conexão $(o(k), i)$ ou $(j, d(k))$ onde i e j representam as viagens vinculadas aos nós de origem e destino do arco a na rede tempo-espço, respectivamente.

Para cada $(k \in K)$

Gere a capacidade da garagem k .

$$M = \begin{pmatrix} -1 & -1 & c_{k_1,1} & c_{k_1,2} & \dots & c_{k_1,n} \\ -1 & -1 & c_{k_2,1} & c_{k_2,2} & \dots & c_{k_2,n} \\ c_{1,k_1} & c_{1,k_2} & -1 & c_{1,2} & \dots & c_{1,n} \\ c_{2,k_1} & c_{2,k_2} & c_{2,1} & -1 & \dots & c_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n,k_1} & c_{n,k_2} & c_{n,1} & c_{n,2} & \dots & -1 \end{pmatrix} \quad (4.1)$$

Onde:

- $c_{i,j}$ representa o custo de atender a viagem j após ter atendido a viagem i
- $c_{k,j}$ ou $c_{i,k}$ representa o custo das viagens de *pull-in* e *pull-out* para ou da viagem i
- -1 representa que o arco não existe na rede (é inviável)

4.3 GERAÇÃO DE COLUNAS TRUNCADA MODIFICADA

Para a resolução via método de geração de colunas, a formulação de conjunto de partições foi utilizada. Baseado em Lopes (2013), da primeira iteração até a primeira estabilização (considerado apto para arredondar), a formulação da restrição (3.2) foi relaxada, assim, acelerando o processo de convergência dos valores. A restrição modificada pode ser descrita conforme a seguir:

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip} \theta_p \geq 1 \quad \forall i \in Trips \quad (4.2)$$

Onde p representa um bloco de veículo e a_{ip} indica se a viagem i pertence ao bloco p e θ_p é a variável de decisão da formulação de partição de conjuntos para o MDVSP. O problema mestre reduzido com a substituição da restrição de atendimento da demanda pela restrição 4.2 será denominado de problema mestre reduzido relaxado.

Para evitar a estagnação, o processo de geração de colunas truncado controla o limite de iterações através de um valor mínimo esperado de redução (Z_{min}) e um número máximo de iterações (I), pré-definidos antes de o método ser executado. Adicionalmente

coloca-se mais um parâmetro de controle *Num_Stabilized* que representa o número máximo de iterações na qual o método pode estagnar em um valor antes de encerrar a execução, evitando assim uma saída precipitada e impactando a solução. O algoritmo 2 apresenta o método de geração de colunas implementado para a solução do MDVSP.

Para um melhor entendimento, os parâmetros utilizados na implementação do método de geração de colunas utilizado nesse trabalho e seus devidos significados constam na Tabela 4.1.

Parâmetro	Significado
<i>Trips</i>	Representa o número de viagens
<i>Depots</i>	Representa o número de garagens
Z_{min}	Valor mínimo esperado de redução na função objetivo entre duas iterações
<i>Iter_min</i>	Número mínimo de iterações entre dois arredondamentos.
<i>Omega_min</i>	Valor mínimo que uma variável deve possuir para que seja setada em 1 na fase de arredondamento
<i>Num_Stabilized</i>	Valor máximo de iterações em que a função objetivo pode ficar constante. Quando excedido efetua-se o teste de integralidade e se necessário executa-se a fase de arredondamento.

Tabela 4.1: Parâmetros do Método de Geração de Colunas

No algoritmo 2 foi utilizado θ que é a variável de decisão binária do problema formulado na forma de partição de conjuntos e δ_n são as variáveis artificiais inseridas para a resolução do problema.

Algoritmo 2 Método de Geração de Colunas Truncado Modificado

Passo 1 (Inicialização): Defina os parâmetros $Z_{min}, Iter_{min}, Num_{Stabilized}$ e $Omega_{min}$. Atribua $n \leftarrow 1, n_0 \leftarrow 1, stabilized \leftarrow 0$ e $Stagnation \leftarrow false$. Adicione as variáveis artificiais δ_i na restrição na restrição de atendimento da demanda e com coeficientes big- M na função objetivo.

Passo 2 (Problema Mestre Restrito Relaxado): Solucione o Problema Mestre Restrito Relaxado, obtendo uma solução primal (θ_n, δ_n) , e custo Z_n^{RMP} . Vá para o passo 4.

Passo 3 (Problema Mestre Restrito): Solucione o Problema Mestre Restrito, obtendo uma solução primal (θ_n, δ_n) , e custo Z_n^{RMP} .

Passo 4 (Teste de Estabilização): Se $(stabilized > Num_{Stabilized})$ então (Se $(Stagnation = false)$ vá para o passo 7; senão, vá para o passo 8); senão vá para o passo 5.

Passo 5 (Teste de Término Antecipado): Se $(n - n_0 > Iter_{min})$ e $(Z_{n_0}^{RMP} - Z_n^{RMP} < Z_{min})$ então Se $(Stagnation = false)$ vá para o passo 7; senão, vá para o passo 8.

Passo 6 (Subproblemas): Para cada $k \in K$ solucione um problema de caminho mais curto usando o algoritmo SLF (Bertsekas, 1993). Se o subproblema apresentar custo negativo então selecione o S_n^k , atribua $\Omega_{n+1}^k \leftarrow \Omega_n^k \cup S_n^k$ e $n \leftarrow n+1$ e Solucione o RMP novamente. Se $Z_{n-1}^{RMP} - Z_n^{RMP} = 0$ então $stabilized \leftarrow stabilized + 1$ senão $stabilized \leftarrow 0$. Se $(Stagnation = false)$ vá para o passo 2; senão vá para o passo 3.

Passo 7 (Teste de Viabilidade e do Problema Mestre): Se $(\delta_n \neq 0)$ PARE; senão defina $Stagnation \leftarrow true$.

Passo 8 (Integralidade): Se $(\theta_n \in \{0, 1\})$ então a solução foi encontrada. Retorne θ_n . PARE.

Passo 9 (Arredondamento): Efetue um arredondamento nas variáveis não-inteiras em θ_n . Se $(\theta_{p,n} \geq \theta_{min})$ atribua $\theta_p \leftarrow 1$. Se nenhuma dessas variáveis atende ao quesito em θ_n , selecione uma. Atribua $n \leftarrow n + 1$ e $n_0 \leftarrow n$. Vá para o passo 3.

São contribuições deste trabalho a inserção de um parâmetro de controle *Num_Stabilized* que tem como função perturbar a geração de colunas quando o método estagnar em um valor ϵ , por dificuldade de gerar novas colunas, não conseguir aprimorar a solução. Os experimentos mostraram que a utilização deste parâmetro reduziu o tempo computacional da solução e, em alguns casos, obteve uma leve melhoria na função objetivo.

Uma questão importante relacionada à implementação da geração de colunas para o MDVSP é a solução dos subproblemas de caminho mais curto em grafos.

Viagens	Garagens	Tempo Total (em s) - TT	Caminho mais curto (em s) - TS	TT / TS (em %)
500	4	15	9	60%
500	16	12	9	75%
500	32	13	10	77%
1000	4	43	36	84%
1000	8	103	71	69%
2000	16	822	469	57%

Tabela 4.2: Tempo Gastos na Resolução dos Subproblemas

É possível verificar pela Tabela 4.2 entre 57% e 84% do tempo total é gasto na solução dos subproblemas. Após um breve estudo sobre os algoritmos existentes e algumas implementações (Ford, 1956; Bellman, 1958; Moore, 1959; Floyd, 1962; Jonker and Volgenant, 1987; Bertsekas, 1991, 1993; Cherkassky *et al.*, 2009), optou-se por utilizar o SLF (Small Label to the Front) (Bertsekas, 1993).

5 RESOLVENDO O MDVTSP

De mesma forma como apresentado no capítulo 4, neste capítulo apresenta-se com mais detalhes como resolveu-se o escalonamento de veículos com múltiplas garagens com frota heterogênea. A seção 5.1 apresenta as alterações na formulação de partição de conjuntos apresentada por Ribeiro e Soumis (1994) que foram efetuadas com o propósito de solucionar o MDVTSP através do método de geração de colunas. Na seção 5.2 apresenta-se como foi gerada e reduzida a rede tempo-espaço para o MDVTSP. A seção 5.3 apresenta o conversor utilizado nesse problema. Por fim, na seção 5.4 descreve-se em detalhes o método de geração de colunas aplicado a esse problema.

