

Uma nova heurística baseada na meta-heurística VNS para o problema de agrupamento de entregas em veículos de uma frota heterogênea

Jorge Von Atzingen dos Reis¹ e Cláudio Barbieri da Cunha²

Resumo: Este artigo trata do problema de distribuição física que envolve o agrupamento e a alocação de entregas a uma frota de veículos visando minimizar o frete total. Este problema surge em diferentes situações práticas em que os pontos a serem atendidos estão suficientemente próximos entre si de tal modo que as distâncias percorridas entre paradas consecutivas podem ser consideradas irrelevantes para o custo total de entregas da frota e, portanto, podem ser desconsideradas para o cálculo dos fretes pagos a terceiros que realizam esse serviço. O problema pode ser modelado como um problema de *bin-packing* unidimensional com *bins* de tamanho variável (do inglês *Variable Sized Bin-Packing Problem*, VSBPP), uma generalização do tradicional problema de *bin-packing* no qual *bins* (veículos) de diferentes capacidades e custos estão disponíveis para a alocação de um conjunto de objetos (cargas) de modo que o custo total dos *bins* (veículos) utilizados seja mínimo. Neste trabalho é proposta uma nova heurística baseada na Busca em Vizinhança Variável (do inglês *Variable Neighborhood Search* ou VNS) para a resolução desse problema. Experimentos computacionais com instâncias derivadas de problemas *benchmarking* da literatura evidenciam que essa nova heurística produz resultados melhores em tempos de processamento bastante reduzidos.

Palavras-chave: busca em vizinhança variável, problema de agrupamento e alocação de entregas, meta-heurística

Abstract: In this paper we deal with the physical distribution problem which comprises grouping and assigning deliveries to a heterogeneous fleet of vehicles aiming to minimize the total freight cost. This problem arises in several practical situations in which the customers to be served are geographically close enough to each other such that distances traveled between consecutive stops can be regarded as irrelevant when compared to the overall vehicle costs, and thus not considered when establishing the freight rate to be paid to third-party carriers. The problem can be modeled as one-dimensional Variable Sized Bin-Packing Problem (VSBPP), a generalization of the traditional bin-packing problem, in which bins (vehicles) with different sizes and costs are available for the assignment of the objects (deliveries) such that the total cost of the used bins (vehicles) is minimized. A new heuristic based on VNS (Variable Neighborhood Search) is proposed to solve this problem. Computational experiments conducted using instances derived from benchmark problems from the literature evidence that our heuristic produces improved results in very short times.

Keywords: variable neighborhood search, variable sized bin-packing problem, metaheuristic.

1. INTRODUÇÃO

No contexto logístico, a distribuição física de produtos geralmente envolve o roteamento de veículos, tendo em vista que o tamanho médio das cargas a serem entregues aos clientes não é suficiente para lotar um veículo, o que acarreta que cada veículo atenda a diversos clientes; conseqüentemente, busca-se determinar quais entregas devem ser alocadas a cada veículo de uma frota disponível, e qual a seqüência de paradas (ou roteiro) de cada veículo, de forma a minimizar o custo total do serviço, geralmente composto da soma ponderada dos custos proporcionais às distâncias percorridas e dos custos fixos dos veículos.

Entretanto, em diversas situações práticas um problema diferente pode surgir, que envolve o transporte de várias cargas para um único destino, ou a entrega para vários destinos próximos entre si, quando se torna necessário otimizar o agrupamento de entregas nos veículos de modo a reduzir a frota necessária e o custo com o frete pago aos transpor-

tadores.

Tal situação ocorre em sistemas de distribuição física tais como, por exemplo, a entrega de cimento ou de aço para a construção civil, os quais abrangem inúmeros clientes, localizados em municípios distintos e atendidos a partir de uma única base de origem dos veículos. Em muitos casos a distância de deslocamento entre base de origem, onde o veículo é carregado, até um município ou região onde se localizam as entregas a serem realizadas, é muito superior às distâncias de percurso em rota (i.e., entre entregas consecutivas). Características das cargas, dos tempos envolvidos (de entregas e de deslocamento do veículo) podem acarretar um número de entregas não elevado para cada veículo (em geral não mais do que 10 entregas); conseqüentemente, normalmente é necessário despachar vários veículos a um município, ou a uma região que compreende municípios vizinhos, próximos entre si, de forma a realizar as entregas para todos os clientes ali localizados (Miura, 2008).

Deve-se ressaltar que, na prática, não se admite que uma entrega possa ser dividida (ou fracionada) em mais de um veículo (estratégia conhecida como *split deliveries*), buscando facilitar o melhor aproveitamento das capacidades dos veículos. Isso decorre do fato de que para os clientes é indesejável receber a mercadoria fracionada, em mais de uma entrega, uma vez que isso dificulta e até impede a conferência do que está sendo entregue de acordo com o pedido, em termos dos itens e respectivas quantidades; já para

¹ Jorge Von Atzingen dos Reis, Programa de Pós-Graduação em Engenharia de Transportes, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil. (e-mail: jorgereis@pontal.ufu.br).

² Cláudio Barbieri da Cunha, Programa de Pós-Graduação em Engenharia de Transportes, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil. (e-mail: cbcunha@usp.br).

Manuscrito recebido em 26/7/2011 e aprovado para publicação em 4/11/2011. Este artigo é parte de TRANSPORTES v.19, n.3, 2011. ISSN: 2237-1346 (online).

os embarcadores, o fracionamento da carga implica na multiplicação de tarefas administrativas e operacionais, para cada pedido, tais como a emissão de mais documentos (notas fiscais, conhecimentos de transporte) e a maior dificuldade de conferência da carga sendo embarcada e, conseqüentemente, do correto atendimento dos pedidos.

Em particular no Brasil, a forma peculiar de contratação do transporte por caminhão, e de remuneração dos transportadores, favorece essa forma peculiar de agrupamento por região ou município, sem preocupação com o roteiro ou a distância entre paradas consecutivas, uma vez que, usualmente, o valor do frete a ser pago ao transportador não é determinado com base na distância total da rota (apurada ou estimada), como é prática usual em muitos países, principalmente os mais avançados, onde grande parte da literatura científica de programação de veículos se origina. Segundo a prática no mercado brasileiro, existe uma tabela pré-acordada entre o embarcador e o transportador que define um valor fixo diário por município e tamanho de veículo ao qual são alocadas entregas, independentemente do número de entregas e da ocupação do veículo; nesse caso, um custo adicional é incorrido apenas quando as entregas ocorrem em municípios distintos e mais distantes entre si (em geral, para distâncias superiores a 10-20 km), uma vez que, nesse caso, o veículo deve percorrer um trajeto maior, normalmente rodoviário, com eventual incidência de pedágio, a fim de concluir as entregas.

Em muitos problemas pode-se assumir produto único ou homogêneo, como é o caso, por exemplo, do cimento ensacado (Miura, 2008), ou de produtos em caixas, ou paletes, ou até mesmo em gaiolas, permitindo assim desconsiderar a questão do arranjo espacial ou acomodação das cargas nos veículos. Como se trata de um problema de entregas de carga fracionada (ou parcelada) pode-se assumir, sem perda de generalidade, que nenhuma entrega supera a capacidade do menor veículo disponível.

