



METAHEURÍSTICAS VNS-VND E GRASP-VND PARA PROBLEMAS DE ROTEAMENTO DE VEÍCULOS COM COLETA E ENTREGA SIMULTÂNEA

Lia Mara Borges de Freitas

Universidade Candido Mendes – UCAM-Campos
Rua Anita Pessanha, 100, Parque São Caetano, Campos dos Goytacazes, RJ, 28040-320.
liamarabf@yahoo.com.br

Fermín Alfredo Tang Montané

Universidade Candido Mendes – UCAM-Campos
Rua Anita Pessanha, 100, Parque São Caetano, Campos dos Goytacazes, RJ, 28040-320.
tang@ucam-campos.br

RESUMO

Neste artigo se apresentam metaheurísticas VNS-VND e GRASP-VND para resolver de forma aproximada o problema de Roteamento de Veículos com Coleta e Entrega Simultânea. Este problema é uma variante do problema clássico de roteamento de veículos na qual cada cliente faz ao mesmo tempo dois tipos de pedidos: coleta e entrega, e a carga do veículo em qualquer rota é uma mistura de pedidos de entrega e coleta. O objetivo é determinar o conjunto de rotas a serem percorridas pelos veículos que minimize o custo total de transporte. Ambas as heurísticas utilizam como mecanismo de busca local uma heurística VND (*Variable Neighborhood Descent*) que explora diferentes estruturas de vizinhanças. As metaheurísticas VNS-VND e GRASP-VND são testadas em problemas com $50 \leq n \leq 199$ clientes. A qualidade das soluções obtidas é avaliada pela comparação com os melhores resultados propostos na literatura.

PALAVRAS CHAVE. GRASP. VNS. Problema de roteamento de veículos. Metaheurística.

ABSTRACT

This article describes VNS-VND and GRASP-VND metaheuristics to generate approximated solutions for the vehicle routing problem with simultaneous pick-up and delivery. This problem is a variant of the classical Vehicle Routing Problem in which each client makes two types of requests: pick-ups and deliveries, and a vehicle's load in any given route is a mix of pick-up and delivery loads. The objective is to determine the set of routes to be used by the vehicles minimizing the total transportation cost. The proposed heuristics uses as a local search mechanism a Variable Neighborhood Descent (VND) heuristic that explores several neighborhood structures. The proposed metaheuristics is tested on problems with $50 \leq n \leq 199$ clients. The quality of the obtained solutions is evaluated by the comparison with the best results obtained from the literature.

KEYWORDS. GRASP. VNS. Vehicle routing problem. Metaheuristic.

1. Introdução

Os Problemas de Roteamento de Veículos com Pedidos de Coleta e Entrega Simultânea são variantes do Problema de Roteamento de Veículos Clássico (VRP). Nestas variantes, as entregas são realizadas a partir de um único depósito no início de cada rota, enquanto que as coletas são conduzidas até o depósito no fim da rota. Uma característica deste tipo de problemas é que a carga do veículo em uma rota qualquer é composta por dois tipos de itens de natureza diferente. Este tipo de problemas é observado com frequência na indústria de transporte de alimentos e bebidas, especificamente nos casos em que além da entrega de um tipo de produto é necessário recolher outros produtos com prazo de validade vencida. Outra aplicação importante deste tipo de problema acontece na indústria petrolífera. O problema surge no contexto de transporte de materiais entre o continente e as plataformas marítimas em alto mar, tarefa que é realizada por uma frota de embarcações. No Brasil, a empresa estatal PETROBRAS realiza este tipo operações a partir do porto de Imbetiba, na Baía de Campos, região norte do Estado do Rio de Janeiro.

Abordaremos duas variantes de roteamento com pedidos de coleta e entrega. O Problema de Roteamento de Veículos com Pedidos de Coleta e Entrega Simultânea e Frota Homogênea (PRVCES-Ho) e o Problema de Roteamento de Veículos com Pedidos de Coleta e Entrega Simultânea e Frota Heterogênea (PRVCES-He). Em ambos problemas, considera-se que a frota de veículos é ilimitada. Vale observar que dois tipos de carga coexistem no veículo que atende uma rota; a carga que ainda não foi entregue e a carga que já foi recolhida.

