PRODUÇÃO ONLINE
REVISTA CIENTÍFICA ELETRÔNICA DE ENGENHARIA DE PRODUÇÃO
ABEPRO
ISSN 1676-1901

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

A TABU-SEARCH HEURISTIC FOR SOLVING THE MULTI-DEPOT VEHICLE SCHEDULING PROBLEM

Gilmar D'Agostini Oliveira Casalinho* E-mail: gdocasalinho@ea.ufrgs.br
Gabriel Machado Braido** E-mail: gabrielb@univates.br
Denis Borenstein* E-mail: denisb@ea.ufrgs.br
*Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, RS
**Centro Universitário UNIVATES, Lajeado, RS

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

Palavras-Chave: Pesquisa Operacional. Heurística. Busca-tabu. Escalonamento de Veículos. MDVSP.

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

Keywords: Operational Research. Heuristics. Tabu-search. Vehicle Scheduling. MDVSP.

1 INTRODUÇÃO

A Pesquisa Operacional (PO) vem sendo utilizada de forma bastante expressiva a fim de obter uma maior eficiência nas operações quando problemas logísticos são encontrados. Às questões relativas ao sequenciamento de produção, instalações de novas plantas, planejamento de distribuição, roteamento de veículos, entre outras, utilizam-se técnicas de PO com o intuito de proporcionar não apenas uma solução eficiente, mas, em muitos casos, uma solução ótima para essa variedade de problemas (BOFFEY, 1984).

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

Dentre os vários problemas relacionados à designação de veículos em um sistema logístico, o de Escalonamento de Veículos com Múltiplas Garagens, MDVSP (*Multi-Depot Vehicle Scheduling Problem*), vem sendo abordado em variadas pesquisas. Este tipo de problema pressupõe a existência de diversas garagens que interferem no planejamento das sequências com as quais as viagens devem ser executadas, tal qual em sistemas reais de frotas de aeronaves ou caminhões de recolhimento de lixo urbano (PEPIN *et al*., 2008) e, frequentemente, sua implementação é inviável em problemas do mundo real. Para poder levar em consideração instâncias cada vez maiores ou mais próximas das reais, várias abordagens heurísticas estão sendo desenvolvidas (BALDACCI; TOTH; VIGO, 2010).

Estas heurísticas variam desde os métodos de busca local aos métodos baseados em programação matemática e técnicas de decomposição, incluindo as meta-heurísticas. Desta forma, muitos dos modelos que representam esse problema são apresentados de modo genérico, o que dificulta sua implementação em um contexto real (BERTOSSI; CARRARESI; GALLO*,* 1987).

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

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

Em adição a isto, a principal motivação para a realização deste trabalho surgiu a partir de sugestões e limitações elencadas no estudo de Rohde (2008), o qual utilizou o algoritmo *branch-and-bound* em uma das etapas da heurística apresentada para resolver o MDVSP, fazendo com que restrições acrescidas ao modelo original começassem a impactar demasiadamente no tempo de solução.

Abordagens heurísticas como, principalmente, a geração de colunas e a busca-tabu, configuram-se como mais apropriadas à resolução desse tipo de problema devido a suas características de memória (ROHDE, 2008). Pepin *et al.*(2008) complementam dizendo que a meta-heurística de busca-tabu é capaz de encontrar soluções muito próximas às ótimas em um tempo computacional muito pequeno.

Dos dois métodos discutidos acima, a resolução via meta-heurística de busca-tabu é um método de solução mais apropriado do que o de geração de colunas quando se deseja encontrar uma solução de boa qualidade em um tempo computacional melhor (PEPIN et al., 2008). Esta meta-heurística é utilizada em vários trabalhos na área de otimização e devido à sua grande adaptação a problemas de roteamento e escalonamento de veículos justifica-se sua escolha para o desenvolvimento deste trabalho (GLOVER, 1989).

Na literatura existente, apenas no estudo de Pepin et al. (2008) foram encontrados desenvolvimentos utilizando a meta-heurística de busca-tabu para a resolução do MDVSP[1]. Os próprios autores reconhecem e desconhecem a inexistência de outros trabalhos abordando este tipo de resolução.

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

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

Portanto, o objetivo deste trabalho é **propor uma heurística para solucionar o MDVSP utilizando busca-tabu** a fim de que se encontrem soluções capazes de representar a realidade de um MDVSP, não apenas apresentando uma redução dos custos e do tamanho da frota, mas também sendo capaz de propiciar uma melhor qualidade de atendimento ao cliente e, por consequência, uma habilidade maior de se adequar às contingências.