Conseqüentemente, esse problema de distribuição pode ser modelado como um problema conhecido na literatura como o problema de *bin-packing* (BPP), definido da seguinte forma: dadas $j = 1, \dots, n$ objetos (ou itens) com seus respectivos pesos w_j e $i = 1, \dots, m$ bins (ou mochilas) idênticas de capacidade finita c , determinar a alocação (ou designação) dos n itens aos bins (sendo $w_j \leq c, \forall j$), de tal modo que o número de bins utilizados seja mínimo, e seja respeitada a restrição de capacidade em cada um dos bins. No caso do problema de distribuição em questão, os veículos de uma frota homogênea correspondem aos bins, e os objetos (ou itens) às cargas a serem entregues, cujos respectivos clientes estão localizados em uma mesma localidade, ou em localidades próximas de tal sorte que frete de entrega não seja afetado pelas diferentes paradas para entrega ao longo do trajeto.

Problemas de *bin-packing* constituem um grupo de problemas desafiadores de otimização combinatória que pertencem a uma classe mais geral denominada “problemas da mochila” (do inglês *knapsack problems*), e que vêm sendo intensamente estudados há várias décadas, atraindo teóricos e práticos (Martello e Toth, 1990). O interesse dos teóricos deve-se, principalmente, à estrutura simples do problema, o que permite, além da exploração de diversas propriedades

do mesmo, a sua resolução como parte de problemas mais complexos. Já do ponto de vista prático, estes problemas aparecem em diversas aplicações reais, incluindo, entre outros, carregamento de veículos (Eilon e Christofides, 1971) e corte e empacotamento (Dyckhoff, 1990; Marques e Arenales, 2007; Hoto *et al.*, 2007).

No caso mais geral, a frota pode ser composta por veículos de diferentes tamanhos e capacidades (em geral, de dois a quatro tipos de veículos), aos quais estão associados custos diferentes. Nesse caso, busca-se não mais minimizar o número total de veículos, mas sim o custo (ou frete) total dos veículos efetivamente utilizados. Usualmente os veículos maiores possuem fretes unitários (por unidade de capacidade) menores, uma vez que tanto o custo fixo quanto o custo variável não são diretamente proporcionais à capacidade de carga do veículo; por exemplo, o consumo unitário de combustível não varia diretamente com a capacidade de carga; tampouco o salário do motorista costuma variar com o tamanho do veículo, o que faz com que, por exemplo, um veículo com o dobro da capacidade de carga do outro não tenha o seu custo fixo dobrado, nem tampouco os custos unitários por quilometro.

Esse problema mais geral com frota heterogênea é conhecido na literatura como o problema de *bin-packing* com bins heterogêneos, ou de tamanho variável (do inglês *variable sized bin-packing problem*, ou simplesmente VSBPP), que é objeto do presente trabalho, tendo em vista sua aplicação prática no contexto logístico e de transporte. Ao contrário do problema de *bin-packing* tradicional, em que os bins são idênticos, este é um problema mais complexo e ainda não muito explorado na literatura.

Assim, tendo em vista a resolução de problemas reais de distribuição física de produtos, é proposta, neste trabalho, uma nova heurística baseada em Busca em Vizinhança Variável (VNS) para o problema de *bin-packing* unidimensional com bins heterogêneos. O problema unidimensional considera que as entregas a serem alocadas aos veículos possuem uma única restrição de capacidade; no caso deste trabalho é considerada a dimensão peso. Mais especificamente, em relação do trabalho de Reis e Cunha (2009), o algoritmo proposto neste trabalho incorpora uma nova estrutura de vizinhança, denominada Reconstrução, que consiste de movimentos que desfazem parcialmente a solução corrente e a reconstruem através de uma heurística gulosa, possibilitando assim escapar de ótimos locais através de uma melhor exploração do espaço de soluções viáveis.

A heurística proposta é avaliada considerando novas instâncias derivadas de instâncias *benchmarking* na literatura. Busca-se encontrar um método de solução eficiente, que produza boas soluções em tempos de processamento reduzidos, a fim de permitir a sua aplicação na programação diária da distribuição de produtos aos pontos de entrega. Tal programação compreende a definição, para cada município ou região de destino, de quantos veículos de cada tipo serão necessários, e a alocação dos pontos de entrega aos mesmos, de modo que o custo total da frota utilizada seja mínimo. Tendo em vista a sua aplicação à programação diária de entregas, e ao prazo reduzido para essa atividade, o tempo de processamento, assim como a qualidade das soluções, são aspectos importantes para a avaliação da heurística pro-

posta.

Este artigo está organizado da seguinte forma: na próxima seção é apresentada uma breve revisão bibliográfica sobre o BPP e o VSBPP; já a seção 3 contém a formulação matemática do problema. Em seguida, na seção 4 é descrita a heurística proposta baseada em Busca em Vizinhança Variável (*Variable Neighborhood Search* - VNS). Os resultados dos experimentos computacionais realizados são apresentados na seção 5, incluindo uma comparação dos resultados com os obtidos através de um pacote de otimização. Por fim, as conclusões e recomendações estão na seção 6.

2. O PROBLEMA DE BIN-PACKING COM FROTA HETEROGÊNEA

Na literatura existem diversas heurísticas propostas para a resolução do Problema de Bin-packing (BPP). As mais populares são as heurísticas nas quais os objetos são ordenados em ordem decrescente de peso (w_i) e são aplicadas regras para a alocação dos objetos aos *bins*, tais como *first-fit*, *best-fit* ou *next-fit* (Coffman *et al.*, 1997).

Uma boa revisão dos principais trabalhos encontrados na literatura para o BPP é apresentada por Scholl (1997), que também propõe um algoritmo exato para a sua resolução. Carvalho (1999) propôs um algoritmo exato baseado em geração de colunas e *branch-and-bound*. Gupta e Ho (1999) propõem uma heurística construtiva baseada na melhor folga do *bin*, denominada *minimum bin slack* (MBS), com resultados superiores. Fleszar e Hindi (2002) propõem diversas heurísticas para resolver o problema de bin-packing unidimensional, incluindo algumas baseadas na heurística MBS proposta por Gupta e Ho (1999) e um algoritmo baseado em VNS (*Variable Neighborhood Search*). A heurística mais eficiente proposta por Fleszar e Hindi (2002) é um algoritmo que consiste de um método de perturbação MBS seguido por uma busca em vizinhança variável (VNS) resultando em uma ferramenta eficiente para encontrar resultados melhores e com tempos computacionais menores.

Motivados pela sua aplicação em problemas reais de distribuição física, Cunha *et al.* (2008) propuseram três heurísticas para a solução desse problema: duas baseadas em GRASP e uma baseada em Busca em Vizinhança Variável (VNS). Experimentos computacionais realizados usando problemas *benchmarking* evidenciaram que as três heurísticas, que são simples e de fácil implementação, fornecem resultados de boa qualidade em tempos de processamento reduzidos, possibilitando a sua aplicação a problemas práticos em logística.