O PRVCES-Ho pode ser definido matematicamente da seguinte maneira. Seja $G = (V, A)$ um grafo onde $V = \{v_0, \dots, v_n\}$ representa o conjunto de vértices e $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ é o conjunto de arestas. O vértice v_0 representa o depósito que serve de base para a frota de veículos. Cada vértice $v_i \in V \setminus \{v_0\}$ representa um cliente que tem associadas duas demandas: coleta DC_i e entrega DE_i . Cada aresta (v_i, v_j) tem associada um custo não negativo, c_{ij} , que representa o custo de percurso. Além disso, considera-se que todos os veículos são idênticos, onde Q representa a capacidade do veículo. No PRVCES-He considera-se que a frota de veículos é formada por um conjunto de M tipos de veículos diferentes, onde cada tipo de veículo k possui capacidade Q_k .

O problema PRVCES-Ho consiste em determinar as rotas que serão utilizadas de maneira que: (1) as rotas se iniciam no depósito e terminam no depósito; (2) cada cliente é visitado exatamente uma vez; (3) ambas as demandas dos clientes são atendidas; (4) a carga máxima na rota não excede a capacidade do veículo utilizado; e (5) a distância total percorrida é minimizada.

No PRVCES-He, além das características mencionadas acima, é preciso determinar a melhor combinação de tipos de veículos que conformará a frota.

No presente artigo, propomos as metaheurísticas VNS-VND e GRASP-VND para resolver os problemas de roteamento com coleta e entrega simultânea. O presente artigo está organizado da seguinte maneira. Na seção 2, se apresenta uma revisão bibliográfica dos algoritmos desenvolvidos para resolver os problemas PRVCES-Ho e PRVCES-He. Nas Seções 3 e 4 descreve-se os algoritmos VNS-VND e GRASP-VND implementados neste trabalho. Os resultados computacionais são apresentados na Seção 5, enquanto na Seção 6 apresentamos as conclusões do trabalho.

2. Revisão Bibliográfica

As primeiras abordagens para resolver problemas de roteamento com coleta e entrega, especificamente o PRVCES-Ho, em geral, centraram-se no desenvolvimento de heurísticas construtivas. O primeiro trabalho sobre o assunto foi realizado por Min (1989) no contexto de um sistema de distribuição para bibliotecas públicas, considerando veículos de igual capacidade. Foi proposta uma heurística clássica de agrupar primeiro, rotear depois. O agrupamento dos clientes é realizado com base em sua proximidade geográfica, enquanto que o roteamento consiste em resolver uma extensão do problema do caixeiro viajante com restrições de capacidade adicionais de coleta e entrega simultâneas (PCVCES). Mosheiov (1998) estudou o caso particular do PRVCES-Ho em que todas as demandas são iguais à unidade. O autor desenvolveu heurísticas construtivas com base no princípio clássico de primeiro rotear, depois agrupar, também denominado de particionamento do tour. Salhi e Nagy (1999) propuseram heurísticas construtivas para o PRVCE, que consiste basicamente em construir uma solução parcial, contendo somente clientes de entrega e em seguida inserir clientes de coleta segundo diferentes critérios de inserção até produzir uma solução completa. Dethloff (2001) desenvolveu heurísticas construtivas com base no princípio da inserção mais barata. As heurísticas propostas pelo autor utilizam diferentes medidas para avaliar o custo da inserção de um cliente, entre elas consideram-se tanto a distância percorrida como a capacidade utilizada na rota.

Duas heurísticas de busca local foram desenvolvidas por Montané e Galvão (2002). A primeira delas é uma adaptação da heurística de particionamento do tour proposta por Beasley (1983) enquanto que a segunda segue o princípio de varredura de Gillet e Miller (1974). Mais recentemente, Nagy e Salhi (2005) desenvolveram uma heurística de busca local, que admite soluções com um certo grau de inviabilidade. Inicialmente, a heurística proposta admite rotas inviáveis, sendo que durante a execução do algoritmo a viabilidade da rota é restaurada.