5.1 FORMULAÇÃO DE PARTIÇÃO DE CONJUNTOS PROPOSTA PARA O MDVTSP

Ribeiro e Soumis (1994) propuseram uma formulação de partição de conjuntos para o MDVSP, entretanto não foi encontrada na literatura nenhuma formulação de partição de conjuntos para o MDVTSP. Portanto, sendo $f \in F$ um tipo de veículo, $k \in K$ uma garagem, $p \in \Omega^{kf}$ um bloco de veículo de um tipo de veículo f da garagem k , onde Ω^{kf} representa todos os blocos de veículos, c_p é o custo do bloco de veículo p , a_{ip} define se a viagem i é atendida no bloco de veículo p , q_f é a capacidade do veículo f , sendo D_i a demanda da viagem i e v_{kf} é a quantidade de veículos do tipo f na garagem k . Desta forma propõe-se a seguinte formulação:

$$\min \sum_{f \in F} \sum_{k \in K} \sum_{p \in \Omega^{k,f}} c_p \theta_p \quad (5.1)$$

st

$$\sum_{f \in F} \sum_{k \in K} \sum_{p \in \Omega^{k,f}} a_{ip} \theta_p = 1, \quad \forall i \in Trips \quad (5.2)$$

$$\sum_{f \in F} \sum_{k \in K} \sum_{p \in \Omega^{k,f}} q_f a_{ip} \theta_p \geq D_i \quad \forall i \in Trips \quad (5.3)$$

$$\sum_{p \in \Omega^{k,f}} \theta_p \leq v_{kf}, \quad \forall k \in K, \forall f \in F \quad (5.4)$$

$$\theta_p \in \{0, 1\}, \quad \forall p \in \Omega^{k,f}. \quad (5.5)$$

Onde a função objetivo (5.1) visa minimizar os custos totais. A restrição de particionamento de conjuntos (5.2) impõe que cada tarefa seja realizada por exatamente um veículo. A restrição (5.3) define que a demanda de cada viagem deve ser atendida, enquanto que a restrição (5.4) expressa a restrição de capacidade por garagem. Finalmente, o requisito de integralidade das variáveis é fornecido por (5.5).

Importante salientar que a variável θ é indexada somente em p , sendo p um bloco de veículo de um determinado tipo f oriundo de um garagem k . Formulado desta forma reduziu-se a complexidade polinomialmente, visto que utiliza-se somente uma dimensão na variável ao invés de duas, no caso de diferenciar-se os veículos na variável.

5.2 GERAÇÃO E REDUÇÃO DA REDE TEMPO-ESPAÇO

A geração e redução da rede tempo-espaco para escalonamento de veículos com múltiplas garagens e frota heterogênea seguiu os mesmos princípios apresentados na seção 4.1. A única modificação foi que a tabela de horários das viagens que inicia o processo de geração da rede apresenta a informação de demanda estimada por viagem. Essa informação embora não seja utilizada nesse passo, é armazenada para futura utilização.

5.3 CONVERSOR DE TSN PARA CONEXÃO

Com o propósito de utilizar o método de geração de colunas truncado modificado, foi desenvolvida uma ferramenta de conversão que de posse de uma rede tempo-espaco constrói uma matriz de custo que representa a rede de conexão modificada, baseada no

algoritmo 1. É importante salientar que a rede de conexão gerada não leva em consideração a frota heterogênea na construção de seus arcos (vide seção 5.4). Como foi utilizado o conceito de fatores de custos esse procedimento foi realizado no método de geração de colunas. Como consequência não se teve uma explosão de arcos desnecessária que penalizaria o desempenho. O algoritmo 3 foi utilizado.

Algoritmo 3 Conversor de TSN para Conexão com Frota Heterogênea

Para Cada ($a \in A^s$)

Gere um novo nó $n \in V$ onde n representará um nó na rede de conexão respectiva (n , computacionalmente, será uma linha e uma coluna na matriz de custos).

Para Cada ($a \in (A^{wait} \cup A^{dh})$)

Crie o arco na rede de conexão (n_i, n_j) onde i e j representam as viagens vinculadas aos nós de origem e destino do arco a na rede tempo-espaço, respectivamente.

Para Cada ($a \in (A^{pout} \cup A^{pin})$)

Crie o arco na rede de conexão $(o(k), n_i)$ ou $(n_j, d(k))$ onde i e j representam as viagens vinculadas aos nós de origem e destino do arco a na rede tempo-espaço, respectivamente.

Para Cada ($k \in K$)

Gere a capacidade da garagem k levando em consideração os diversos tipos de veículos f .

5.4 GERAÇÃO DE COLUNAS MODIFICADA PARA O MDVTSP

O algoritmo proposto para MDVTSP foi baseado no algoritmo proposto para o MDVSP (algoritmo 2) nesse documento. Uma modificação significativa foi a inserção de uma projeção dos custos fixos dos diferentes tipos de veículos na reta dos reais positivos com o propósito de obter fatores de custos. A Tabela 5.1 e a Figura 5.1 exemplifica para o caso de 5 veículos. Essa projeção é automática ou pode ser passada como parâmetro pelo tomador de decisão. O decisor pode então optar por passar os custos fixos dos veículos e os fatores serão calculados ou podem ser passados como parâmetros caso seja estrategicamente mais interessante ao decisor.

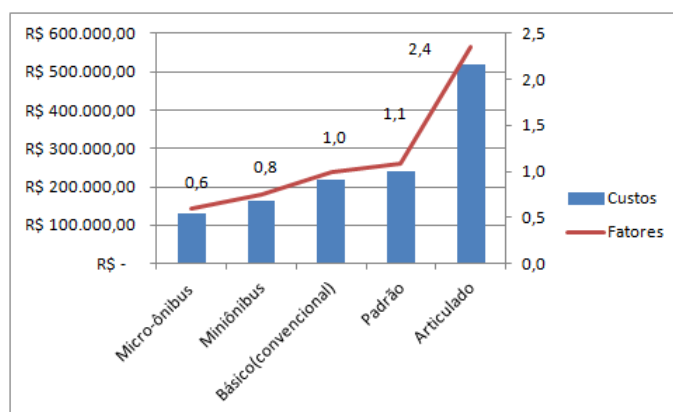


Figura 5.1: Gráfico da relação dos custos com os fatores.

Tipo	Custo Fixo	Fator utilizado
Micro-ônibus	R\$ 132.238,00	0,6
Miniônibus	R\$ 165.000,00	0,8
Básico(convencional)	R\$ 220.000,00	1,0
Padrão	R\$ 240.000,00	1,1
Articulado	R\$ 518.373,00	2,4

Tabela 5.1: Exemplo de Geração de Fatores de Custos

Foi implementada uma subrotina que efetua, durante a leitura da rede, reduções nos arcos onde a demanda é maior do que a capacidade do veículo, fazendo com que essa variável nem seja considerada. Nesta subrotina, a matriz de custos obtidas pelo conversor se transforma em uma matriz de custos reduzida e com os fatores de custos aplicados.

Em relação ao algoritmo proposto para a solução do MDVSP ressaltamos que para o

MDVTSP não utilizou-se o relaxamento da restrição de atendimento das viagens por somente um veículo e somente uma vez. Isso ocorreu, pois, conforme pode ser visto na Tabela 5.2, o relaxamento da restrição de que somente um veículo pode atender uma viagem demonstrou piora significativa em diversos experimentos. A piora no desempenho computacional ocorre nas instâncias com grande número de garagens. Os melhores resultados estão ressaltados em negrito. Todos os testes da Tabela 5.2, foram realizados com parâmetros: $Z_{MIN} = 0$, $Iter_Min = 50$, $Omega_Min = 0,8$ e $Num_Stabilized = 10$.