Entretanto, no caso do Problema de Bin-Packing com frota heterogênea (VSBPP), a literatura é bem mais restrita, e só mais recentemente esta generalização do problema clássico passou a atrair a atenção dos pesquisadores. O VSBPP é *NP-hard*, uma vez que se reduz ao BPP no caso particular em que todos os *bins* são idênticos, que é *NP-hard* (Garey e Johnson, 1979). Segundo Zhang (1997) as heurísticas clássicas para o BPP não são capazes de fornecer bons resultados para o VSBPP. O trabalho de Friesen e Langston (1986) é um dos pioneiros sobre o VSBPP definindo limitantes (*bounds*) e apontando a direção de

pesquisas futuras com a utilização de heurísticas.

Burkard e Zhang (1997) e Kang e Park (2003) propõem algoritmos aplicados ao problema de VSBPP e determinam o desempenho no pior caso dos algoritmos para casos específicos. Kang e Park (2003) propõem dois algoritmos gulosos, os quais são testados em três casos especiais: os tamanhos dos itens e dos *bins* são divisíveis; somente o tamanho dos *bins* é divisível; ou o tamanho dos *bins* não é divisível. Nos casos em que o tamanho dos itens e dos *bins* são divisíveis, o algoritmo guloso proposto obtém a solução ótima.

Correia *et al.* (2008) utilizam uma formulação de programação linear inteira e uma reformulação do modelo de otimização discreta para a solução do VSBPP. Novas desigualdades sugeridas pelas variáveis do modelo discreto são adicionadas para melhorar o relaxamento dos limites do modelo linear original. Os resultados obtidos pelo trabalho de Correia *et al.* (2008) indicam que o modelo proposto obteve bons limites para a programação linear tornando-o adequado para resolver instâncias consideradas pequenas ou com até 1.000 itens. Para instâncias maiores o modelo ainda pode ser utilizado para a obtenção dos limites inferiores da solução.

Alves e Carvalho (2007) apresentam diferentes estratégias para aperfeiçoar o método de geração de colunas quando aplicado ao VSBPP e propõem dois algoritmos, um baseado em relaxação Lagrangiana e outro que resolve iterativamente sequências de modelos de agregação. Os resultados comprovam uma redução significativa no número de colunas geradas e no tempo computacional necessário.

Mais recentemente, Haouari e Serairi (2009) analisaram o desempenho empírico de seis heurísticas, sendo quatro baseadas em métodos construtivos a partir da solução exata de subconjuntos dos dados; uma heurística baseada em Algoritmo Genético e uma heurística baseada em cobertura de conjuntos (*set covering*). As heurísticas propostas foram testadas e comparadas em conjuntos de dados, com até 2.000 objetos e sete tamanhos de *bins*, randomicamente gerados. A heurística baseada em cobertura de conjunto obteve sucesso conseguindo soluções de alta qualidade em baixo tempo computacional.

3. FORMULAÇÃO MATEMÁTICA PARA O VSBPP

O problema de *bin-packing* com *bins* heterogêneos (VSBPP) pode ser descrito da seguinte forma: dados $j = 1, \dots, n$ objetos ou itens (representando as entregas a serem realizadas), com seus respectivos pesos w_j , e $i = 1, \dots, m$ *bins* (ou mochilas, denotando os veículos a serem utilizados), para os quais são conhecidos seus respectivos custos f_i e suas capacidades b_i , determinar a alocação (ou designação) dos n itens (entregas) aos m *bins* (veículos), de tal modo a minimizar o custo total dos *bins* utilizados, e respeitando-se as restrições de capacidade nos mesmos. Assume-se, sem perda de generalidade, que os objetos j estejam ordenados em ordem não crescente de peso ($w_1 \geq w_2 \geq \dots \geq w_n$) e que os *bins* i estejam ordenados em ordem não decrescente de tamanho ($b_1 \leq b_2 \leq \dots \leq b_m$); adicionalmente, assume-se ainda que $w_n \leq b_m$ e que o número m de *bins* seja suficientemente elevado a fim de assegurar a viabilidade do problema.

Sejam y_i uma variável binária que recebe o valor 1 quando o veículo i é utilizado na solução e zero caso contrário; x_{ij} a variável binária que assume o valor 1 quando a carga j é alocada ao veículo i e zero caso contrário.

Assim, a formulação matemática do VSBPP pode ser escrita da seguinte forma:

$$\text{Min} \sum_{i=1}^m f_i y_i \quad (1)$$

$$\sum_{j=1}^n w_j x_{ij} \leq b_i y_i \quad \forall i = 1, \dots, m \quad (2)$$

$$\sum_{i=1}^m x_{ij} = 1 \quad \forall j = 1, \dots, n \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad \forall i = 1, \dots, m \quad \forall j = 1, \dots, n \quad (4)$$

$$y_i \in \{0, 1\} \quad \forall i = 1, \dots, m \quad (5)$$

A função objetivo (1) busca minimizar o custo total dos veículos aos quais são alocadas às entregas. As restrições (2) asseguram que a capacidade de cada veículo i utilizado seja respeitada. Já as restrições (3) garantem que cada carga (ou objeto) j seja alocada a exatamente um veículo. As restrições (4) e (5) asseguram a integralidade das variáveis de decisão.

4. META-HEURÍSTICA VNS APLICADA AO VSBPP

O Método de Busca em Vizinhança Variável (VNS) é um método de busca local que consiste em explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança. Contrariamente a outras meta-heurísticas baseadas em métodos de busca local, o VNS não segue uma trajetória, mas sim explora vizinhanças gradativamente mais “distantes” da solução corrente e focaliza a busca em torno de uma nova solução se, e somente se, um movimento de melhoria é realizado.

As soluções vizinhas, também denominada vizinhança de

uma solução, correspondem a todas as soluções que podem ser encontradas a partir de movimentos que modificam a solução corrente, ou solução atual. O pseudocódigo desta meta-heurística é apresentado na Figura 1. Maiores detalhes desse algoritmo podem ser encontrados em Mladenovic & Hansen (1997).

No VNS, parte-se de uma solução inicial qualquer e, a cada iteração, seleciona-se aleatoriamente, a partir da solução corrente s , uma solução vizinha s' pertencente à vizinhança $N^{(k)}(s)$, que é obtida através da estrutura de vizinhança de ordem k ($1 \leq k \leq r$). Essa solução vizinha s' é então submetida a um procedimento de busca local. Se a nova solução obtida s'' , resultante dessa busca local no entorno de s' , for melhor que a solução corrente s (i.e., $f(s'') < f(s)$), a busca prossegue a partir da nova solução encontrada s'' , reiniciando a partir da primeira estrutura de vizinhança $N^{(1)}(s)$, que corresponde a $k = 1$. Caso contrário, a busca prossegue a partir da próxima estrutura de vizinhança $N^{(k+1)}(s)$. Esse procedimento é encerrado quando uma condição de parada for atingida (por exemplo, tempo limite de processamento, número máximo de iterações ou número máximo de iterações consecutivas sem melhorias). Deve-se destacar que a solução s' é obtida de maneira aleatória no passo 6 do pseudocódigo da Figura 1 com a finalidade de se evitarem ciclos, situação que pode ocorrer caso alguma regra determinística seja utilizada para a obtenção de s' .