A seguir, apresenta-se uma contextualização dos problemas de designação de veículos em um sistema logístico. Após, na seção 3, apresenta-se o caso específico do MDVSP, seguido de algumas abordagens de busca-tabu utilizadas na literatura (seção 4) e do método adotado para este estudo (seção 5). Os experimentos computacionais e as análises dos dados são descritos na seção 6 e, após, finaliza-se com as conclusões do estudo.

2 PROBLEMAS DE DESIGNAÇÃO DE VEÍCULOS EM UM SISTEMA LOGÍSTICO

O atual interesse na logística de transportes, a contínua busca pela redução dos custos de produção e os incrementos cada vez mais frequentes na tecnologia

---

[1] Através de busca das seguintes palavras-chave nas principais bases de dados internacionais: *"tabu-search", "heuristics", "metaheuristics", "MDVSP" "scheduling problem"*.

da computação faz com que muitos dos pesquisadores operacionais foquem seus interesses nos problemas de roteamento e escalonamento de veículos (GOLDBARG; LUNA, 2005).

Esses problemas geralmente envolvem a designação de veículos a viagens, de forma que os custos de designação e da rota correspondente sejam minimizados. Os aspectos relacionados ao tempo de duração dessas viagens tornam-se muito importantes em contextos industriais, de serviços e das empresas de transporte, que tentam não apenas diminuir ou acabar com alguns de seus custos logísticos, mas também competir com serviços diferenciados (CEDER, 2011, DESROSIERS et al., 1995).

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

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

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

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

O Problema de Escalonamento de Veículos para uma única garagem (SDVSP, do inglês Single-Depot Vehicle Scheduling Problem) tornou-se uma área de pesquisa amplamente estudada nos últimos 40 anos (BUNTE; KLIEWER, 2009), porém devido às configurações atuais das redes logísticas, a abordagem de estudo utilizando múltiplas garagens tem sido mais utilizada.

O problema de escalonamento de veículos com múltiplas garagens (MDVSP) é um problema bastante conhecido e que possui aplicações em vários campos, como no transporte público e privado de passageiros e em sistemas de entrega e coleta de mercadorias através de caminhões e aeronaves (PEPIN et al., 2008; WANG, 2013).

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

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

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

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

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

Em segundo lugar, vários softwares utilizados por empresas de transporte público e de entregas são capazes de resolver o MDVSP. Porém, na maioria das vezes, heurísticas são usadas para encontrar soluções mais rapidamente (HADJAR; MARCOTTE; SOUMIS, 2006).

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

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

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

A seguir apresenta-se a formulação para o MDVSP de acordo com o modelo de fluxo de rede proposto por Ribeiro e Soumis (1994).

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

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

$$\text{Minimizar} \sum_{k \in K} \sum_{(i,j) \in A^k} c_{ij} X_{ij}^k \qquad (1)$$

Sujeito a:

$$\sum_{k \in K} \sum_{j:(i,j) \in A^k} X_{ij}^k = 1, \qquad \forall i \in T \tag{2}$$

$$\sum_{j:(o(k),j) \in A^k} X_{o(k),j}^k \leq v_k \quad \forall k \in K \tag{3}$$

$$\sum_{j:(j,i) \in A^k} X_{ji}^k - \sum_{j:(i,j) \in A^k} X_{ij}^k = 0 \quad \forall i \in V^k \setminus \{o(k), d(k)\}, k \in K \tag{4}$$

$$X_{ij}^k \in \{0,1\}, \forall (i,j) \in A^k, k \in K. \tag{5}$$

Neste modelo, a função (1) minimiza o total dos custos, enquanto a restrição (2) assegura que cada viagem seja executada apenas uma vez por veículo. Por sua vez, a restrição (3) limita o número de veículos que pode ser usado por garagem e a restrição (4) é a de fluxo conservativo, a qual define estruturas de caminhos múltiplos para cada garagem. Por fim, os requisitos para as variáveis binárias são encontrados em (5).

4 BUSCA-TABU (BT)

A principal ideia desta meta-heurística consiste em avaliar a vizinhança da solução atual a cada iteração, sendo que a solução inicial pode ser construída aleatoriamente ou através de outro método estruturado. Uma heurística de busca local convencional pararia quando achasse uma solução atual melhor do que suas soluções vizinhas, no entanto, a busca-tabu aceita soluções de não-melhoria como estratégia para ir além desse ótimo local (PEPIN et al., 2008). Alé, disso, Krajewska e Kopfer (2009) consideram que uma vantagem da Busca Tabu é que boas soluções podem ser obtidas em um tempo consideravelmente reduzido.

