



UMA NOVA IMPLEMENTAÇÃO PARA O PROBLEMA DE ESCALONAMENTO DE VEÍCULOS COM MÚLTIPLAS GARAGENS USANDO GERAÇÃO DE COLUNAS.

Pablo Cristini Guedes

Escola de Administração – Universidade Federal do Rio Grande do Sul
Rua Washington Luiz, 855 - CEP 90010-460 – Porto Alegre – RS – Brasil
pcguedes@ea.ufrgs.br

Denis Borenstein

Escola de Administração – Universidade Federal do Rio Grande do Sul
Rua Washington Luiz, 855 - CEP 90010-460 – Porto Alegre – RS – Brasil
denisb@ea.ufrgs.br

Sylvain Fournier

WPLEX Software Ltda.
Rod. SC 401 no. 8600 Corporate Park bloco 5 sala 101 – Florianópolis – SC – Brasil
sylvain@wplex.com.br

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. Embora vários métodos de solução matemáticos tenham sido desenvolvidos, grandes instâncias (envolvendo milhares de viagens e várias garagens) são ainda difíceis de resolver em um tempo razoável. O objetivo deste trabalho de pesquisa é apresentar um método de solução baseado em geração de colunas para a solução de grandes instâncias não-estruturadas do MDVSP. Grandes instâncias obtidas randomicamente com base em uma distribuição de demandas envolvendo milhares de viagens (entre 1.000 e 10.000) e dezenas de garagens (4-128) foram obtidas em tempos razoáveis.

PALAVRAS CHAVE. MDVSP, rede tempo-espaço, Geração de Colunas.

ABSTRACT

The multiple-depot vehicle-scheduling problem (MDVSP) is a classic logistics and transportation problem. The MDVSP is also a sub problem 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 solution methods have been developed in the literature, large instances (involving thousands of trips and several depots) are still difficult to solve in reasonable time. The objective of this research is to present a solution method based on column generation for solving large unstructured MDVSP. Extensive testing was carried out using randomly generated instances, based on demands distribution. Large instances, involving thousands of trips (between 1,000 and 10,000) and dozen (4-128) depots, are solved in reasonable times.

KEYWORDS. MDVSP. time-space network. Column Generation.

L&T - Logística e Transportes

1. Introdução

O problema de escalonamento de veículos (MDVSP, do inglês *Multi-Depot Vehicle Scheduling Problem*) 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 deve começar o seu dia de trabalho em uma das garagens e terminar na mesma. 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 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; Chen et al., 2011); (ii) *disruption management* (Sato and Fukumura, 2012; Jozefowicz et al., 2013) e (iii) soluções integradas de problemas tais como veículos e tripulação (Huisman et al., 2005; Goel, 2009; Steinzen et al., 2010). Desta 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.

Em Bunte e Kliewer (2009) é possível ver que diversos tipos de formulações matemáticas já 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; Gintner et al., 2005; Kliewer 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) e o uso de heurísticas (Kliewer 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 (Kliewer et al., 2002; Gintner et al., 2005; Kliewer 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 e não-estruturadas na literatura obtidas computacionalmente são de 1500 viagens e 8 garagens em (Pepin et al., 2009).

2. Revisão da Literatura

A Tabela 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 1 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 et al. (1987)	Conexão	3	50	Relaxamento Lagrangeano	Prova que o problema é NP-Difícil
1989	Carpaneto et al. (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	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 et al. (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	Soluciona a maior instância estruturada vista na literatura, porém de forma não ótima
2002	Kliwer et al. (2002)	TSN	3	3776	Modelagem TSN e reduções na rede	Propõe a rede tempo-espaço
2008	Pepin et al. (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 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. Já Carpaneto et al. (1989) além de apresentarem uma formulação de transporte para a modelagem *single-commodity* com uma restrição de sub-ciclos, onde são representados as viagens e veículos como nós os autores são os primeiros a apresentar uma solução ótima para o MDVSP. Outra contribuição importante foi a definição de uma rotina para geração de instâncias aleatórias.

Forbes et al. (1994) propõem outra abordagem (*multi-commodity*), eles utilizam um relaxamento lagrangeano na execução de um algoritmo *dual simplex*. Forbes et al. (1994) também observaram 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. Os autores 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 na seção 4.

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), o autor 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 pro-

blema, 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 neste trabalho) utilizaram até 16 horas de processamento computacional em uma máquina dedicada.

[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 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](#)).

3. Modelagem

Para atingir o objetivo proposto, o problema descrito na introdução foi modelado com base na formulação da rede apresentada em [Guedes e Borenstein \(2013\)](#). 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.

3.1. Formulação do MDVSP

O MDVSP pode ser definido através de uma rede de conexão, como se segue, seja A^s 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 $i \in A^s$ 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:

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

s.t.

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

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

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

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

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).

4. Método de Solução

Para a solução do problema formulado em 3.1 foi utilizada a rede tempo-espço e o método de geração de colunas truncado modificado descrito em 4.2. 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 sub-seção 4.1). 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 em relação à rede conexão em trabalhos futuros, uma vez que agora é possível a partir de uma chegar na outra.

4.1. 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 da equação 4.1) representativa de uma rede de

conexão com n viagens e m garagens. Na rede tempo-espaço as viagens são representadas por arcos e os nós são duplas ordenadas do tipo (s, t) , onde $s \in K \cup S$ e $t \in T$ sendo K o conjunto de garagens, S o conjunto de terminais (representados por linhas na rede tempo-espaço) e T o tempo. Em suma, os arcos são pontos no tempo-espaço. Desta forma como na rede de conexão as viagens são nós e os arcos são transições entre as viagens, ao efetuar a conversão de tempo-espaço para conexão iremos transformar os arcos de serviço em nós da conexão, desprezando sua localização e tempo.