4.1. Solução inicial

Para a aplicação do VNS, é necessária a geração de uma solução inicial s , na qual todas as entregas $j = 1, \dots, n$ são alocadas a veículos $i = 1, \dots, m$ da frota. A solução inicial s (passo 1 da Figura 1) é obtida através de uma heurística construtiva, cujo pseudocódigo é mostrado na Figura 2.

Inicialmente são ordenadas as cargas (objetos) em ordem não crescente de peso (isto é, $w_1 \geq w_2 \geq \dots \geq w_n$) (passo 1) e os veículos (*bins*) em ordem não crescente de capacidade (passo 2) utilizando o método de ordenação do algoritmo *QuickSort*. Maiores detalhes sobre o algoritmo *QuickSort* podem ser obtidos em Hoare (1962).

Em seguida, cada uma das cargas é inserida, uma a uma e em ordem, no primeiro veículo, em ordem de capacidade, que possuir espaço suficiente para recebê-la (passos 5 a

```

Início VNS
1. Seja  $s_0$  uma solução inicial e  $r$  o número de estruturas de vizinhança;
2.  $s \leftarrow s_0$ ; {Solução corrente}
3. Enquanto (Critério de parada não satisfeito) faça
4.  $k \leftarrow 1$ ; {Tipo de estrutura de vizinhança}
5. Enquanto ( $k \leq r$ ) faça
6. Gere um vizinho qualquer  $s' \in N^{(k)}(s)$ ;
7.  $s'' \leftarrow \text{BuscaLocal}(s')$ ;
8. Se  $f(s'') < f(s)$  então
9.  $s \leftarrow s''$ ;
10.  $k \leftarrow 1$ ;
11. senão  $k \leftarrow k + 1$ ;
12. Fim-se;
13. Fim-enquanto;
14. Fim-enquanto;
15. Retorne  $s$ ;
Fim VNS

```

Figura 1. Pseudocódigo com a estrutura geral de funcionamento do VNS

```

Início SoluçãoInicial
1. Ordenar cargas/ entregas em ordem decrescente de peso  $w_j$ ;
2. Ordenar veículos em ordem crescente de capacidade  $b_i$ ;
3. Para cada veículo  $i=1$  até  $m$  faça
4.    $\bar{b}_i = b_i$ 
5. Inicializar o custo total da solução  $C \leftarrow 0$ 
6. Para cada carga  $j = 1$  até  $n$  faça
7.    $i \leftarrow 0$ 
8.   Enquanto  $i \leq m$  e  $w_j > \bar{b}_i$  faça
9.      $i \leftarrow i + 1$ 
10.    Se  $i \leq m$  então
11.       $x_{ij} \leftarrow 1$ 
12.       $\bar{b}_i = \bar{b}_i - w_j$ 
13.      Se  $y_i = 0$  então
14.         $y_i \leftarrow 1$ 
15.         $C \leftarrow C + f_i$ 
16.      Fim-se
17.    Fim-se
18. Fim-para
Fim SoluçãoInicial

```

Figura 2. Pseudocódigo da heurística construtiva para obtenção da solução inicial

14). Essa heurística construtiva gera somente soluções viáveis, isto é, garante que todos os objetos serão alocados a exatamente um único veículo e que nenhum veículo possuirá a sua capacidade violada.

Assume-se que a frota de veículos de cada tipo seja suficientemente grande a fim de possibilitar a alocação de todos os objetos a cada um dos tipos de veículo; em outras palavras, o número de veículos de cada tamanho é grande o suficiente para transportar todos os objetos do problema, de modo que a condição expressa no passo 8 da Figura 2 seja sempre satisfeita.

4.2. Estruturas de vizinhança

Para a aplicação do método de busca em vizinhança variável é necessário obter uma solução vizinha da solução corrente (passo 6 da Figura 1). Essa solução vizinha é gerada a partir de estruturas de vizinhança que visam encontrar soluções cada vez mais “distantes” da solução corrente. A sequência proposta de movimentos para o VNS é a seguinte:

- Troca 1-1;
- Troca 2-0A;
- Troca 2-0B;
- Troca 2-1; e
- Reconstrução.

Na estrutura de vizinhança Troca 1-1 escolhe-se aleatoriamente um veículo, que não esteja vazio, e se sorteia aleatoriamente uma entrega desse veículo. Essa entrega é candidata a ser removida do veículo ao qual está alocada e trocada com outra entrega alocada a outro veículo da frota. São testadas todas as possibilidades de troca da entrega selecionada aleatoriamente com as entregas alocadas aos demais veículos, até que uma troca viável seja encontrada. Caso não seja encontrada nenhuma troca viável em termos das capacidades dos veículos cujas respectivas entregas são candidatas a serem trocadas, é realizada uma nova tentativa de troca, ou seja, repete-se o procedimento de escolher outro veículo da frota e selecionar aleatoriamente uma entrega

alocada ao mesmo.

Já na estrutura de vizinhança Troca 2-0A é escolhido aleatoriamente um veículo, que não esteja vazio, e sorteadas duas entregas desse veículo. Estas duas entregas são removidas do veículo e o movimento de vizinhança tenta inserir ambas as entregas em outro veículo da frota, até ser gerado um movimento de vizinhança viável. Caso a troca não resulte em um movimento viável, é realizada uma nova tentativa de troca selecionando outro veículo da frota.

Na estrutura de vizinhança Troca 2-0B escolhe-se aleatoriamente um veículo, que não esteja vazio, e sorteiam-se duas entregas desse veículo. Estas duas entregas são removidas do veículo e através do movimento de vizinhança tenta-se inseri-las em outro veículo da frota, sendo que as entregas podem ser inseridas em um mesmo veículo ou em dois veículos distintos, de maneira mais flexível que o movimento de troca 2-0A, em que ambas as entregas devem ser alocadas a um mesmo veículo.

Já na estrutura de vizinhança denominada Troca 2-1 escolhe-se aleatoriamente um veículo, que não esteja vazio, e sorteiam-se duas entregas desse veículo. Estas duas entregas são removidas do veículo, e o movimento de vizinhança tenta inserir ambas as entregas em outro veículo da frota, sendo que este veículo que recebeu as duas entregas cede uma entrega (escolhida aleatoriamente) para o veículo de onde foram originariamente removidas as duas entregas.

Por fim, propõe-se a estrutura de vizinhança denominada Reconstrução, que consiste em retirar todas as entregas do veículo i^* com maior folga de capacidade \bar{b}_i dentre os veículos utilizados (isto é, que possuam pelo menos uma entrega alocadas a ele) e também remover 20% das entregas alocadas a cada um dos demais veículos; em seguida, tenta-se sistematicamente inserir as entregas removidas do veículo i^* nos demais outros veículos até encontrar um movimento viável. As entregas a serem removidas dos demais veículos são escolhidas aleatoriamente e utiliza-se o método da solução inicial descrito na Figura 2 para realocar as entre-

gas nos demais veículos. Esta estrutura de vizinhança permite uma maior perturbação na solução e, conseqüentemente, uma melhor exploração do espaço de soluções viáveis devido a sua capacidade de escapar de ótimos locais.