Viagens	Garagens	Tipos de Veículos	Sem a Redução			Com a Redução		
			Tempo	Custo	Núm. Veículos	Tempo	Custo	Núm. Veículos
500	16	3	23	1336120	123	60	1335280	123
500	32	3	47	1374520	130	125	1374520	130
500	128	3	202	1283980	123	431	1283980	123
1000	32	3	196	2717750	257	472	2717750	257
500	16	5	37	1319260	123	81	1319260	123
500	32	5	70	1320590	130	177	1320590	130
500	128	5	237	1226700	125	528	1226700	125
1000	32	5	293	2588150	259	506	2588150	259
2000	16	5	1355	5475250	531	1941	5464030	531
500	16	8	63	1738980	125	91	1720490	124
500	128	8	428	1597460	126	600	1597460	126
1000	32	8	547	3369080	261	789	3367860	261

Tabela 5.2: Análise sobre o Uso do Relaxamento na Restrição (3.8)

Outro ponto a se ressaltar é que o algoritmo de caminho mais curto foi modificado para atender aos diferentes tipos de veículos, uma vez que a formulação do MDVTSP foi modificada em relação a do MDVSP proposta por Ribeiro e Soumis (1994). A formulação com o uso das variáveis duais pode ser vista a seguir:

$$\min \quad -\sigma_{kf} + \sum_{(i,j) \in A^{kf}} (c_{ij}^f - \pi_j - q_f \gamma_j) x_{ij}^{kf} \quad (5.6)$$

st

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

$$\forall k \in K, \quad f \in F \quad (5.7)$$

$$x_{ij}^{kf} \in \{0, 1\} \quad \forall (i,j) \in A^k, k \in K, f \in F. \quad (5.8)$$

onde:

i) γ representa o dual da restrição 3.7.

- ii) π representa o dual da restrição 3.8.
- iii) σ representa o dual da restrição 3.9.
- iv) q_f representa a capacidade do veículo f .

Através dos experimentos comprovamos que o elemento $q_f \gamma_j$ da equação contribui significativamente para a estagnação rápida do método uma vez que gera mais colunas com custos reduzidos, maiores em módulo, negativos. Com isso tem-se menor tempo de processamento para se obter a solução.

Outro ponto que devemos salientar é que a utilização dos fatores de custos fez com que o subproblema da geração de colunas deixasse de ser um *Resource Constrained Shortest Path* facilitando a resolução, pois o RCSP é NP-Completo (Mehlhorn and Ziegelmann, 2000; Beasley and Christofides, 1989).

A proposta apresentada nesse trabalho consiste em executar para cada tipo de veículo localizado em cada garagem um algoritmo de caminho mais curto modificado. Em outras palavras, para cada tipo de veículo em cada garagem obtemos um bloco de veículo de custo mínimo que tornar-se-á uma coluna no problema mestre. Essa abordagem é inédita e apresentou bons resultados, por ser claramente de complexidade polinomial, diante da alternativa encontrada na literatura que seria um caminho mais curto com restrição de recursos que é notoriamente de complexidade NP-Difícil (Huisman *et al.*, 2005).

A implementação utilizada foi baseada no algoritmo 2. O algoritmo modificado 4 foi desenvolvido para atender o MDVTSP.

Algoritmo 4 Método de Geração de Colunas Truncado Modificado

Passo 1 (Inicialização): Defina os parâmetros $Z_{min}, ITER_{min}, Num_Stabilized$ e θ_{min} . Atribua $n \leftarrow 1, n_0 \leftarrow 1$ e $stabilized \leftarrow 0$. Adicione as variáveis artificiais δ^i em (2.7) e com coeficientes big- M em (2.6).

Passo 2 (Problema Mestre Restrito): Solucione o Problema Mestre Restrito, obtendo uma solução primal (θ_n, δ_n) , e custo Z_n^{RMP} .

Passo 3 (Teste de Estabilização): Se $(stabilized > Num_Stabilized)$ então vá para o passo 6; senão vá para o passo 4.

Passo 4 (Teste de Término Antecipado): Se $(n - n_0 > ITER_{min})$ e $(Z_{n_0}^{RMP} - Z_n^{RMP} < Z_{min})$ então vá para o passo 6; senão vá para o passo 5.

Passo 5 (Subproblemas): Para cada $k \in K$ e para cada $f \in F$ solucione um problema de caminho mais curto modificado usando o algoritmo SLF (Bertsekas, 1993) modificado. Se o subproblema apresentar custo negativo então selecione o S_n^k , atribua $\Omega_{n+1}^k \leftarrow \Omega_n^k \cup S_n^k$ e $n \leftarrow n + 1$ e Solucione o RMP novamente. Se $Z_{n-1}^{RMP} - Z_n^{RMP} = 0$ então $stabilized \leftarrow stabilized + 1$ senão $stabilized \leftarrow 0$. Vá para o passo 3.

Passo 6 (Teste de Viabilidade e do Problema Mestre): Se $(\delta_n \neq 0)$ PARE senão vá para o passo 7.

Passo 7 (Integralidade): Se $(\theta_n \in \{0, 1\})$ então a solução foi encontrada. Retorne θ_n . PARE.

Passo 8 (Arredondamento): Efetue um arredondamento nas variáveis não-inteiras em θ_n . Se $(\theta_{p,n} \geq \theta_{min})$ atribua $\theta_p \leftarrow 1$. Se nenhuma dessas variáveis atendem ao quesito em θ_n , selecione uma. Atribua $n \leftarrow n + 1$ e $n_0 \leftarrow n$. Vá para o passo 2.

6 EXPERIMENTAÇÃO COMPUTACIONAL

Este capítulo aborda os experimentos computacionais realizados no presente estudo. Os experimentos foram executados em uma máquina Intel® Core™ i7-3632QM com 2,2 GHz, 8 GB de memória rodando em uma plataforma Windows 8 versão 64bits. Foi utilizado o solver Cplex 12.5 e a linguagem de desenvolvimento foi o C++. Os métodos de geração de instâncias, construção e reduções da rede tempo-espaço e de geração de colunas foram desenvolvidos, sem o auxílio de *frameworks*, pelo autor. O presente capítulo divide-se em duas seções: Na seção 6.1, é apresentado o método de geração das instâncias e na seção 6.2 são abordadas as escolhas feitas para ajustar as calibrações dos procedimentos no algoritmo.

6.1 GERADOR DE INSTÂNCIAS

Uma das principais contribuições de Carpaneto *et al.* (1989) foi a criação de um gerador de instâncias aleatórias que vem sendo largamente utilizado na literatura (Ribeiro and Soumis, 1994; Pepin *et al.*, 2009; Rohde, 2008). Entretanto, a maior instância gerada a partir de Carpaneto *et al.* (1989) é de 1500 viagens com 8 garagens, e está no formato de rede de conexão. Infelizmente, as instâncias geradas representam uma realidade diferente da qual se pretende abordar. As instâncias obtidas consideram na geração das viagens uma distribuição de demandas modeladas através de uma curva que neste trabalho será chamada de função *Frequencia*. O gerador proposto apresenta como saída uma tabela de horários, com um número de garagens e suas respectivas capacidades e localizações. A função *Frequencia* utilizada trata-se da combinação (através de uma soma) de três

gaussianas, uma de média 7 e desvio-padrão 2, outra de média 12 e desvio-padrão 1,5 e por fim outra de média 18 e desvio-padrão 3 - as médias representam os horários de maior demanda -. Um exemplo dessa função pode ser visto na figura 6.1.