Dentre as metaheurísticas propostas para resolver o problema PRVCES-Ho vale destacar as seguintes. Vural (2003) desenvolveu dois métodos com base em Algoritmos Genéticos. Considera-se que esta é a primeira aplicação de uma metaheurística para o problema. Gökçe (2004) propôs uma metaheurística com base no método de Colônia de Formigas. Por outro lado, vários autores propuseram algoritmos com base no método de Busca Tabu. Gribkovskaia *et al.* (2006), consideraram o caso particular do PRVCES-Ho com apenas um veículo. Os autores propõem um modelo de programação inteira mista que admite que as coletas e entregas sejam realizadas tanto de forma mista como simultaneamente ao longo da rota. Montané e Galvão (2006) desenvolveram uma metaheurística para o PRVCES-Ho com vários veículos. O algoritmo utiliza uma vizinhança composta com base em três movimentos de busca local: realocação, intercâmbio e cruzamento. Os parâmetros tabu são ajustados no início do algoritmo e são mantidos fixos durante a execução do mesmo. Conjuntos de problemas com até 400 nós foram resolvidos. Os autores calcularam limites inferiores para medir a qualidade de seus resultados. Bianchessi e Righini (2007), propuseram uma metaheurística semelhante à anterior na qual se destaca: (i) o uso do princípio de particionamento do tour não consecutivo para o cálculo das soluções iniciais; (ii) a implementação de um movimento que combina triplas de rotas e (iii) o ajuste dinâmico dos parâmetros da busca tabu. O algoritmo proposto por Montané e Galvão (2006) é estendido em Montané (2006) mediante a incorporação de um mecanismo de busca adaptativa cujo objetivo é ajustar os parâmetros tabu de maneira dinâmica. Este mecanismo consiste em identificar padrões na trajetória de busca e em perturbar os valores dos parâmetros da busca tabu de acordo com o padrão observado.

Entre as técnicas metaheurísticas mais recentes aplicadas ao PRVCES-Ho destacam-se a metodologia GRASP-VND desenvolvida por Freitas *et al.* (2007); a técnica VNS-VND desenvolvida por Freitas, Arroyo e Montané (2007); o uso da técnica LNS (*Large Neighborhood Search*) utilizada por Ropke e Pisinger (2006); assim como a metodologia *record-to-record travel*, uma variante da técnica *simulated annealing*, aplicada por Chen e Wu (2006). Algoritmos exatos foram propostos por Angelelli e Mansini (2002) e Dell'Amico *et al.* (2005), fazendo uso da técnica *branch-and-price* para resolver a variante do PRVCES-Ho com janelas de tempo.

A literatura é escassa para problemas de roteamento de veículo com coleta e entrega e frota heterogênea, PRVCES-He. Em Galvão e Guimarães (1990), um caso particular de coleta e entrega simultânea com frota heterogênea é abordado no contexto de roteamento de helicópteros.

3. Metaheurística GRASP-VND

A metaheurística GRASP (*Greedy Random Adaptive Search Procedure*) (Feo e Resende, 1995) é um método de múltiplas partidas, na qual cada iteração consiste de duas fases: uma fase de construção, na qual uma solução inicial viável é construída utilizando um algoritmo guloso aleatorizado; e uma fase de busca local, na qual se procura melhorar a qualidade da solução inicial. A melhor solução calculada dentre todas as iterações é definida como resultado. No presente trabalho, propomos uma metaheurística GRASP-VND que utiliza a heurística *Variable Neighborhood Descent* (VND) (Mladenovic e Hansen, 1997) como método de busca local.

A Figura 1 ilustra o algoritmo da metaheurística GRASP-VND.

Procedimento GRASP-VND

início

1. $s^* \leftarrow \emptyset$;
2. $f(s^*) \leftarrow \infty$;
3. $I_{max} \leftarrow$ Número de iterações; // Critério de parada
4. $\alpha \leftarrow$ Número entre 0 e 1;
5. **para** $i \leftarrow 1$ até I_{max} **faça**
6. $s' \leftarrow$ PART_IMBEA (α); // fase de construção
7. $s'' \leftarrow$ VND(s'); // fase de busca local

```

8.   se  $f(s'') < f(s^*)$  então
9.      $s^* \leftarrow s''$ ;
10.  fim-se;
11.  fim-para;
12.  retorne  $s^*$ ;
fim GRASP-VND;

```

Figura 1 - Algoritmo da metaheurística GRASP-VND.

O critério de parada da metaheurística GRASP-VND é um número máximo de iterações (vide Figura 1, passo 3). Neste trabalho fixou-se o número máximo de iterações $I_{max} = 80$.

As fases de construção e busca local são descritas nas subseções 3.1 e 3.2, respectivamente.

3.1 Heurística de construção PART_IMBEA

A heurística construtiva PART_IMBEA, proposta no presente trabalho, consiste no uso de uma heurística de Particionamento do Tour (Beasley, 1983). Quando aplicadas aos problemas PRVCES estas heurísticas compreendem 3 etapas:

- 1) Construção de um *tour* inicial considerando o depósito e todos os clientes;
- 2) Formação de grupos de clientes pelo particionamento do *tour* inicial da Etapa 1;
- 3) Construção de rotas viáveis a partir dos grupos de clientes formados na Etapa 2.