Deve-se notar que as estruturas de vizinhança propostas são progressivamente mais complexas em termos dos tamanhos das vizinhanças a serem exploradas em cada movimento, o que está de acordo com os princípios que norteiam o VNS.

4.3. Busca local

Uma vez encontrada uma nova solução viável s' vizinha da solução corrente, é realizada uma busca local com a finalidade de tentar melhorar s' (passo 7 da Figura 1). Essa busca local é realizada utilizando-se um movimento de realocação de entregas, uma a uma, de modo a buscar diminuir o número de veículos utilizados, uma vez que o custo da solução depende apenas do número de veículos de cada tipo utilizado, e não das entregas alocadas a cada veículo, assegurada a viabilidade das restrições de capacidade dos veículos. Esse procedimento de melhoria da solução corrente corresponde a uma busca na vizinhança, em que a vizinhança é definida como o espaço de soluções possíveis de serem atingidas a partir da transferência de um objeto a outro veículo. Neste contexto, a busca local considera como um ótimo local uma solução na qual transferências adicionais de um objeto de um veículo a outro não são possíveis.

Mais especificamente, o procedimento de busca local é realizado da seguinte maneira: os veículos efetivamente utilizados (isto é, tais que $y_i = 1$) são ordenados em ordem decrescente de folga \bar{b}_i . Seleciona-se o veículo i^* com maior capacidade ociosa \bar{b}_{i^*} e ordenam-se as entregas alocadas ao mesmo (i.e. tais que $x_{i^*j} = 1$) por ordem decrescente de peso w_j . Em seguida, tentar realocar cada uma das entregas desse veículo i^* , em ordem, a outro veículo; ou seja, a realocação é sempre iniciada pela entrega j^* de maior peso alocada ao veículo i^* , e sempre buscando o outro veículo i que corresponda à menor folga de capacidade \bar{b}_i tal que $w_j \leq \bar{b}_i$, isto é, que o veículo i tenha capacidade para receber a carga j^* candidata a ser realocada.

A realocação de cargas do veículo i^* é repetida até que todas as entregas originalmente alocadas ao mesmo tenham sido reinseridas em outros veículos, ou não seja mais possível encontrar alguma reinserção viável devido à falta de capacidade nos demais veículos. Neste caso, é verificado o menor tipo de veículo que permite acomodar as cargas remanescentes no veículo i^* , que não puderam ser reacomodadas em outros veículos, e atualizado o custo da solução.

Deve-se notar que, a cada nova reinserção de uma carga do veículo i^* em outro veículo i' , há necessidade de reatualizar a ordenação dos demais veículos segundo ordem decrescente de folga \bar{b}_i , uma vez que a inserção da entrega j^* acarreta a alteração da capacidade ociosa do veículo i' que a recebe. No entanto, isso é feito de maneira simplificada e rápida, sem necessidade de uma reordenação completa de todos os veículos, uma vez que basta comparar a nova folga $\bar{b}_{i'}$ do veículo i' com as folgas dos demais veículos i que

possuíam folga menor que a do veículo i' (isto é, com $\bar{b}_i < \bar{b}_{i'}$) antes da inserção da carga j^* , de modo a encontrar a nova posição do veículo i' na lista ordenada de veículos ordem decrescente de folga. Em outras palavras, apenas o novo veículo i' que receber da carga j^* tem a sua posição relativa alterada, sendo que os demais veículos permanecem na ordem correta. Deve-se notar que, embora a folga do veículo i^* seja alterada a cada vez que uma entrega é removida, não há necessidade de rever a sua ordem dentre os veículos, uma vez que a sua ociosidade \bar{b}_{i^*} só aumenta conforme entregas vão sendo retiradas e realocadas a outros veículos; como i^* corresponde ao veículo com maior ociosidade de capacidade ao início da realocação, sua posição não muda à medida que cargas vão sendo retiradas do mesmo.

Em síntese, através da busca local busca-se reduzir o número de veículos utilizados por meio da realocação das entregas dos veículos com maior ociosidade \bar{b}_i para veículos que possuem uma maior quantidade de carga, mas ainda não estão transportando carga suficiente para atingir o limite de capacidade.

4.4. Função de avaliação

A função de avaliação tem o objetivo de avaliar a qualidade de uma solução vizinha obtida através de um movimento realizado por uma das estruturas de vizinhança utilizada pela meta-heurística VNS. A função de avaliação proposta visa a avaliar a qualidade de uma solução para o VSBPP considerando o custo da frota alocada e pode ser calculada através da expressão (6).

$$FA = \sum_{i=1}^3 \text{Custo}_i \text{NumBinTipo}_i \quad (6)$$

em que,

Custo_i : custo por utilizar um veículo do tipo i ; e
 NumBinTipo_i : número de veículos (*bins*) do tipo i utilizados.

5. EXPERIMENTOS COMPUTACIONAIS

A heurística baseada em VNS para o VSBPP proposta neste trabalho foi testada com dados de derivados de instâncias da literatura. As novas instâncias são derivadas das instâncias *benchmarking* do BPP disponíveis na biblioteca denominada *OR Library* (<http://people.brunel.ac.uk/~mastjib/jeb/orlib/binpackinfo.html>), originalmente propostas por Falkenauer (1996) para o BPP, no qual os *bins* são idênticos, e que haviam sido utilizadas por Cunha *et al.* (2008). A diferença básica das novas instâncias aqui considerada foi a necessidade de geração de mais dois tamanhos de *bins* (veículos) além do tamanho original, (cuja capacidade é igual a 150), segundo o proposto pelo autor.

Considerando por base o tamanho de *bin* correspondente à capacidade igual a 150 conforme proposto por Falkenauer (1996), e atribuindo um custo de 100, foram criados mais dois tipos de *bins* da seguinte forma: um tipo de *bin* com capacidade 25% inferior e custo 20% inferior e um tipo de *bin* com capacidade 25% superior e custo 20% superior, e cujos valores resultantes são sumarizados na Tabela 1. Des-

Tabela 1. Custo e Capacidade dos veículos

	<i>Grande</i>	<i>Médio</i>	<i>Pequeno</i>
Custo (u.m.)	120	100	80
Capacidade (u.c.)	187	150	112

sa forma, o *bin* de maior tamanho é o que apresenta o menor custo por unidade de capacidade, tal como ocorre na prática onde veículos maiores são os de menor custo unitário.

Os testes foram realizados em um total de três conjuntos de dados com problemas com o número de objetos variando entre 10 e 1.000. O programa desenvolvido em C++ foi rodado cinco vezes para cada instância de teste. Foi utilizado um computador Core2Quad 2.4GHz com 2GB de RAM, embora não haja nenhuma limitação que impeça o processamento da heurística baseada em VNS em computadores mais modestos.