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

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

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

5 MÉTODO DE PESQUISA

Esta pesquisa se apoia nos preceitos metodológicos da pesquisa operacional propostos por Arenales et al. (2007), Miguel et al. (2012), Goldbarg e Luna (2005) e Andrade (2009), que a consideram, por si só, um método de pesquisa que agrega, em sua teoria, quatro ciências fundamentais para o processo decisório: economia, matemática, estatística e computação.

Como parte da pesquisa desenvolvida, o referencial teórico apresentado também serve para delimitar as fronteiras do que será investigado, proporcionar o suporte técnico para a pesquisa (fundamentos) e também explicitar o grau de evolução (estado-da-arte) sobre o tema investigado. (MIGUEL et al., 2012). Assim, a partir deste momento, pode-se desempenhar as funções aplicadas a que esta pesquisa se propõe.

Adaptando-se as instruções metodológicas dos autores acima, o método aplicado nesta pesquisa divide-se em três etapas: a) desenvolvimento de uma abordagem heurística para resolução do MDVSP; b) implementação computacional do modelo; e, c) avaliação da solução encontrada.

5.1 Desenvolvimento de uma abordagem heurística

A primeira etapa da pesquisa visou desenvolver uma abordagem heurística a fim de solucionar o MDVSP para problemas reais ou para instâncias que bem os representem. Para encontrar a solução do problema, desenvolveu-se um método baseado na meta-heurística de busca-tabu, a qual apresenta como principal objetivo melhorar uma solução corrente por meio da execução de movimentos dentro de uma vizinhança (CHIANG *e*t al., 2009).

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

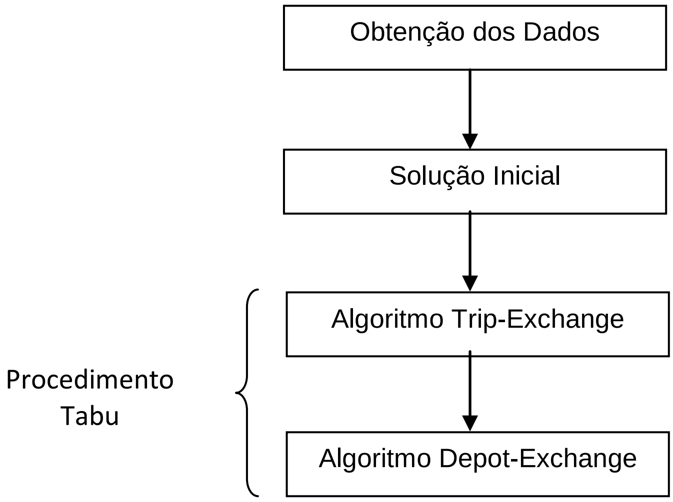
Uma meta-heurística considerada robusta não deve apresentar resultados ruins para dada instância. Além disso, o algoritmo precisa ser capaz de produzir boas soluções a qualquer momento que seja aplicado a alguma instância. Isto é de extrema importância uma vez que muitas heurísticas não são determinísticas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

Utilizando apenas os melhores resultados de uma heurística não-determinística, como geralmente é feito na literatura, pode-se fazer com que se crie uma falsa impressão do seu real desempenho. Portanto, a média dos resultados obtidos através da múltipla execução de cada um dos problemas é uma importante base para a comparação de métodos não-determinísticos. Mais do que isso, também é importante explicitar o caso com o pior desempenho apresentado (*worst-case performance*) (CORDEAU et al., 2002).

Para reduzir a complexidade da busca, alguns autores propõem estratégias especiais a fim de limitar a vizinhança. Por exemplo, Garcia *et al.* (1994), somente permitem movimentos envolvendo arcos que são próximos em distância. Taillard et al*.* (1997) decompõem a solução em subconjuntos disjuntos de rotas, utilizando o ângulo polar associado com o centro de gravidade de cada rota. A busca-tabu é aplicada, então, para cada subconjunto de forma separada. Uma solução completa é reconstruída adicionando-se as novas rotas encontradas pela busca-tabu (BRÄYSY, DULLAERT, GENDREAU; 2004).