Adotamos uma nova metodologia para realizar a Etapa 1, a qual é denominada de heurística IMBEA, Inserção Mais Barata com Escolha Aleatória. Por este motivo, descreveremos esta heurística com maior detalhamento.

Etapa 1. Construção de um Tour Inicial.- O *tour* inicial é construído utilizando a heurística de inserção IMBEA. A heurística IMBEA é uma adaptação da heurística clássica de Inserção Mais Barata descrita em detalhes em Goldbarg e Luna (2005). Na versão clássica os clientes são inseridos no *tour* inicial levando em consideração tanto a escolha do próximo cliente que será inserido como a posição de inserção no *tour* parcial. Os clientes são inseridos sempre na mesma ordem, começando por aqueles cuja inserção é mais barata. Em contrapartida, na heurística IMBEA, apenas a posição de inserção é levada em consideração, enquanto a escolha do próximo cliente que será inserido é realizada de maneira aleatória. Cada cliente é inserido na melhor posição de inserção, aquela que resulte no menor incremento no custo do *tour* parcial. Este procedimento permite que diferentes *tours* iniciais sejam gerados.

O processo de construção do *tour* inicial é aleatorizado. Primeiramente é escolhido aleatoriamente um cliente base i_B . A partir deste cliente aplica-se a heurística do Vizinho Mais Próximo (VMP) (Goldbarg e Luna, 2005) e constrói-se uma lista de clientes, chamada de lista candidatos LC. O tamanho de LC (LC_{Tam}) inicialmente é o mesmo número de clientes e a cada vez que insere um elemento na solução atualiza-se LC removendo o elemento inserido e diminuindo o seu tamanho. Com base em LC e usando um parâmetro α ($0 \leq \alpha \leq 1$) define-se um subconjunto restrito chamado de Lista de Candidatos Restritos ($LCR \subseteq LC$), de tamanho (LCR_{Tam}) igual a $\max(1, \alpha * LC_{Tam})$, formado pelos primeiros elementos que compõem a lista de candidatos. Em cada iteração, um cliente da lista LCR é escolhido aleatoriamente e é inserido no *tour* inicial em uma posição que causa menor custo de inserção, ou seja, busca-se, entre todas as posições do *tour*, a melhor posição para inserir o cliente escolhido. Este processo termina quando todos os clientes são inseridos no *tour* inicial.

A Figura 3 ilustra o processo de inserção de um cliente em um *tour* inicial parcial, usando um exemplo com 11 clientes. Na Figura 2(a) considera-se que 3 clientes fazem parte do *tour* inicial parcial, enquanto que os 8 clientes restantes se encontram na lista LC. A mesma figura mostra que o tamanho da lista LCR (LCR_{Tam}) foi definido com 4 clientes. A Figura 2(b) ilustra a escolha aleatória de um cliente da lista LCR (cliente 1) e a procura da melhor posição para inserir este cliente no *tour* parcial. A Figura 2(c) mostra a inserção do cliente 1 no *tour* na posição que causou menor custo de inserção (cliente 1 foi inserido após o cliente 8 e antes do cliente 3). A Figura 3 mostra o algoritmo de construção do *tour* inicial IMBEA.