A fim de determinar o número de veículos, de cada tamanho, suficiente para transportar todos os objetos do problema, a heurística da solução inicial é aplicada isoladamente para cada tamanho de veículo de forma a alocar todas as entregas para um único tipo de veículo. Deve-se notar que essa heurística aplicada desta forma fornece um limitante superior da frota total necessária, uma vez que não há limitação no número de veículos de cada tipo disponível.

As melhores soluções obtidas para cada instância são apresentadas nas Tabelas 2, 3 e 4.

Os resultados apresentados na Tabela 2 são relativos aos problemas de pequeno porte com o número de entregas a serem alocadas aos veículos variando entre 10 e 40. Este conjunto de problemas foi definido como pequeno porte devido a ser possível encontrar a solução ótima para cada um deles utilizando o ILOG CPLEX, tendo possibilitado determinar os valores da função de avaliação das soluções ótimas apresentadas na Tabela 2 através da resolução do modelo matemático descrito na seção 3.

A primeira coluna da Tabela 2 indica o número de obje-

tos (cargas) a serem alocadas. A segunda coluna corresponde ao limite inferior (*lower bound*) para cada uma das instâncias de teste, obtido pela expressão (7); esse limite inferior representa o custo mínimo para cada instância, e é calculado considerando que todas as cargas sejam alocadas ao veículo com a melhor relação custo/benefício.

$$LowerBound = \frac{Custo_{GDE} * \sum_{i=1}^n Peso_i}{Capacidade_{GDE}} \quad (7)$$

em que,

$Custo_{GDE}$: custo por utilizar um veículo do tipo grande;

$Capacidade_{GDE}$: capacidade dos veículos do tipo grande;

$Peso_i$: peso da carga i ; e

n : total de cargas da instância do problema.

As colunas *FA* (*Função de Avaliação*) da Tabela 2 indicam o custo da solução obtida para cada instância tanto pelo pacote de programação matemática ILOG CPLEX 11.0 (o qual possibilitou determinar a solução ótima somente para as instâncias da Tabela 2) quanto pelo VNS. As colunas *Tempo* representam os tempos computacionais, medido em segundos, para a obtenção da solução apresentada para cada um dos dois métodos. As colunas *Grande*, *Médio* e *Pequeno* apresentam, respectivamente, o número de veículos do tipo grande, o número de veículos do tipo médio e o número de veículos do tipo pequeno utilizados na solução encontrada, tanto pelo CPLEX quanto pelo VNS, para cada instância considerada. Os resultados destacados em negrito mostram as instâncias nas quais o método proposto baseado em VNS obteve a solução ótima, o que ocorreu em 14 das 17 instâncias testadas. Nas instâncias em que o algoritmo proposto não atingiu a solução ótima os resultados foram apenas 1,3% superiores, em média. E o tempo computacional (apresentado na tabela em segundos) utilizado pela metodologia proposta é expressivamente inferior ao tempo uti-

Tabela 2. Resultados obtidos para instâncias de pequeno porte

<i>Nº de objetos</i>	<i>Lower Bound</i>	<i>Solução Ótima</i>			<i>Solução VNS</i>						
		<i>FA</i>	<i>Tempo (seg)</i>	<i>Grande</i>	<i>Médio</i>	<i>Pequeno</i>	<i>FA</i>	<i>Tempo (seg)</i>	<i>Grande</i>	<i>Médio</i>	<i>Pequeno</i>
10	371	400	2	1	2	1	400	0	2	0	2
12	443	460	1	3	1	0	460	1	3	1	0
12	464	480	4	3	0	0	480	0	4	0	0
15	589	600	0	5	0	0	600	1	5	0	0
20	681	700	1	5	1	0	700	0	5	1	0
25	1.037	1.060	9	8	1	0	1.060	1	8	1	0
40	1.635	1.660	48	13	1	0	1.660	2	13	1	0
40	1.634	1.660	10	13	1	0	1.660	1	13	1	0
40	1.592	1.620	6	12	1	1	1.620	2	12	1	1
40	1.530	1.540	2.104	12	1	0	1.540	2	12	1	0
40	1.496	1.520	1.460	11	2	0	1.520	1	11	2	0
40	1.633	1.660	1.671	13	1	0	1.660	2	13	1	0
40	1.503	1.520	77	11	2	0	1.520	2	11	2	0
40	1.634	1.660	9	13	1	0	1.660	2	13	1	0
40	1.542	1.560	1	13	0	0	1.580	1	10	3	1
40	1.542	1.560	3	13	0	0	1.580	2	9	5	0
40	1.551	1.560	2.823	13	0	0	1.580	2	11	1	2
Total	20.877	21.220	8.229	162	15	2	21.280	22	155	22	6

Tabela 3. Resultados obtidos para instâncias de médio porte

Nº de objetos	Lower Bound	Solução CPLEX				Solução VNS				GAP	
		FA	Grande	Médio	Pequeno	FA	Tempo (seg)	Grande	Médio		Pequeno
35	1.362	1.400	11	0	1	1.380	1	10	1	1	2,24%
40	1.444	1.500	11	1	1	1.480	1	9	4	0	3,26%
55	1.998	2.040	17	0	0	2.020	2	16	1	0	1,82%
60	2.356	2.400	20	0	0	2.380	2	19	1	0	1,58%
70	2.783	2.860	23	1	0	2.820	3	22	1	1	2,33%
120	4.921	5.000	41	0	1	4.960	8	38	4	0	1,26%
120	4.687	4.760	39	0	1	4.720	16	37	2	1	1,29%
120	4.930	5.040	42	0	0	4.980	8	39	3	0	1,91%
120	4.530	4.600	37	0	2	4.580	10	36	1	2	1,28%
120	4.644	4.720	38	0	2	4.700	8	36	3	1	1,36%
120	4.687	4.760	39	0	1	4.740	7	37	3	0	1,29%
250	9.462	9.680	80	0	1	9.520	26	77	2	1	1,99%
250	9.507	9.680	80	0	1	9.560	23	78	2	0	1,53%
250	9.731	9.880	81	0	2	9.780	25	80	1	1	1,25%
250	9.737	9.880	81	0	2	9.780	61	79	3	0	1,38%
250	9.659	9.820	81	1	0	9.740	32	77	5	0	1,17%
250	9.572	9.680	80	0	1	9.620	28	78	1	2	0,86%
500	19.689	20.720	167	6	1	19.780	169	161	3	2	3,61%
500	19.862	20.760	168	6	0	19.920	162	162	4	1	4,63%
500	19.339	20.260	164	5	1	19.420	163	156	7	0	3,72%
500	16.567	20.400	165	6	0	19.600	135	161	2	1	2,69%
500	18.953	19.800	160	6	0	19.020	234	155	1	4	4,02%
500	18.968	19.780	159	7	0	19.020	184	157	1	1	4,43%
500	19.691	20.540	167	5	0	19.800	221	162	2	2	3,88%
500	19.282	20.060	164	3	1	19.340	123	158	3	1	3,62%
500	19.297	20.060	163	5	0	19.360	184	159	2	1	3,55%
500	19.583	20.319	166	4	0	19.620	139	162	1	1	3,37%
500	19.325	20.060	164	3	1	19.420	287	158	3	2	3,41%
500	19.213	19.940	162	5	0	19.340	131	158	3	1	3,39%
500	18.781	19.380	159	3	0	18.840	143	155	0	3	2,83%
30	1.215	1.240	7	4	0	1.240	2	9	0	2	1,32%
45	1.966	2.000	15	2	0	2.000	2	15	2	0	1,28%
50	2.002	2.040	17	0	0	2.040	3	12	6	0	1,60%
Total	349.743	365.059	2.968	73	20	354.520	2.543	2.868	78	32	

lizado para a obtenção da solução ótima.