Neste trabalho, após a obtenção dos dados das instâncias, uma solução inicial é gerada conforme o algoritmo ilustrado no Quadro 1. Uma vez esta solução inicial sendo encontrada, dois algoritmos distintos, o denominado *Trip-Exchange* (Quadro 2) e o outro denominado *Depot-Exchange* (Quadro 3) são implementados orientados por uma lista-tabu. O resumo do método heurístico desenvolvido pode ser visualizado na Fig. 1:

Figura 1 - Método Heurístico Desenvolvido



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

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

As viagens entre os nós são selecionadas levando-se em consideração a melhor garagem usando o custo das viagens de *pull-out*. Assim, um nó é selecionado de forma randômica usando o índice de *pull-out* do depósito selecionado. A seguir, adiciona-se um vizinho do nó anteriormente selecionado enquanto o nó não esteja no índice de *pull-in*. Caso o nó esteja no índice de *pull-in* ou não haja mais vizinhos, selecionam-se vizinhos para novas viagens de *pull-in* ou, se todos os nós estiverem alocados, é verificado se todas as viagens são possíveis. Caso não sejam tenta-se inseri-las nas viagens possíveis.

O procedimento heurístico para geração da solução inicial é realizado conforme o Quadro 1.

Quadro 1 – Procedimento heurístico para geração da solução inicial

Passo 1: Obtendo vizinhos. - Verifica-se a possibilidade da viagem entre todos os nós; - Aloca-se em um índice somente as viagens possíveis, e organiza-as em forma crescente conforme o custo; Passo 2: Obtendo *pull-out* e *pull-in.* - Verifica-se a possibilidade de cada depósito atender cada nó; - Aloca-os em dois índices organizando-os em forma crescente conforme o custo. Passo 3: Obtendo as viagens. - Seleciona-se o melhor depósito usando o custo de *pull-out;* - Seleciona-se um nó de forma randômica usando o índice de *pull-out*, do depósito selecionado; - Continua-se adicionando um vizinho do nó anteriormente selecionado enquanto o nó não esteja no índice de *pull-in;* - Caso o nó esteja no índice de *pull-in* ou não haja mais vizinhos, retorna-se ao passo 2 ou se todos nós estão alocados passa-se para o seguinte; - Verifica-se a possibilidade de execução de todas as viagens, caso não seja possível, tenta-se inseri-las nas possíveis.

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

O algoritmo de *trip-exchange* seleciona uma sequência de viagens e um nó pertencente a ela de forma randômica. Caso seja possível uma troca com redução do custo, esta é realizada. Se a viagem estiver presente na lista-tabu, volta-se ao passo anterior e, se não, verifica-se com todas as viagens e todos os nós se a troca é possível e se há redução do custo. Caso haja a troca, retorna-se ao primeiro passo 1.500 vezes (número escolhido devido ao máximo de viagens que pode ser configurado no sistema). Executa-se, então, o algoritmo até não encontrar mais trocas possíveis.

O procedimento heurístico de *Trip-Exchange* é apresentado no Quadro 2:

Quadro 2 – Procedimento heurístico para o *Trip-Exchange*

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

Já o algoritmo de *depot-exchange* inicialmente seleciona duas viagens de forma randômica e faz a consulta na lista-tabu a fim de garantir que a viagem selecionada não esteja contida nela. Zäpfel,Braune e Bögl (2010) consideram a lista tabu, representada por uma memória de curto prazo, como elemento central da BT. Caso a viagem esteja na lista-tabu, volta-se ao passo anterior, conforme o Quadro 3. Senão, verifica-se a possibilidade de troca de garagens entre as viagens e se há redução do custo. Caso haja, efetua-se a troca e retorna-se ao primeiro passo 500 vezes. Executa-se o algoritmo até não se encontrar mais trocas possíveis.

Quadro 3 – Procedimento heurístico para o *Depot-Exchange*

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

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

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

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

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

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

A busca-tabu pode ser notada da seguinte forma: para uma solução $s \in S$, diz-se que $N(s)$ denota a vizinhança de s através de procedimentos de troca local, chamados movimentos. As técnicas de busca local visitam uma sequência $s0, \ldots, st$ de soluções, onde $s0$ é a solução inicial e $si+1 \in N(si)(i = 1, ...t\text{-}1)$. Maiores esclarecimentos do algoritmo de busca-tabu, bem como detalhes da descrição do método e de como foi concebido pode ser encontrado em Glover e Laguna (1997).

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

5.2 Implementação computacional

Quanto às instâncias utilizadas neste trabalho, foram utilizadas as geradas aleatoriamente por Carpaneto *et al.* (1989) e posteriormente utilizadas por diversos outros autores como Huisman, Freling e Wagelmans (2004). A escolha das instâncias foi baseada justamente na possibilidade de comparação com outros trabalhos que já apresentaram soluções para o MDVSP a partir do mesmo conjunto de dados.

O programa foi desenvolvido na linguagem C ANSI sob plataforma Linux através do compilador GCC 4.6.3 e MAKE 3.81. Todos os testes foram rodados em um *laptop* com Linux FastGene 3.2.0-24-generic #37-Ubuntu, equipado com processador Intel® Core™2 Duo com 2.66 GHz e 4 Gb de memória.