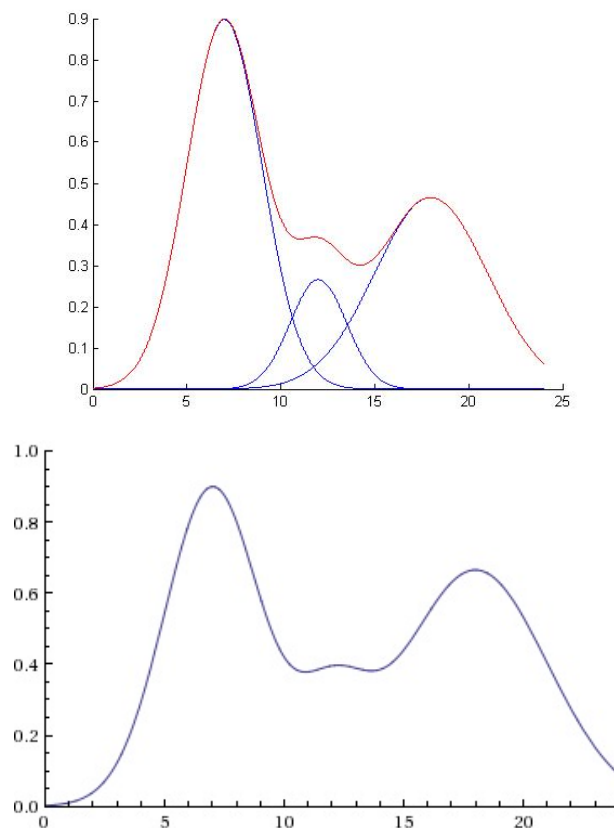


Figura 6.1: Distribuição de Demandas

Inicialmente é gerado um número randômico entre 0 e 23 esse número será o horário de início da viagem. Esse mesmo valor é dado como entrada na função *Frequencia* que retorna uma probabilidade entre 0 e 1. Para a viagem ser aceita é gerado um número randômico entre 0 e 1. Assim se faz um teste verificando se número gerado é menor do que o número retornado pela função aplicada ao horário de início da viagem. Como a função é resultante da soma das três normais de médias 7, 12 e 18, espera-se que nesses horários e em sua proximidade o número de viagens seja maior. A seguir são gerados dois números entre 0 e n para representar os terminais de origem e destino, onde n é número de vértices no grafo. O horário de fim da viagem é dado pelo horário de início mais a distância entre os dois vértices. O algoritmo 5 esboça o procedimento realizado. Para melhor compreender-lo assumo que r será um número aleatório que representará a hora

de início da viagem, u e v são números aleatórios que representarão os terminais de início e fim da viagem e por último que a demanda de cada viagem será calculada através da equação 6.1.

Equação 6.1: Demanda = (randômico entre 0 e (maior capacidade dos tipos de veículo))

* Frecuencia(r) + demanda mínima.

Algoritmo 5 Gerador de Viagens

Para ($i \leq |Trips|$)

Gere um número randômico r entre 0 e 23.

Gere um número randômico rv entre 0 e 1.

Se ($rv \leq Frecuencia(r)$)

Então

Gere dois número randômico u e v entre 0 e n .

Calcule o tempo de viagem (hora final) através das distâncias.

Calcule a demanda através da fórmula 6.1.

Escreva a nova viagem no arquivo.

6.2 CALIBRAGEM DOS PARÂMETROS

Conforme Oukil *et al.* (2007), e como verificado empiricamente nesse trabalho, uma das grandes dificuldades da utilização do método de geração de colunas é a parametrização, uma vez que esse método costuma ser sensível a ela. Desta forma experimentos computacionais e de calibragem dos parâmetros se fizeram necessários para uma melhor utilização do método. Portanto serão analisados os parâmetros conforme tabela 6.1.

Parâmetro	Valor Mínimo	Valor Máximo
$ZMin$	0	10000
$Iter_min$	20	100
$Omega_min$	0,7	0,99
$Num_Stabilized$	10	100

Tabela 6.1: Limites dos Parâmetros do Método de Geração de Colunas

Após iniciados os experimentos foi verificado que o parâmetro $Num_Stabilized$ não gerou impacto na função objetivo, somente no tempo, como pode ser visto na Tabela

6.3. Desta forma ele foi definido com o valor mínimo, como consequência reduziu-se algumas combinações de testes. A Tabela 6.4 apresenta os resultados dos demais testes considerando somente $Num_Stabilized = 10$.

Trips	Depots	Z_in	$Iter_Min$	$Omega_min$	$Num_Stabilized$	Tempo	Custo	Núm. Veículos
500	4	0,00E+00	20	0,7	10	20	2,20E+06	180
500	4	0,00E+00	20	0,7	50	20	2,20E+06	180
500	4	0,00E+00	20	0,7	100	20	2,20E+06	180
500	4	0,00E+00	20	0,8	10	19	2,13E+06	173
500	4	0,00E+00	20	0,8	50	19	2,13E+06	173
500	4	0,00E+00	20	0,8	100	19	2,13E+06	173
500	4	0,00E+00	20	0,9	10	23	2,13E+06	173
500	4	0,00E+00	20	0,9	50	55	2,13E+06	173
500	4	0,00E+00	20	0,9	100	48	2,13E+06	173
500	4	0,00E+00	20	0,99	10	19	2,13E+06	173
500	4	0,00E+00	20	0,99	50	40	2,13E+06	173
500	4	0,00E+00	20	0,99	100	41	2,13E+06	173
500	4	0,00E+00	50	0,7	10	20	2,14E+06	175
500	4	0,00E+00	50	0,7	50	21	2,14E+06	175
500	4	0,00E+00	50	0,7	100	21	2,14E+06	175
500	4	0,00E+00	50	0,8	10	14	2,13E+06	173
500	4	0,00E+00	50	0,8	50	17	2,13E+06	173
500	4	0,00E+00	50	0,8	100	17	2,13E+06	173
500	4	0,00E+00	50	0,9	10	35	2,13E+06	173
500	4	0,00E+00	50	0,9	50	116	2,13E+06	173
500	4	0,00E+00	50	0,9	100	127	2,13E+06	173
500	4	0,00E+00	50	0,99	10	21	2,13E+06	173
500	4	0,00E+00	50	0,99	50	45	2,13E+06	173
500	4	0,00E+00	50	0,99	100	41	2,13E+06	173
500	4	0,00E+00	100	0,7	10	18	2,13E+06	173
500	4	0,00E+00	100	0,7	50	17	2,13E+06	173
500	4	0,00E+00	100	0,7	100	17	2,13E+06	173
500	4	0,00E+00	100	0,8	10	16	2,13E+06	173
500	4	0,00E+00	100	0,8	50	19	2,13E+06	173
500	4	0,00E+00	100	0,8	100	22	2,13E+06	173
500	4	0,00E+00	100	0,9	10	51	2,13E+06	173
500	4	0,00E+00	100	0,9	50	167	2,13E+06	173
500	4	0,00E+00	100	0,9	100	340	2,13E+06	173
500	4	0,00E+00	100	0,99	10	22	2,13E+06	173
500	4	0,00E+00	100	0,99	50	145	2,13E+06	173
500	4	0,00E+00	100	0,99	100	301	2,13E+06	173

Tabela 6.2: Calibragem dos Parâmetros Usados na Geração de Colunas

TTrips	Depots	Z_{min}	$Iter_Min$	Ω_{min}	$Num_Stabilized$	Tempo	Custo	Núm. Veículos
500	4	1,00E+02	20	0,7	10	20	2,22E+06	180
500	4	1,00E+02	20	0,85	10	16	2,13E+06	173
500	4	1,00E+02	20	0,9	10	14	2,13E+06	173
500	4	1,00E+02	20	0,99	10	16	2,13E+06	173
500	4	1,00E+02	50	0,7	10	18	2,14E+06	174
500	4	1,00E+02	50	0,8	10	14	2,13E+06	173
500	4	1,00E+02	50	0,9	10	16	2,13E+06	173
500	4	1,00E+02	50	0,99	10	20	2,13E+06	173
500	4	1,00E+02	100	0,7	10	15	2,14E+06	174
500	4	1,00E+02	100	0,8	10	14	2,13E+06	173
500	4	1,00E+02	100	0,9	10	18	2,13E+06	173
500	4	1,00E+02	100	0,99	10	51	2,13E+06	173
500	4	1,00E+04	20	0,7	10	23	2,34E+06	190
500	4	1,00E+04	20	0,8	10	20	2,19E+06	179
500	4	1,00E+04	20	0,9	10	18	2,16E+06	175
500	4	1,00E+04	20	0,99	10	20	2,13E+06	173
500	4	1,00E+04	50	0,7	10	21	2,28E+06	185
500	4	1,00E+04	50	0,8	10	15	2,16E+06	176
500	4	1,00E+04	50	0,9	10	15	2,15E+06	175
500	4	1,00E+04	50	0,99	10	22	2,13E+06	173
500	4	1,00E+04	100	0,7	10	19	2,20E+06	179
500	4	1,00E+04	100	0,8	10	16	2,14E+06	175
500	4	1,00E+04	100	0,9	10	60	2,14E+06	175
500	4	1,00E+04	100	0,99	10	91	2,14E+06	174

Tabela 6.3: Calibragem dos Parâmetros Usados na Geração de Colunas - Parte 2.