Outra observação interessante é que a solução ótima esteve em média 2,1% acima do valor do limite inferior (*lower bound*) e a solução obtida pela metodologia proposta esteve em média 2,4% acima. No trabalho de Reis e Cunha (2009), a heurística proposta conseguiu obter a solução ótima em somente 10 das 17 instâncias testadas e nas demais obteve um resultado, em média, 1,46% acima do valor obtido pela solução ótima.

Os resultados apresentados na Tabela 3 correspondem a instâncias com 30 e 500 objetos, e foram considerados como instâncias de médio porte devido ao CPLEX não conseguir encontrar uma solução ótima em um tempo inferior a uma hora (3.600 segundos). A coluna *solução CPLEX* apresentada na Tabela 3 apresenta a melhor solução encontrada dentro de uma hora e a coluna *GAP* apresenta a porcentagem que esta solução está acima do limite inferior calculado pelo CPLEX.

Dentre os 33 problemas resolvidos de médio porte em trinta deles a metodologia proposta encontrou uma solução melhor que a obtida pelo CPLEX (em média 2,2% abaixo). Em 3 dos problemas a heurística proposta possibilitou en-

contrar uma solução igual à obtida pelo CPLEX. No geral, a heurística proposta ficou em média 1,37% acima do valor do limite inferior, ou seja, abaixo do valor que a solução ótima esteve acima do limite inferior (2,1%) nos problemas de pequeno porte.

A heurística proposta por Reis e Cunha (2009) foi testada em 23 instâncias de médio porte e obteve resultados melhores que os obtidos pelo CPLEX em somente 11 instâncias. Em 7 instâncias obteve o resultado igual ao do CPLEX e em 5 instâncias obteve resultados piores, ficando em média 0,92% acima. A heurística de Reis e Cunha (2009) obteve resultados 2,59% acima do limite inferior.

Os resultados da Tabela 4 são relativos às instâncias de grande porte e correspondem a 1.000 cargas a serem alocados aos veículos. Para os problemas cujos resultados estão apresentados na Tabela 4, o CPLEX não conseguiu nem carregar o modelo matemático na memória do computador utilizado nos testes, mesmo configurando o CPLEX para utilizar o espaço disponível no disco rígido para diminuir a necessidade de utilização da memória RAM do computador utilizado nos testes. Isso evidencia a complexidade matemática do problema tratado, em particular o elevado núme-

Tabela 4. Resultados obtidos para instâncias de grande porte

Nº de objetos	Lower Bound	Solução VNS				
		FO	Tempo (seg)	Grande	Médio	Pequeno
1.000	38.249	38.340	468	314	5	2
1.000	38.905	39.000	487	321	4	1
1.000	39.380	39.500	512	319	9	4
1.000	39.444	39.540	629	323	7	1
1.000	38.087	38.220	487	314	3	3
1.000	38.256	38.420	486	313	7	2
1.000	37.844	37.920	690	308	8	2
1.000	38.704	38.800	481	317	6	2
1.000	38.250	38.340	578	313	3	6
1.000	38.105	38.260	1.068	311	7	3
1.000	38.337	38.420	754	313	7	2
1.000	38.450	38.540	471	314	7	2
1.000	37.655	37.760	1.071	306	8	3
1.000	37.947	38.000	713	312	4	2
1.000	37.814	37.920	758	308	8	2
1.000	38.574	38.680	1.010	314	6	5
1.000	38.691	38.820	483	317	7	1
1.000	38.765	38.940	663	319	5	2
1.000	38.277	38.340	637	312	9	0
1.000	38.336	38.440	536	310	10	3
Total	768.020	770.200	12.982	6.278	130	48

ro de variáveis de decisões e de restrições, que não pode ser resolvido em máquinas com 32 bits devido a limitações de memória.

A heurística baseada em VNS proposta neste trabalho conseguiu resolver os problemas relativos às instâncias de grande porte com um tempo computacional, aproximadamente entre cinco e onze minutos (o tempo indicado na Tabela 4 está em segundos). Os resultados obtidos estão 0,28% acima do limite inferior (*lower bound*), o que pode indicar que a heurística proposta pode ter conseguido se aproximar da solução ótima. Cabe ressaltar que a mesma consegue resolver estes problemas de grande porte mesmo em computadores mais modestos (entretanto, obviamente o tempo de processamento será maior do que os apresentados na Tabela 4).

Por outro lado, a heurística proposta por Reis e Cunha (2009) obteve resultados, em média, 0,83% acima do limite inferior, enquanto que a heurística proposta neste trabalho obteve resultados 0,28% acima do limite inferior para o mesmo conjunto de instâncias de teste de grande porte. Esta diferença entre os resultados obtidos comprovam a eficácia da nova estrutura de vizinhança proposta e a melhoria obtida devido ao uso do novo movimento de vizinhança denominada reconstrução.

5.1. Variação nos resultados obtidos

Tendo em vista a característica aleatória da heurística baseada em VNS, em que um vizinho da solução corrente é obtido aleatoriamente (passo 6 da Figura 1), cada instância de teste foi submetida a cinco processamentos distintos, a fim de avaliar a sua robustez. Os resultados apresentados anteriormente nas Tabelas 2 a 4 correspondem ao melhor resultado obtido dentre os cinco processamentos. Já a Tabela 5 apresenta um detalhamento das soluções obtidas; mais especificamente, são apresentados o menor valor, o valor médio e o desvio padrão para cada uma das instâncias de

grande porte. Observe que não existem grandes diferenças entre os valores médios e os valores mínimos encontrados para cada uma das instâncias testadas, mostrando que a heurística desenvolvida apresenta boa robustez.

6. CONSIDERAÇÕES FINAIS

Este trabalho abordou um problema de distribuição física em que se busca a otimização do agrupamento de entregas e carregamento nos veículos de modo a reduzir a frota necessária e o custo com o frete pago aos transportadores, que é modelado como um problema de *bin-packing* com *bins* heterogêneos (VSBPP). Trata-se de um problema ainda pouco estudado na literatura, ao contrário do problema com *bins* idênticos.

A metodologia proposta baseia-se na meta-heurística VNS e para a sua avaliação foram consideradas instâncias de teste derivadas de instâncias *benchmarking* da literatura para o BPP, adaptadas para o VSBPP. Também foi feita uma comparação com os resultados exatos obtidos através da utilização de um pacote de otimização linear (ILOG CPLEX versão 11).