Os dados foram obtidos através dos arquivos *.inp contidos nas instâncias de Carpaneto et al. (1989) disponibilizadas em meio eletrônico, de domínio público (http://people.few.eur.nl/huisman/instances.htm) e organizados em uma estrutura de memória.

5.3 Avaliação da solução

A terceira e última etapa da pesquisa correspondeu à avaliação da solução encontrada para o problema, onde o modelo implementado foi rodado para cenários variados com o objetivo de mostrar a capacidade de solucionar problemas do tipo MDVSP.

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

O tempo que uma heurística leva para produzir boas soluções pode ser crucial na escolha entre diferentes técnicas, da mesma forma que a qualidade da solução final, mensurada pelo resultado da função objetivo, também é importante (BRAYSY; DULLAERT; GENDREAU, 2004). Assim, o procedimento heurístico proposto será avaliado levando-se em consideração estes e outros critérios na próxima seção.

6 EXPERIMENTOS COMPUTACIONAIS E ANÁLISES

As instâncias utilizadas neste trabalho representam problemas diários do tipo MDVSP, onde a tarefa (ou viagem) de início e a localização final são selecionadas a partir de um conjunto restrito de pontos aleatórios em um plano.

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

Porém, o número total de veículos não se configura como uma restrição nestas instâncias. Nestas, o objetivo consiste em minimizar, primeiramente, o número de veículos utilizados e, após, os custos totais de operação. Para que isso possa ocorrer, um alto custo fixo (10.000 unidades monetárias - u.m.) é aplicado para cada veículo utilizado, a fim de priorizar fortemente o primeiro objetivo.

Em uma solução ótima que utilize um número mínimo de veículos, cada um realiza, em média, de quatro a cinco viagens. De acordo com Pepin *et al.* (2008), a maior destas instâncias que pode ser resolvida na otimalidade corresponde a 800 viagens (T) e 6 garagens (K). Nos testes utilizados neste trabalho, assim como no estudo supracitado, foram utilizadas instâncias onde $|T| \in \{500, 1.000, 1.500\}$ *e* $|K| \in \{4, 8\}$. Todos os resultados correspondem a cinco diferentes instâncias, do mesmo tamanho, geradas aleatoriamente.

6.1 Configuração dos Experimentos

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

O primeiro cenário apresentado corresponde a cinco instâncias (s0; s1; s2; s3; s4) do MDVSP as quais contém em seus sistemas quatro garagens disponíveis e quinhentas viagens a serem executadas. A Tabela 1 apresenta a configuração deste cenário e dos outros cinco. Na última coluna apresenta-se o número máximo de veículos disponíveis para cada garagem (respectivamente de *k0* até *k3* para instâncias com quatro garagens; e *k0* até *k7* para instâncias com 8 garagens) em cada instância.

Tabela 1 – Configuração do Cenários

Cenário	Instância	Nº Garagens	Nº Viagens	Nº Máximo de Veículos em cada Garagem
1	0	4	500	62/59/56/56
	1	4	500	57/59/50/62
	2	4	500	49/54/55/59
	3	4	500	57/61/60/52
	4	4	500	54/63/45/54
2	0	4	1000	102/118/100/127
	1	4	1000	96/112/120/110
	2	4	1000	106/96/113/92
	3	4	1000	96/112/106/115
	4	4	1000	127/106/99/97
3	0	4	1500	129/180/188/178
	1	4	1500	175/154/147/180
	2	4	1500	159/180/166/171
	3	4	1500	144/171/180/153
	4	4	1500	129/171/141/152
4	0	8	500	24/28/33/25/31/32/33/30
	1	8	500	27/30/27/30/24/25/26/33
	2	8	500	30/25/31/26/25/27/30/33
	3	8	500	34/30/33/33/26/27/32/24
	4	8	500	21/28/28/29/26/31/26/32
5	0	8	1000	61/58/52/47/51/51/54/49
	1	8	1000	56/50/51/64/52/61/61/45
	2	8	1000	46/56/57/62/59/57/57/45
	3	8	1000	56/62/61/48/63/51/51/45
	4	8	1000	49/48/49/60/54/49/49/46
6	0	8	1500	79/71/76/74/69/81/84/80
	1	8	1500	74/66/76/87/82/71/83/96
	2	8	1500	82/67/87/79/68/78/71/96
	3	8	1500	82/67/87/79/68/78/71/96
	4	8	1500	68/68/78/92/72/90/76/96

6.2 Resultados e análises

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

Desta forma, é possível que se faça uma comparação entre os resultados iniciais e os finais (após a aplicação dos outros algoritmos), aqui denominados de "Solução Final".