A partir da leitura das tabelas 6.3 e 6.4 é possível verificar que o uso de $\Omega_{min} \in [0,8; 0,99]$ apresenta melhores resultados e não possuem uma variação alta do custo total, somente variando o tempo de execução. Portanto um bom valor para o parâmetro Ω_{min} seria 0,8. Pode-se concluir também que $Iter_{min} \in [50,100]$, da mesma forma, não gera impacto no valor da função objetivo, assim pode-se fixá-lo em $Iter_{min} = 50$ sem prejuízos significativos à solução. Para um melhor esclarecimento sugere-se a Figura 6.2.

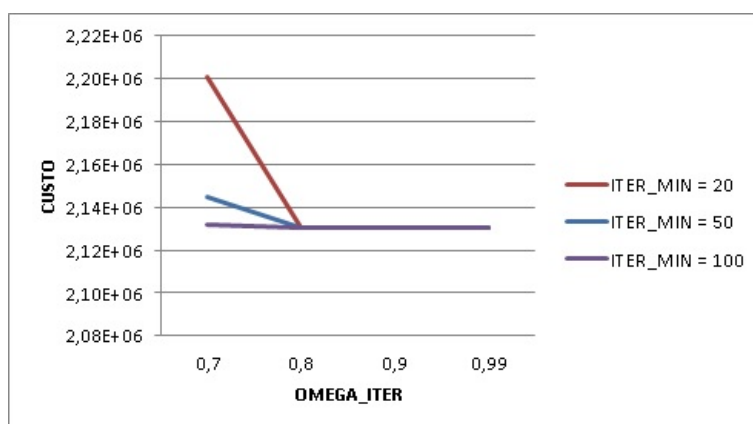


Figura 6.2: Calibrando os Parâmetros: Z_{min} igual a zero

Outra constatação é que o uso de Z_{min} entre 0 e 100 não impacta muito a função

objetivo e apresenta melhor resultado que valores maiores. Pode-se afirmar que o uso de $Z_{min} = 100$ pode ser uma boa parametrização. O gráfico 6.3 ajuda no entendimento.

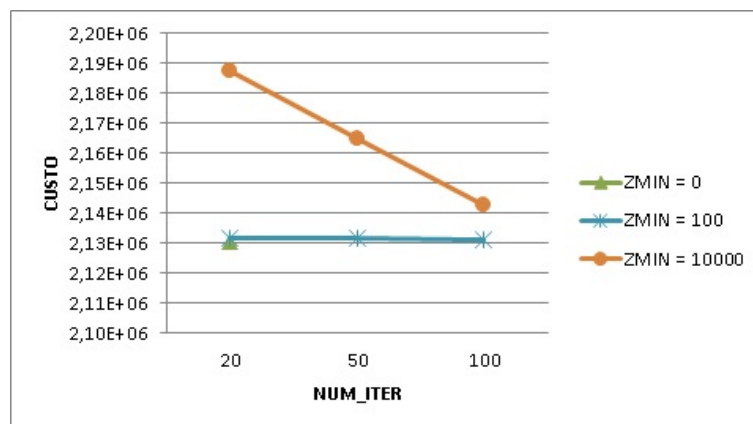


Figura 6.3: Calibrando os Parâmetros: Ω_{min} igual a 0,8

Após analisar as tabelas 6.3 e 6.4 e as figuras 6.2 e 6.3 conclui-se que uma parametrização aceitável é $Z_{MIN} = 100$, $Iter_{Min} = 50$, $\Omega_{Min} = 0,8$ e $Num_{Stabilized} = 10$. Claro que quanto maior o valor de Ω_{Min} , melhor será o resultado da função objetivo, em contrapartida maior será o tempo computacional, trata-se de um *trade-off* comum em pesquisa operacional. Uma vez que a perda na função objetivo é muito pequena ou nula em relação ao ganho de tempo, este trabalho irá utilizar $\Omega_{Min} = 0,8$.

Efetuuou-se mais alguns testes, com pequenas variações ao redor desses parâmetros e com maior número de viagens e garagens para verificar o impacto na função objetivo, estes testes são importantes, pois o objetivo é solucionar grandes instâncias e portanto, verificar se a parametrização se mantém adequada. A tabela 6.5 apresenta uma análise do impacto da escolha destes parâmetros e algumas variações para comparação.

Como pode-se verificar, a parametrização apresentou-se válida e aceitável para maiores números de viagens e garagens, pois apresentou nenhuma ou pequena variação na função objetivo e no tempo, validando, desta forma, a parametrização proposta.

Trips	Depots	Z_{in}	$Iter_Min$	$Omega_min$	$Num_Stabilized$	Tempo	Custo	Núm. Veículos
500	16	0,00E+00	50	0,8	10	12	1391680	122
500	16	1,00E+02	50	0,8	10	12	1391680	122
500	16	1,00E+04	50	0,9	10	12	1391680	122
500	16	1,00E+02	50	0,9	10	12	1391680	122
500	16	0,00E+00	50	0,99	10	12	1391680	122
500	32	0,00E+00	50	0,8	10	13	1427960	129
500	32	1,00E+02	50	0,8	10	13	1427960	129
500	32	1,00E+04	50	0,9	10	13	1427960	129
500	32	1,00E+02	50	0,9	10	13	1427960	129
500	32	0,00E+00	50	0,99	10	13	1427960	129
2000	16	0,00E+00	50	0,8	10	829	5929280	526
2000	16	1,00E+02	50	0,8	10	819	5929280	526
2000	16	1,00E+04	50	0,9	10	637	6156610	527
2000	16	1,00E+02	50	0,9	10	889	5929280	526
2000	16	0,00E+00	50	0,99	10	868	5929280	526
4000	8	1,00E+06	30	0,75	10	5957	14365000	1161
4000	8	0,00E+00	50	0,8	10	5980	13900000	1135
4000	8	1,00E+04	50	0,8	10	5453	14200000	1147
4000	8	1,00E+02	50	0,8	10	5747	13800000	1130
4000	8	1,00E+04	50	0,9	10	5056	13900000	1136
4000	8	1,00E+02	50	0,9	10	6062	13600000	1129
4000	8	0,00E+00	50	0,99	10	5515	13500000	1127
4000	16	1,00E+06	30	0,75	10	5892	14432500	1194
4000	32	1,00E+08	10	0,75	5	6294	13629400	1153
5000	16	1,00E+06	30	0,75	10	6740	19450000	1527
10000	4	1,00E+06	50	0,8	10	32578	45808300	3652
10000	4	1,00E+06	30	0,75	10	32620	45855300	3653

Tabela 6.4: Calibragem dos Parâmetros Usados na Geração de Colunas - Parte 3.

7 SOLUCIONANDO O MDVSP

Nesse capítulo soluciona-se o problema de escalonamento de múltiplas garagens. Na seção 7.1 apresenta-se um conjunto de soluções ótimas obtidas junto ao solver Cplex. Na seção 7.2 apresenta-se um conjunto de soluções obtidas após a aplicação do método de geração de colunas proposto nesse trabalho.

7.1 SOLUÇÃO ÓTIMA COM TSN

Com o propósito de verificar a escalabilidade da rede tempo-espaco após aplicadas as reduções descritas na seção 4.1, criou-se o modelo respectivo e submeteu-se ao solver IBM ILOG Cplex 12.5 para obter a solução ótima de algumas instâncias. O uso de uma linguagem de modelagem como o AMPL como intermediário na construção do modelo e geração de um arquivo de entrada no formato MPS (MPS é um formato de arquivo de texto ASCII padrão da indústria para os problemas de programação matemática) para o Cplex, auxiliou na redução do tempo e na escalabilidade obtida. Os resultados estão na Tabela 7.1.

Viagens	Garagens	Tempo (em s)	Var. (approx)
500	4	1,92	26 mil
500	8	38,36	52 mil
500	16	382,20	81 mil
500	64	4.685,40	320 mil
1000	4	16,46	50 mil
1000	8	92,59	78 mil
1000	16	2.244,61	203 mil
1000	32	26.200,17	406 mil
1500	4	25,68	77 mil
1500	8	2.188,01	230 mil
1500	16	19.569,26	309 mil
3000	4	408,55	151 mil
3000	8	3130,3	307 mil
5000	4	1.654,86	249 mil
5000	8	17.653,85	390 mil

Tabela 7.1: Solução Ótima Utilizando a Rede Tempo-Espaço

As linhas que estão em negrito representam os tamanhos de instâncias que não apresentavam solução ótima conhecida na literatura. Conforme a tabela 7.1, foram obtidos resultados ótimos com instâncias de até 5000 viagens e 8 garagens, com o uso da rede tempo-espaco. Esse resultado sugere que a utilização da rede tempo-espaco possibilita uma escalabilidade maior no que tange o tamanho das instâncias. Dadas a aleatoriedade do gerador de instâncias e a independência estatística na geração entre duas viagens quaisquer, considera-se satisfatórios os resultados, mesmo que não utilizando a instância de Carpaneto *et al.* (1989).