Claramente na transformação há perda significativa de informação, fato este que impede a transformação na ordem inversa. A única preocupação da rede de conexão é que exista arcos somente entre as viagens compatíveis (isto é, que o tempo de início da seguinte seja superior ao tempo de fim da anterior mais o tempo de deslocamento), mas a rede tempo-espaço naturalmente mantém essa condição satisfeita. 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 (arco de serviço a)

Gere um novo nó $i \in V$ onde i representará um nó na rede de conexão respectiva.

Para cada (arco de espera ou arco de *deadheading* a)

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-espaço, respectivamente.

Para cada (arco *pull-out* ou *pull-in* a)

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-espaço, respectivamente.

Para cada ($k \in K$)

Gere a capacidade da garagem k .

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

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)

4.2. Geração de Colunas Truncada Modificada

Para a resolução via método de geração de colunas, a formulação de partição de conjuntos 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:

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

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.

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 e, 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.

5. Experimentação Computacional

Esta seção 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-espço e de geração de colunas foram desenvolvidos, sem o auxílio de *frameworks*, pelo autor.

5.1. 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 1.

Inicialmente é gerado um número aleatório 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

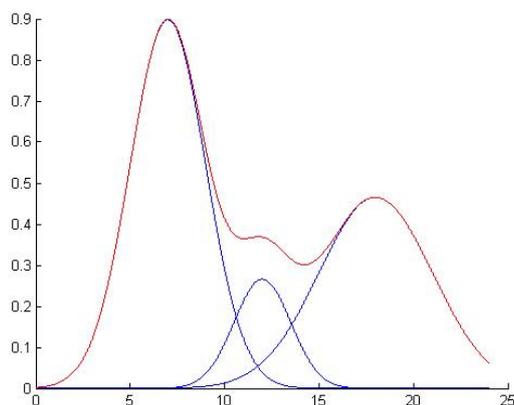


Figura 1. Distribuição de Demandas

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. Para melhor compreender-lo assumamos 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 5.1.

$$Q = \omega(0, Q_{max}) * Freq(r) + Q_{min} \quad (5.1)$$

Sendo:

- i) Q é a demanda calculada.
- ii) Q_{max} , a demanda máxima.
- iii) Q_{min} , a demanda mínima.
- iv) ω é um número randômico entre 0 e Q_{max} .
- v) r é um número randômico entre 0 e 23 (representa a hora de início da viagem).
- vi) $Freq(r)$ é o resultado da função frequência aplicado em r .

Na tabela 3 o resultado do método de geração de colunas aplicado ao MDVSP utilizando a rede tempo-espaco e as reduções descritas. Foram geradas 5 versões distintas para cada tamanho de instâncias, a tabela apresenta as médias dos tempos, valores da função objetivo e do número de veículos.

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	32	7.345	17472500	1526
8000	8	24.485	33013400	2772
8000	16	30.257	29598900	2569
10000	4	32.578	45808300	3652

Tabela 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.

Definindo como *lower bound* de desempenho 1h de processamento, 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. Como trata-se de planejamento, um *gap* máximo (estimado) encontrado de 3,5% demonstrou ser insignificante sendo que em diversos casos foi encontrada a solução ótima.

6. 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 artigo desenvolveu um método de solução, baseado na integração da rede tempo-espaco e da geração de colunas para a solução de grandes instâncias não estruturadas do *Multiple Depot Vehicle Scheduling Problem*, envolvendo dezenas de garagens e milhares de viagens. A resolução de instâncias de tamanhos maiores aos encontrados atualmente na literatura para o MDVSP indicam a capacidade de solucionar grandes problemas reais. Pelos resultados obtidos é concluso que o objetivo do trabalho foi plenamente atendido, buscando resolver o problema MDVSP 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.

Como trabalhos futuros sugere-se: (i) 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; (ii) Utilizar o estudo aqui realizado no contexto de transporte de cargas expressas; (iii) Aplicar o método no contexto aéreo e, mesmo que com adaptações necessárias, como base para a gestão de imprevistos (do inglês *disruption management*); e (iv) Aplicar o método proposto nesse trabalho no contexto real com suas restrições específicas de cada modal e localização.

Referências

- Bertossi, A. A., Carraresi, P., and Gallo, G.** (1987), On some matching problems arising in vehicle scheduling models, *Networks*, 17(3):271–281.
- 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 Klierer, 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.
- 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.
- 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.
- Gintner, V., Klierer, 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.
- 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.
- Huisman, D., Freling, R., and Wagelmans, A. P. M.** (2005), Multiple-depot integrated vehicle and crew scheduling, *Transportation Science*, 39(4):491–502.
- 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.
- Klierer, 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.
- Klierer, 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.

- 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.
- Mesquita, M. and Paixão, J.** (1992), Multiple depot vehicle scheduling problem: A new heuristic based on quasi-assignment algorithms, In Desrochers, M. and Rousseau, J.-M., editors, *Computer-Aided Transit Scheduling*, volume 386 of *Lecture Notes in Economics and Mathematical Systems*, pages 167–180, Springer Berlin Heidelberg.
- 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.
- 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*, pages 135–152, Positiva.
- 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.
- Yang, J., Jaillet, P., and Mahmassani, H. S.** (2004), Real-time multivehicle truckload pickup and delivery problems, *Transportation Science*, 38(2):135–148.