6.2.1 Comparação entre a solução inicial e a solução final encontrada

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

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

O resumo das soluções iniciais e finais encontradas neste trabalho é apresentado de acordo com o que mostra a Tabela 2. Nela são elencadas as soluções iniciais e finais para cada uma das cinco instâncias dos seis cenários analisados e, na última coluna, a melhoria percentual atingida de uma fase para a outra. Estes resultados já foram discutidos em seções anteriores, mas pode-se perceber uma melhoria em todas as instâncias testadas.

Tabela 2 - Resumo das soluções inicial e final para os seis Cenários

Cenário	Instância	Solução Inicial	Solução Final	Melhoria (%)
1	0	1.358.431	1.316.453	3,09
	1	1.437.743	1.396.346	2,88
	2	1.440.531	1.418.062	1,56
	3	1.270.282	1.249.628	1,63
	4	1.221.037	1.211.123	0,81
2	0	2.769.424	2.704.635	2,34
	1	2.698.002	2.651.215	1,73
	2	2.538.263	2.491.529	1,84
	3	2.567.810	2.519.289	1,89
	4	2.796.170	2.742.430	1,92
3	0	3.693.191	3.662.253	0,84
	1	3.910.828	3.826.654	2,15
	2	3.777.073	3.720.400	1,50
	3	3.734.302	3.671.426	1,68
	4	4.019.305	3.959.818	1,48
4	0	1.367.528	1.322.293	3,31
	1	1.411.747	1.380.047	2,25
	2	1.315.647	1.269.947	3,47
	3	1.340.134	1.316.075	1,80
	4	1.416.222	1.361.857	3,84
5	0	2.469.038	2.395.948	2,96
	1	2.597.432	2.525.733	2,76
	2	2.769.520	2.679.415	3,25
	3	2.671.393	2.608.787	2,34
	4	2.559.445	2.473.958	3,34
6	0	3.916.768	3.784.325	3,38
	1	3.969.693	3.818.718	3,80
	2	4.017.578	3.920.566	2,41
	3	4.012.240	3.911.726	2,51
	4	3.962.513	3.845.059	2,96

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

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

Desta forma, apesar dos resultados da heurística proposta neste trabalho apresentarem-se um pouco superiores aos mostrados por Pepin *et al.* (2008), o tempo computacional atingido foi significativamente menor por ter-se adotado um método de geração de soluções iniciais bastante adequado e pela própria característica do método de busca-tabu de apresentar tempos computacionais menores.

É importante salientar, também, que a diferença expressiva no tempo computacional apresentado por este estudo e naquele de Pepin *et al.* (2008), deve-se ao fato de que estes autores pré-definiram o tempo de execução da heurística de busca-tabu, como pode ser visualizado nas tabelas a seguir (por esta razão, estes tempos apresentam-se de forma destacada na Tabela 15).

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

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

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

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

A fim de tornar mais clara a leitura e comparação dos resultados obtidos por esta pesquisa com aqueles obtidos por Pepin et al. (2008), apresenta-se a Tabela 3 que traz um resumo de alguns resultados obtidos por ambos os trabalhos.

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

Tabela 3 - Resumo dos Custos Finais e Nº de Veículos Utilizados comparados aos Melhores Valores Encontrados para Pepin et al. (2008)