Outra conclusão é que o que degrada mais o tempo de solução é o número de garagens e não o número de viagens. Com propósitos de comparação, a maior instância formulada através de uma modelagem de conexão solucionada de forma ótima na mesma máquina foi 1000 viagens com 8 garagens. Instâncias maiores que esta apresentavam estouro de memória.

Definindo como aceitável a espera de 1h para a obtenção de um resultado ótimo, o uso da rede tempo-espaco somado com as reduções aplicadas permite solucionar instâncias com 3000 viagens e 8 garagens. Isso define um *lower bound* de desempenho que qualquer método heurístico alternativo proposto deve superar.

7.2 MÉTODO DE GERAÇÃO DE COLUNAS TRUNCADO MODIFICADO

Na tabela 7.2 o resultado do método de geração de colunas aplicado ao MDVSP utilizando a rede tempo-espaço e as reduções descritas.

Viagens	Garagens	Tempo (s)	Custo	Núm. Veículos
500	4	14	2130830	173
500	16	12	1391680	122
500	32	13	1427960	129
500	128	43	1346620	124
1000	4	155	3329190	256
1000	8	159	3051980	242
2000	16	819	5929280	526
2000	64	1.207	5447030	510
4000	8	5.957	14365000	1161
4000	16	5.892	14432500	1194
4000	32	6.294	13629400	1153
5000	16	13.815	16145300	1394
5000	32	7.345	17472500	1526
8000	8	24.485	33013400	2772
8000	16	30.257	29598900	2569
10000	4	32.578	45808300	3652

Tabela 7.2: Resultados Usando Geração de Colunas

Os resultados são satisfatórios visto que se consegue solucionar instâncias de até 10000 viagens e instâncias com até 128 garagens. Esses resultados nos levam a crer que o método está eficiente o suficiente para que possa ser utilizado como base a outros métodos mais complexos na solução dos problemas correlatos ao MDVSP. Foram obtidos excelentes resultados em um tempo pequeno, onde a instância com maior *gap* em relação à solução ótima (estimado) foi de 3,5%, sendo que em diversos casos foi encontrada a solução ótima.

Baseado no *lower bound* de desempenho sugerido em 7.1, verifica-se que é superado com êxito uma vez que com o tempo de 20 minutos é possível solucionar um instância de 2000 viagens com 64 garagens e com o tempo de 1h e 40 minutos uma instância de 4000 viagens e 16 garagens. Lembrando que, conforme Tabela 7.1, o acréscimo de garagens impacta significativamente mais do que o aumento de viagens. Como trata-se de planejamento, um *gap* máximo (estimado) encontrado de 3,5% demonstrou ser insignificante.

8 SOLUCIONANDO O MDVTSP

Para verificar o método proposto para o MDVTSP, optou-se por instanciar na forma de um escalonamento de ônibus no contexto do transporte público de passageiros. Desta forma foi necessário obter novas informações, principalmente, no que tange uma diferenciação nos custos de utilização de cada veículo. Para atender tal objetivo utilizou-se como base a tabela 8.1, onde constam os custos fixos de cada tipo de veículo utilizado no transporte de passageiros urbano na cidade de São Paulo. A partir dela foi possível efetuar uma escala de fatores, além de representar os custos fixos, que foram agregados nos custos das viagens efetuadas pelos respectivos veículos.

Tipo	Preço (R\$)
Micro-ônibus	132.238,00
Miniônibus	165.000,00
Básico(Convencional)	220.000,00
Padrão	240.000,00
Articulado	518.373,00
Biarticulado	898.516,00
Padrão Low Center	282.040,00

FONTE: SPTrans/Superintendência de Receita e Remuneração – DG/SRR

Tabela 8.1: Custo de cada Tipo de Veículo

A tabela 8.2 mostra as capacidades de passageiros de cada um dos tipos veículos existentes na cidade de São Paulo, de acordo com SPTrans/Superintendência de Receita e Remuneração – DG/SRR.

Categoria	Capacidade
Microônibus	21
Miniônibus	40
Midiônibus	53
Básico (Convencional)	75
Padrão	83
Padrão 15M	98
Articulado	120
Biarticulado	190

Tabela 8.2: Tipos de Veículos

8.1 MÉTODO DE GERAÇÃO DE COLUNAS TRUNCADO MODIFICADO

Nas tabelas 8.3, 8.4 e 8.5 verifica-se o resultado do método de geração de colunas modificado aplicado ao MDVTSP utilizando a rede tempo-espaco e as reduções descritas. Os resultados são satisfatórios visto que conseguiu-se solucionar instâncias de até 5000 viagens ou com até 128 garagens. Esses resultados são significativos uma vez que são os primeiros a serem apresentados para esse problema na literatura.

Viagens	Garagens	Capacidades	Tempo (em s)	Custo	Núm. Veículos
500	16	84, 95, 108	21	1335570	123
500	32	84, 95, 108	58	1374520	130
500	128	84, 95, 108	128	1283980	123
1000	4	84, 95, 108	83	3087350	255
1000	8	84, 95, 108	139	2726970	240
2000	16	84, 95, 108	1323	5628780	525
2000	64	84, 95, 108	2323	5178550	510
4000	8	84, 95, 108	6264	14200500	1147
4000	16	84, 95, 108	7994	12224100	1153
4000	32	84, 95, 108	12650	11353700	1110
5000	4	84, 95, 108	10834	19134000	1505
5000	16	84, 95, 108	24054	15983500	1403

Tabela 8.3: Frota Heterogênea : 3 Tipos de Veículos

Viagens	Garagens	Capacidades	Tempo (em s)	Custo	Núm. Veículos
500	16	21,75,83,98,120	36	1319260	123
500	32	21,75,83,98,120	71	1320590	130
500	128	21,75,83,98,120	269	1226700	125
1000	4	21,75,83,98,120	127	2968640	258
1000	8	21,75,83,98,120	251	2642080	241
2000	16	21,75,83,98,120	1382	5475250	531
2000	64	21,75,83,98,120	2414	4972190	515
4000	8	21,75,83,98,120	6880	13045500	1186
4000	16	21,75,83,98,120	9736	11528300	1160
4000	32	21,75,83,98,120	12261	10961800	1122
5000	4	21,75,83,98,120	8531	18169900	1578
5000	16	21,75,83,98,120	21690	14862800	1443

Tabela 8.4: Frota Heterogênea : 5 Tipos de Veículos

Viagens	Garagens	Capacidades	Tempo (em s)	Custo	Núm. Veículos
500	16	21,40,53,75,83,98,120,190	66	1738980	125
500	128	21,40,53,75,83,98,120,190	381	1597460	126
1000	4	21,40,53,75,83,98,120,190	412	3998550	259
1000	8	21,40,53,75,83,98,120,190	576	3464780	248
2000	16	21,40,53,75,83,98,120,190	3734	7246590	534
2000	64	21,40,53,75,83,98,120,190	8316	6917590	528
4000	8	21,40,53,75,83,98,120,190	9498	12078800	1146
4000	16	21,40,53,75,83,98,120,190	8750	11378600	1165
4000	32	21,40,53,75,83,98,120,190	15849	10639000	1126
5000	4	21,40,53,75,83,98,120,190	8788	17562700	1548

Tabela 8.5: Frota Heterogênea : 8 Tipos de Veículos

Através das tabelas 8.3, 8.4 e 8.5 é possível verificar que instâncias com até 5000 viagens e instâncias com até 128 garagens foram solucionadas em tempos que não excedem 7h de processamento. Novamente utilizando o *benchmark* proposto em 7.1 verifica-se que foi possível solucionar instâncias de 2000 viagens, 16 garagens e 8 tipos de veículos em pouco mais de 1h e instâncias de 2000 viagens, 64 garagens e 5 tipos de veículos em 40 min.