Os resultados indicam que o algoritmo proposto permitiu obter a solução ótima na maioria dos testes, nos quais foi possível a obtenção da solução ótima da mesma. E nos demais testes a solução obtida sempre esteve próxima do limitante inferior (menos de 2% acima), sendo que as soluções ótimas encontradas encontram-se sempre acima dos respectivos limitantes (mais do que 2% em média), o que indica que a estratégia de solução proposta atinge resultados de excelente qualidade com tempos de processamento muito reduzidos. A heurística proposta, baseada em VNS, também se mostrou mais rápida que a solução ótima do modelo matemático e nos problemas de grande porte foi a única a conseguir obter uma solução, a qual esteve menos de 1% acima do limite inferior.

Tabela 5. Variação nos resultados obtidos para instâncias de grande porte

<i>Número de objetos</i>	<i>Mínimo</i>	<i>Médio</i>	<i>Desvio Padrão</i>
1.000	38.340	38.356	0,02%
1.000	39.000	39.036	0,07%
1.000	39.500	39.520	0,07%
1.000	39.540	39.560	0,09%
1.000	38.220	38.240	0,05%
1.000	38.420	38.432	0,05%
1.000	37.920	37.948	0,06%
1.000	38.800	38.820	0,05%
1.000	38.340	38.388	0,08%
1.000	38.260	38.268	0,03%
1.000	38.420	38.460	0,06%
1.000	38.540	38.552	0,02%
1.000	37.760	37.776	0,04%
1.000	38.000	38.020	0,04%
1.000	37.920	37.948	0,05%
1.000	38.680	38.724	0,08%
1.000	38.820	38.848	0,05%
1.000	38.940	38.960	0,04%
1.000	38.340	38.380	0,08%
1.000	38.440	38.492	0,09%

Inúmeras são as possíveis extensões para a continuidade desta pesquisa, tendo em vista o número reduzido de trabalhos na literatura, incluindo a investigação de novos métodos de solução e a aplicação e validação dos métodos propostos a problemas reais que envolvem o agrupamento de cargas no contexto da distribuição física de produtos.

Outra proposta para a continuidade deste trabalho é o estudo do problema de *bin-packing* duas dimensões (volume e peso dos objetos) e frota heterogênea. Este é um problema muito comum no cotidiano das empresas e ainda praticamente inexplorado na literatura científica.

AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq – Brasil pelo apoio para a realização da pesquisa que resultou no presente trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

Alves, C. e J. M. V. Carvalho (2007) Accelerating Column Generation for Variable Sized Bin-Packing Problems, *European Journal of Operational Research*, v. 183, n. 3, p. 1333–1352

Burkard, R. E. e G. Zhang (1997) Bounded Space On-Line Variable-Sized Bin Packing. *Acta Cybernetica* v. 13, p. 63–76.

Carvalho, J.M.V. (1999) Exact Solutions of Bin-Packing Problems Using Column Generation and Branch-and-Bound, *Annals of Operations Research*, v. 86, p. 629–659.

Coffman, E.G.; M.R. Garey e D.S. Johnson (1997) Approximation Algorithm for Bin-Packing: a Survey. *Approximation algorithm for NP-hard problems*. PWS Publishing, Boston, p. 46–93.

Correia, I.; L. Gouveia e F. Saldanha-da-Gama (2008) Solving the Variable Size Bin Packing Problem with Discretized Formulations. *Computers & Operations Research*, v. 35, n. 6, p. 2103–2113.

Cunha, C. B.; G. Manieri; H.T.Y. Yoshizaki; L. Maluta e L.R.S. Henriques (2008) Heurísticas para o Problema de Bin-Packing no Contexto da Distribuição Física de Produtos. *XL SBPO - 40º Simpósio Brasileiro de Pesquisa Operacional*, João Pessoa-PB. p. 712–723.

Dyckhoff, H. (1990) A Typology of Cutting and Packing Problems. *European Journal of Operational Research*, v. 44, n. 2, p. 145–159.

Eilon, S. e N. Christofides (1971) The Loading Problem, *Management Science*, v. 17, p. 259–268.

Falkenauer E. (1996) A Hybrid Grouping Genetic Algorithm for Bin Packing. *Journal of Heuristics* v. 2, n. 1, p. 5–30.

Fleszar, K. e K.S. Hindi (2002) New Heuristics for One-Dimensional Bin-Packing, *Computers & Operations Research*, v. 29, p. 821–839.

Friesen, D.K. e M.A. Langston (1986) Variable-Sized Bin Packing. *SIAM Journal on Computing*, v. 15, p. 222–230.

Garey, M.R. e D.S. Johnson (1979) *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman, San Francisco

Gupta, J.N.D. e J.C. Ho (1999) A New Heuristic Algorithm for the One-Dimensional Bin-Packing Problem. *Production Planning & Control*, v. 10, p. 598–603.

Haouari, M. e M. Serairi (2009) Heuristics for The Variable Sized Bin-Packing Problem. *Computers & Operations Research*, v. 36, p. 2877–2884.

Hoare, C. A. R. (1962) Quicksort. *Computer Journal*, v. 5, n. 1, p. 10–15.

Hoto, R.; M. Arenales e N. Maculan (2007) The One Dimensional Compartmentalized Knapsack Problem: a Case Study. *European Journal of Operational Research*, v. 183, n. 3, p. 1183–1195.

Kang, J. e S. Park (2003) Algorithms for the Variable Sized Bin-Packing Problem. *European Journal of Operation Research*, v. 147, n. 2, p. 365–372.

Marques, F. P. e M.N. Arenales (2007) The Constrained Compartmentalized Knapsack Problem. *Computers & Operations Research*, v. 34, n. 7, p. 2109–2129.

Martello, S. e P. Toth (1990) *Knapsack problems: algorithms and computer implementations*. Wiley, Chichester, UK

Miura, M. (2008) *Modelagem Heurística no Problema de Distribuição de Cargas Fracionadas de Cimento*. Dissertação (mestrado). Escola Politécnica, USP, São Paulo-SP.

Mladenovic, N. e P. Hansen (1997) Variable Neighborhood Search. *Computers & Operations Research*, v. 24, n. 11, p. 1097–1100.

Reis, J. A. e Cunha, C. B. (2009) Uma heurística baseada em busca em vizinhança variável para o problema de agrupamento de entregas em veículos de uma frota heterogênea. *XXIII Congresso de Pesquisa e Ensino em Transportes*, Vitória-ES. Anais do XXIII ANPET. Rio de Janeiro, ANPET.

Scholl, A.; R. Klein e C. Jürgens (1997) BISON: A Fast Hybrid Procedure for Exactly Solving the One-Dimensional Bin-Packing Problem. *Computers & Operations Research*, v. 24, n. 7, p. 627–645.

Zhang, G. (1997) A New Version of On-Line Variable-Sized Bin-Packing. *Discrete Applied Mathematics*, v. 72, p. 193–197.