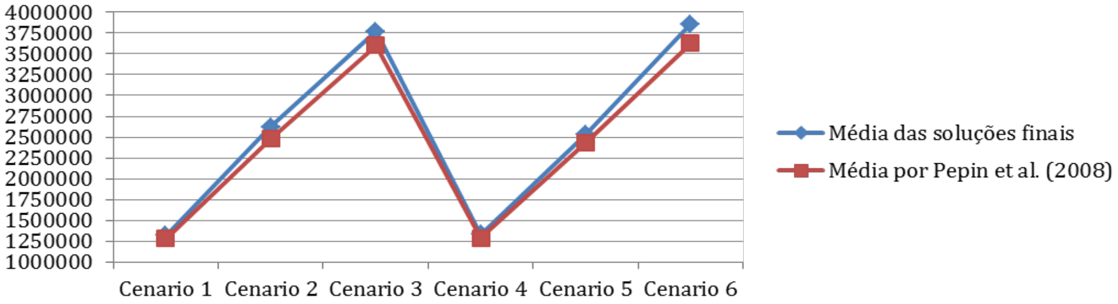
Cenário	Instância	Custo Final	Custo por Pepin et al. *(2008)*	Gap (% de melhoria)	Nº de Veículos Utilizados	Nº de Veículos Utilizados por Pepin et al. (2008)	Diferença no Número de Veículos
1	0	1.316.453	1.289.114	-2,12	117	123	-6
	1	1.396.346	1.241.618	-12,46	132	118	14
	2	1.418.062	1.283.811	-10,46	130	123	7
	3	1.249.628	1.258.634	0,72	111	120	-9
	4	1.211.123	1.317.077	8,04	109	126	-17
2	0	2.704.635	2.516.247	-7,49	246	241	5
	1	2.651.215	2.413.393	-9,85	242	229	13
	2	2.491.529	2.452.905	-1,57	223	233	-10
	3	2.519.289	2.490.812	-1,14	235	237	-2
	4	2.742.430	2.519.191	-8,86	242	238	4
3	0	3.662.253	3.830.912	4,40	344	368	-24
	1	3.826.654	3.559.576	-7,50	349	338	11
	2	3.720.400	3.649.759	-1,94	343	350	-7
	3	3.671.426	3.406.815	-7,77	340	326	14
	4	3.959.818	3.567.122	-11,01	369	343	26
4	0	1.322.293	1.292.411	-2,31	121	124	-3
	1	1.380.047	1.276.419	-8,12	125	123	2
	2	1.269.947	1.304.251	2,63	120	126	-6
	3	1.316.075	1.277.388	-3,03	117	123	-6
	4	1.361.857	1.276.010	-6,73	122	123	-1
5	0	2.395.948	2.422.112	1,08	225	232	-7
	1	2.525.733	2.524.293	-0,06	232	244	-12
	2	2.679.415	2.256.313	-18,75	244	247	-3
	3	2.608.787	2.478.393	-5,26	244	237	7
	4	2.473.958	2.498.388	0,98	230	240	-10
6	0	3.784.325	3.500.160	-8,12	360	337	23
	1	3.818.718	3.802.650	-0,42	354	366	-12
	2	3.920.566	3.605.094	-8,75	360	349	11
	3	3.911.726	3.515.802	-11,26	359	338	21
	4	3.845.059	3.704.953	-3,78	363	360	3

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

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

Os resultados quanto ao valor da função objetivo ficam mais evidentes se analisarmos a Fig. 2, onde são evidenciadas as médias para cada uma das instâncias do MDVSP resolvidas tanto pela heurística deste trabalho como pelo método de Pepin et al. (2008).

Figura 2 – Médias das soluções encontradas



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

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

7 CONCLUSÕES

7.1 Síntese das conclusões

A dificuldade de se resolver um problema da classe NP-*hard* como o problema de escalonamento de veículos com múltiplas garagens (MDVSP) requer estratégias de solução baseadas em heurísticas para a maioria das instâncias de tamanho real. Recentemente, algumas meta-heurísticas estão sendo discutidas para a resolução deste tipo específico de problema (Pepin *et al.* 2008), e foi o que este trabalho se propôs a apresentar, utilizando o método de busca-tabu em função das indicações apresentadas na literatura.

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

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

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

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

Somando-se a esses pontos, outra grande vantagem no trabalho apresentado foi a possibilidade de fechamento de algumas garagens (ou depósitos) igualmente sem penalizar os custos da função objetivo e sem necessidade de relaxação de nenhuma das restrições (via algoritmo *depot-exchange*).

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

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

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

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

7.2 Limitações

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

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

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

7.3 Sugestões para trabalhos futuros

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

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

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

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

REFERÊNCIAS

ANDRADE, E L. **Introdução a Pesquisa Operacional**: métodos e modelos para a análise de decisão. 4 ed. Rio de Janeiro: LTC, 2009.

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

BALDACCI, A., TOTH, P., VIGO, D. Exact algorithms for routing problems under vehicle capacity constraints. **Ann. Oper. Res.** n. 175, p. 213-245.

BERTOSSI, A. A., CARRARESI, P.; GALLO, G. On some matching problems arising in vehicle scheduling models. **Networks** n.17, p. 271–281, 1987. http://dx.doi.org/10.1002/net.3230170303

BOFFEY, T.B. **Graph theory in operations research**. Hong Kong: Macmillan, 1984.