Verificou-se também que o aumento no número de tipos de veículos não implicou diretamente no aumento do tempo total de processamento, ao contrário, em diversos casos há uma redução no tempo total de processamento. Esse fato ocorre, pois com uma diversidade maior de opções de tipos de veículos há um melhor ajuste entre esses e as demandas de cada viagem.

Como o processo consiste na inserção iterativa de colunas e cada coluna no método trata-se de um bloco de veículo, esse melhor ajuste proporcionou um menor número de colunas geradas para se obter a solução de qualidade, levando a um menor tempo computacional. Por exemplo, uma viagem com 20 passageiros na instância com 3 tipos de veículos foi solucionada utilizando um veículo de 84 lugares e na instância com 5 tipos de veículos a solução encontrada foi utilizar um microônibus de 21 lugares, obtendo uma redução de custo de aproximadamente 45%.

Não foi possível solucionar uma instância com 5000 viagens, 8 garagens e 8 tipos de veículos, pois ocorreu um estouro de memória no software Cplex no computador com as capacidades descritas no capítulo 6, caracterizando o *upper bound* de desempenho na configuração de máquina utilizada nesse estudo para a frota heterogênea. Em uma máquina com mais recursos é possível que se obtenha algum resultado diferente.

9 CONCLUSÕES

O escalonamento de veículos com múltiplas garagens é extremamente complexo, especialmente se o modelo busca solucionar grandes instâncias em um tempo aceitável. A literatura demonstrou extrema dificuldade de solucionar esse tipo de problema com instâncias não estruturadas. Outro complicador que pode ser agregado ao escalonamento de veículos com múltiplas garagens é o uso de diferentes tipos de veículos.

Esta dissertação desenvolveu um método de solução, baseado na integração da rede tempo-espaço e da geração de colunas para a solução de grandes instâncias não estruturadas do *Multiple Depot Vehicle Scheduling Problem* e do *Multiple Depot Vehicle-Type Scheduling Problem*, envolvendo dezenas de garagens e milhares de viagens. A resolução de instâncias de tamanhos maiores aos encontrados atualmente na literatura para o MDVTSP e de grandes instâncias para o MDVSP indicam a capacidade de solucionar grandes problemas reais. Pelos resultados obtidos é conclusivo que o objetivo geral e cada objetivo específico do trabalho foi plenamente atendido, buscando resolver o problema MDVSP e MDVTSP em tempo aceitável. Para tanto, o método aplicado pode ser considerado como base para a solução de outros problemas, no qual o relevante é uma solução de compromisso entre tempo e precisão, como exemplo, gestão de imprevistos, escalonamento de veículos e tripulação integrados e escalonamento em tempo-real.

A abordagem aqui realizada é inédita para MDVTSP, visto que não se encontra solução como tal apresentada na literatura. Nosso método soluciona o MDVTSP de forma quase ótima, com grandes instâncias e com pouco tempo computacional.

É interessante observar que o modelo desenvolvido utilizou técnicas consagradas e

validadas da literatura e contribuiu com a melhoria e evolução das mesmas. Pode-se concluir que o uso de redes tempo-espaço contribuiu com o tempo de solução e escalabilidade em relação à rede de conexão e que o método de geração de colunas modificado proposto mostrou-se um excelente método de solução a ser empregado.

9.1 SÍNTESE DAS CONTRIBUIÇÕES

1. Gerador de instâncias não-estruturadas levando em consideração uma distribuição de demandas conforme uma tabela de horários.
2. Método de geração de colunas modificado que permitiu escalabilidade e eficiência na solução do MDVSP.
3. Solução de grandes instâncias do MDVSP com até 10000 viagens e 128 garagens.
4. Formulação e modelagem inéditas do MDVTSP.
5. Método de solução via geração de colunas para o MDVTSP.

Como limitações do trabalho, tem-se que não foi aplicado a uma instância real, isto ocorreu pela dificuldade encontrada junto às empresas de liberar os dados necessários. O mercado de transporte está passando por uma fase de mudanças e sofrendo cada vez mais pressões da sociedade por melhores condições de trabalho, isto fez com que os gestores estivessem retraídos e com receio de efetuar parceria neste momento. Após as diversas manifestações públicas por melhorias no transporte, os decisores, inevitavelmente, precisarão lançar mão de recursos computacionais e científicos para obter uma solução de melhor qualidade, agregando valor aos trabalhos desses tópicos.

9.2 TRABALHOS FUTUROS

Como trabalhos futuros sugere-se:

- Aprimoramentos no método de solução para que seja possível solucionar os seguintes problemas de forma integrada com o escalonamento de veículos: Otimização da

tabela de horários e escalonamento da tripulação, levando em consideração a legislação brasileira, para os modais aéreo e rodoviário.

- Implementar e testar outros métodos de caminho mais curto na solução dos subproblemas do método de geração de colunas.
- Utilizar o estudo aqui realizado no contexto de transporte de cargas expressas.
- Aplicar o método no contexto aéreo e, mesmo que com adaptações necessárias, como base para a gestão de imprevistos (do inglês *disruption management*).
- Aplicar o método proposto nesse trabalho no contexto real com suas restrições específicas de cada modal e localização.

REFERÊNCIAS

- Alfandari, L., Sadki, J., Plateau, A., and Nagih, A. (2013). Hybrid column generation for large-size covering integer programs: Application to transportation planning. *Computers & Operations Research*, **40**(8), 1938 – 1946.
- Archetti, C., Bianchessi, N., and Speranza, M.G. (2013). Optimal solutions for routing problems with profits. *Discrete Appl Math*, **161**(4-5), 547–557.
- Arenales, M., Armentano, V., Morabito, R., and Yanasse, H. (2011). *Pesquisa Operacional*. Elsevier Brasil.
- Ball, M., Bodin, L., and Dial, R. (1983). A matching based heuristic for scheduling mass transit crews and vehicles. *Transportation Science*, **17**(1), 4–31.
- Beasley, J.E. and Christofides, N. (1989). An algorithm for the resource constrained shortest path problem. *Networks*, **19**(4), 379–394.
- Bellman, R. (1958). On a routing problem. *Quarterly of Applied Mathematics*, **46**, 87–90.
- Bertossi, A.A., Carraresi, P., and Gallo, G. (1987). On some matching problems arising in vehicle scheduling models. *Networks*, **17**(3), 271–281.
- Bertsekas, D. (1991). *Linear Network Optimization: Algorithms and Codes*. The MIT Press, Massachusetts.
- Bertsekas, D. (1993). A simple and fast label correcting algorithm for shortest paths. *Networks*, **23**, 703–709.

- Bianco, L., Mingozzi, A., and Ricciardelli, S. (1994). A set partitioning approach to the multiple depot vehicle scheduling problem. *Optimization Methods & Software*, **3**(1-3), 163–194.
- Bodin, L., Golden, B., Assad, A., and Ball, M. (1983). Routing and scheduling of vehicles and crews: the state of the art. *Computers and Operations Research*, **10**(2), 63–211.
- Bunte, S. and Kliwer, N. (2009). An overview on vehicle scheduling models. *Public Transport*, **1**(4), 299–317.
- Carpaneto, G., Dell’Amico, M., Fischetti, M., and Toth, P. (1989). A branch and bound algorithm for the multiple depot vehicle scheduling problem. *Networks*, **19**(5), 531–548.
- Ceder, A.A. (2011a). Optimal multi-vehicle type transit timetabling and vehicle scheduling. *Procedia - Social and Behavioral Sciences*, **20**(0), 19 – 30.
- Ceder, A.A. (2011b). Public-transport vehicle scheduling with multi vehicle type. *Transportation Research Part C: Emerging Technologies*, **19**(3), 485 – 497.
- Chen, C., Xi, L.F., Zhou, B.H., and Zhou, S.S. (2011). A multiple-criteria real-time scheduling approach for multiple-load carriers subject to LIFO loading constraints. *International Journal of Production Research*, **49**(16), 4787–4806.
- Cherkassky, B.V., Georgiadis, L., Goldberg, A.V., Tarjan, R.E., and Werneck, R.F. (2009). Shortest-path feasibility algorithms: An experimental evaluation. *ACM Journal of Experimental Algorithmics*, **14**(2), 269–271.
- Desaulniers, G., Desrosiers, J., and Solomon, M.M. (eds.) (2005). *Column generation*. GERAD 25th anniversary series. Springer, New York. GERAD : Groupe d’études et de recherche en analyse des décisions.
- Farahani, R.Z., Miandoabchi, E., Szeto, W., and Rashidi, H. (2013). A review of urban transportation network design problems. *European Journal of Operational Research*, **229**(2), 281 – 302.
- Floyd, R.W. (1962). Algorithm 97: Shortest path. *Commun ACM*, **5**(6).

- Forbes, M.A., Holt, J.N., and Watts, A.M. (1994). An exact algorithm for multiple depot bus scheduling. *European Journal of Operational Research*, **72**(1), 115–124.
- Ford, L.R. (1956). Network flow theory. *Technical report*, Rand Corporation.
- Freling, R., Huisman, D., and Wagelmans, A.P. (2001). Applying an integrated approach to vehicle and crew scheduling in practice. In S. Voß and J. Daduna (eds.) *Computer-aided transit scheduling. Lecture Notes in Economics and Mathematical Systems*, vol 505, Springer, Berlin, pages 73–90.
- Gintner, V., Kliewer, N., and Suhl, L. (2005). Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice. *OR Spectrum*, **27**, 507–523.
- Goel, A. (2009). Vehicle scheduling and routing with drivers' working hours. *Transportation Science*, **43**(1), 17–26.
- Haase, K., Desaulniers, G., and Desrosiers, J. (2001). Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transportation Science*, **35**(3), 286–303.
- Hadjar, A., Marcotte, O., and Soumis, F. (2006). A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. *Operations Research*, **54**(1), 130–149.
- Haghani, A., Banihashemi, M., and Chiang, K.H. (2003). A comparative analysis of bus transit vehicle scheduling models. *Transportation Research Part B-Methodological*, **37**(4), 301–322.
- Hane, C.A., Barnhart, C., Johnson, E.L., Marsten, R.E., Nemhauser, G.L., and Sigismondi, G. (1995). The fleet assignment problem: solving a large-scale integer program. *Math Program*, **70**(2), 211–232.
- Huisman, D., Freling, R., and Wagelmans, A.P.M. (2005). Multiple-depot integrated vehicle and crew scheduling. *Transportation Science*, **39**(4), 491–502.
- Huisman, D. and Wagelmans, A.P.M. (2006). A solution approach for the dynamic vehicle and crew scheduling. *European Journal of Operational Research*, **172**(2), 453–471.

Jin, J.G., Zhao, J., and Lee, D.H. (2013). A column generation based approach for the train network design optimization problem. *Transportation Research Part E: Logistics and Transportation Review*, **50**, 1 – 17.

Jonker, R. and Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, **38**(4), 325–340.

Jozefowicz, N., Mancel, C., and Mora-Camino, F. (2013). A heuristic approach based on shortest path problems for integrated flight, aircraft, and passenger rescheduling under disruptions. *Journal of the Operational Research Society*, **64**(3), 384–395.

Kliwer, N., Mellouli, T., and Suhl, L. (2006). A time-space network based exact optimization model for multi-depot bus scheduling. *European Journal of Operational Research*, **175**(3), 1616–1627.

Kliwer, N., Mellouli, T., and Suhl, L. (2002). A new solution model for multi-depot multi-vehicle-type vehicle scheduling in (sub) urban public transport. In *Proceedings of the 13th Mini-EURO Conference and the 9th meeting of the EURO working group on transportation, Politechnic of Bari*.

Laurent, B. and Hao, J.K. (2009). Iterated local search for the multiple depot vehicle scheduling problem. *Computers & Industrial Engineering*, **57**(1), 277–286.

Laurent, B. and Hao, J.K. (2008). Simultaneous vehicle and crew scheduling for extra urban transports. In N.T. Nguyen, L. Borzemski, A. Grzech, and M. Ali (eds.) *IEA/AIE*. Springer, volume 5027 of *Lecture Notes in Computer Science*, pages 466–475.

Li, J.Q., Mirchandani, P.B., and Borenstein, D. (2009). Real-time vehicle rerouting problems with time windows. *European Journal of Operational Research*, **194**, 711–727.

Löbel, A. (1998). Vehicle scheduling in public transit and Lagrangean pricing. *Management Science*, **44**(12), 1637–1649.

Lopes, W.P. (2013). *Desenvolvimento de uma Heurística para o Problema de Escalonamento de Veículos com Múltiplas Garagens*. Dissertação de mestrado, UFRGS, Brasil.

- Mehlhorn, K. and Ziegelmann, M. (2000). Resource constrained shortest paths. In M. Paterson (ed.) *Algorithms - ESA 2000*, Springer Berlin Heidelberg, volume 1879 of *Lecture Notes in Computer Science*, pages 326–337.
- Mesquita, M. and Paixão, J. (1992). Multiple depot vehicle scheduling problem: A new heuristic based on quasi-assignment algorithms. In M. Desrochers and J.M. Rousseau (eds.) *Computer-Aided Transit Scheduling*, Springer Berlin Heidelberg, volume 386 of *Lecture Notes in Economics and Mathematical Systems*, pages 167–180.
- Moore, E.F. (1959). The shortest path through a maze. In H.U. Press (ed.) *Proc. Internat. Sympos. Switching Theory 1957, Part II. Cambridge, Mass.* pages 285–292.
- Oughalime, A., Ismail, W.R., Liong, C.Y., and Ayob, M.B. (2009). Vehicle and driver scheduling modelling: A case study in ukm. In *Proceedings of the 2nd Conference on Data Mining and Optimization, DMO 2009, Universiti Kebangsaan Malaysia, 27-28 October 2009*. IEEE, pages 53–59.
- Oukil, A., Ben Amor, H., Desrosiers, J., and El Gueddari, H. (2007). Stabilized column generation for highly degenerate multiple-depot vehicle scheduling problems. *Computers and Operations Research*, **34**, 817–834.
- Pepin, A., Desaulniers, G., Hertz, A., and Huisman, D. (2009). A comparison of five heuristics for the multiple depot vehicle scheduling problem. *Journal of Scheduling*, **12**(1), 17–30.
- Powell, W.B. and Carvalho, T.A. (1998). Real-time optimization of containers and flatcars for intermodal operations. *Transportation Science*, **32**, 110–126.
- Ramos, J.A., Reis, L.P., and Pedrosa, D. (2011). Solving heterogeneous fleet multiple depot vehicle scheduling problem as an asymmetric traveling salesman problem. In L. Antunes and H.S. Pinto (eds.) *EPIA*. Springer, volume 7026 of *Lecture Notes in Computer Science*, pages 98–109.
- Ribeiro, C.C. and Soumis, F. (1994). A column generation approach to the multiple-depot vehicle scheduling problem. *Operations Research*, **42**(1), 41–52.

Rohde, L.R. (2008). *Desenvolvimento de heurística para solução do problema de escalonamento de veículos com múltiplas garagens*. Tese de doutorado, UFRGS, Brasil.

Sato, K. and Fukumura, N. (2012). Real-time freight locomotive rescheduling and uncovered train detection during disruption. *European Journal of Operational Research*, **221**(3), 636–648.

Silva, G.P. (2010). Programação de veículos com múltiplas garagens: Soluções de apoio ao processo de licitação de linhas urbanas. In *Transporte em Transformação XV*. Positiva, pages 135–152.

Steinzen, I., Gintner, V., Suhl, L., and Kliewer, N. (2010). A time-space network approach for the integrated vehicle- and crew-scheduling problem with multiple depots. *Transportation Science*, **44**(3), 367–382.

van den Heuvel, A.P.R. and van Kooten, M.E. (2008). Integrating timetabling and vehicle scheduling in public bus transportation. *Technical Report*.

Visentini, M.S., Borenstein, D., Araujo, O.C.B., and Guedes, P.C. (2013). A new implementation to the vehicle type scheduling problem with time windows for scheduled trips. In *Simpósio Brasileiro de Pesquisa Operacional - SBPO, 2013*. Natal- RN.

Yang, J., Jaillet, P., and Mahmassani, H.S. (2004). Real-time multivehicle truckload pickup and delivery problems. *Transportation Science*, **38**(2), 135–148.