BRÄYSY, O.; GENDREAU, M. Vehicle routing problem with time windows, part ii: metaheuristics. **Transportation Science,** v. 39, n. 1, p. 119-139, 2005. http://dx.doi.org/10.1287/trsc.1030.0056

BRÄYSY, O.; DULLAERT, W.; GENDREAU, M. Evolutionary algorithms for the vehicle routing problem with time windows. **Journal of Heuristics**, v. 10, n. 6, p. 587-611, 2004. http://dx.doi.org/10.1007/s10732-005-5431-6

BUNTE, S., KLIEWER, N. An Overview on Vehicle Scheduling Models. **Journal of Public Transport.** v. 1, n. 4, p. 299-317, 2009. http://dx.doi.org/10.1007/s12469-010-0018-5

CARPANETO, G; DELL'AMICO, M; FISCHETTI, M; TOTH P. A branch and bound algorithm for the multiple depot vehicle scheduling problem. **Networks**, n. 19, p.531-548, 1989. http://dx.doi.org/10.1002/net.3230190505

CEDER, A. Public-transport Vehicle Scheduling with Multi Vehicle Type. **Transportation Research,** v. 19, n. 3, p. 485-497, 2011.

CORDEAU, J. F., GENDREAU, M., LAPORTE, G., POTVIN, J. Y.; SEMET, F. A guide to vehicle routing problem. **Journal of the Operational Research Society** , v.53, p. 512–522, 2002. http://dx.doi.org/10.1057/palgrave.jors.2601319

EKSIOGLU, B.; VURAL, A.; REISMAN, A. The Vehicle Routing Problem: a Taxonomic Review. **Computers and Industrial Engineering,** n. 57, p. 1472 – 1483, 2009. http://dx.doi.org/10.1016/j.cie.2009.05.009

GENDREAU, M., GUERTIN, F., POTVIN, J.-Y., TAILLARD, E. D. Parallel tabu search for real-time vehicle routing and dispatching, **Transportation Science** 33: 381–390, 1999. http://dx.doi.org/10.1287/trsc.33.4.381

GLOVER, F. Tabu search: Part i, **ORSA Journal of Computing** 1: 190–206, 1989. http://dx.doi.org/10.1287/ijoc.1.3.190

GOLDBARG, M. C. e LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. Rio de Janeiro: Campus, 2005.

HADJAR, A.; MARCOTTE, O.; SOUMIS, F. A Branch-and-cut algorithm for the multiple depot vehicle scheduling problem. **Operations Research,** v. 54, n. 1, p. 130-149, 2006. http://dx.doi.org/10.1287/opre.1050.0240

HAGHANI, A.. BANIHASHEMI, M.; CHIANG, K. A Comparative analysis of bus transit vehicle scheduling models. **Transportation Research Part B,** v. 37, n. 4, p. 301-322, 2003. http://dx.doi.org/10.1016/S0191-2615(02)00007-3

HUISMAN, D.; FRELING, R.; WAGELMANS, A. A Robust solution approach to the dynamic vehicle scheduling problem. **Transportation Science**, v. 38, n. 4, p. 447 – 458, 2004. http://dx.doi.org/10.1287/trsc.1030.0069

KRAJEWSKA, M. A.; KOPFER, H. Transportation planning in freight forwarding companies: Tabu search algorithm for the integrated operational transportation planning problem. **European Journal of Operational Research,** v. 197, n. 2, p. 741-751, 2009. http://dx.doi.org/10.1016/j.ejor.2008.06.042

LI J Q; BORENSTEIN D; MIRCHANDANI P B. Truck schedule recovery for solid waste collection in Porto Alegre, Brazil. **International Transactions in Operational Research,** 15. p. 565-562, 2008. http://dx.doi.org/10.1111/j.1475-3995.2008.00648.x

MIGUEL, P. C. et al. **Metodologia da pesquisa em engenharia de produção e gestão de operações**. Rio de Janeiro: Campus, 2012.

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

RIBEIRO, C.; SOUMIS, F. A Column Generation Approach to the multiple Depot Vehicle Scheduling Problem. **Operations Research,** v. 42, p. 41-52, 1994. http://dx.doi.org/10.1287/opre.42.1.41

ROHDE, L. R. **Desenvolvimento de heurística para solução do problema de escalonamento de veículos com múltiplas garagens.** 2008. 122f. Tese (Doutorado em Administração) – Escola de Administração, UFRGS, Porto Alegre, 2008.

WANG, Y. Research of Multi-Depot Vehicle Routing Problem by Cellular Ant Algorithm. **Journal of Computers.** v. 8, n. 7, p. 1722-1727, 2013. http://dx.doi.org/10.4304/jcp.8.7.1722-1727

ZAPFEL, G., BRAUNE, R, BOGL, M. **Metaheuristic Search Concepts.** Springer: 2010. http://dx.doi.org/10.1007/978-3-642-11343-7

Artigo recebido em 30/10/2012 e aceito para publicação em 23/06/2014
DOI: http://dx.doi.org/ 10.14488/1676-1901.v14i3.1475