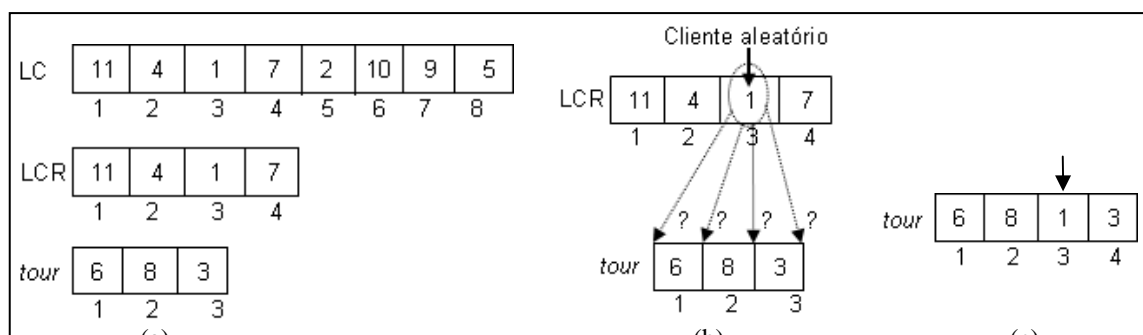


Figura 2 - Inserção de um cliente no *tour* inicial.

```

Procedimento IMBEA ( $\alpha$ , tour)
início
1. tour  $\leftarrow \emptyset$ ; // tour inicial
2.  $i_B \leftarrow$  cliente base(aleatório);
3. LC  $\leftarrow$  OrdenaCliente( $i_B$ ); // Ordenação pela heurística do Vizinho mais próximo
4. enquanto ( $LC_{Tam} > 0$ ) faça
5.    $LCR_{Tam} \leftarrow \text{Max}(1, \alpha * LC_{Tam})$ ;
6.    $i \leftarrow \text{Random}(1, LCR_{Tam})$ ; // escolhe um elemento de LC
7.   pos  $\leftarrow$  Busca_PosInsercao;
8.   tour[pos]  $\leftarrow$  tour  $\cup$  { $LC_i$ }; // adiciona na posição pos do tour o elemento  $LC_i$ 
9.   LC  $\leftarrow$  LC - { $LC_i$ }; // remove o elemento  $LC_i$  de LC
10.   $LC_{Tam} \leftarrow LC_{Tam} - 1$ ;
11. fim-enquanto;
12. retorne tour;
fim IMBEA;
  
```

Figura 3 – Construção do Tour Inicial Aleatorizada (IMBEA).

Etapa 2. Formação de grupos de clientes.- Os grupos de clientes são formados particionando o *tour* inicial, de maneira seqüencial, começando no primeiro cliente. Para garantir que a capacidade do veículo não será violada, a soma das demandas de coleta assim como a soma das demandas de entrega dos clientes em cada grupo não deve exceder a capacidade do veículo.

A Figura 4 ilustra o processo de particionamento do *tour*.

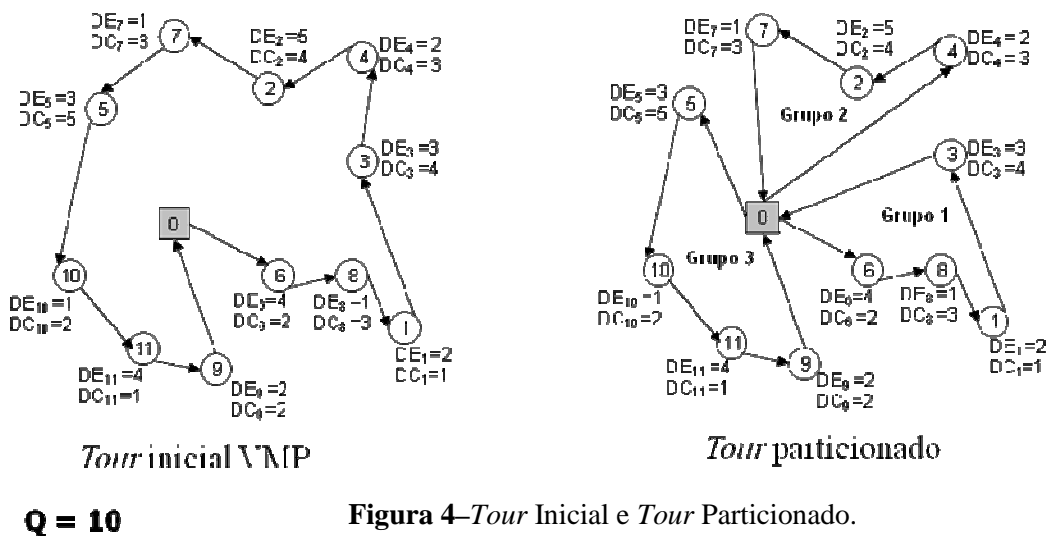


Figura 4–Tour Inicial e Tour Particionado.

Etapa 3. Construção de rotas viáveis.- Para cada grupo de clientes formado na Etapa 2, adota-se como ordenação inicial, àquela ordenação dada pelo *tour* inicial. Em princípio, esta ordenação dos clientes resulta em uma rota inviável com relação a capacidade do veículo. Nesta etapa é preciso calcular uma ordenação viável para

os clientes em cada grupo. Cada grupo de clientes é transformado em uma rota viável usando o método de viabilização de rotas ORD-VIAV que consiste atender primeiro os clientes com mais carga de entrega do que de coleta.

3.2 Busca local com base na heurística VND

A heurística VND (Mladenovic & Hansen, 1997) é um método de busca local que consiste em explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança. O método utiliza uma estrutura de vizinhança principal visando melhorar a solução corrente, quando não é mais possível, o método troca a vizinhança corrente por outra. O procedimento retorna a vizinhança principal quando uma melhor solução é encontrada. O algoritmo termina após utilizar todas as estruturas de vizinhança sem conseguir melhorar a solução corrente.

A Figura 5 mostra o algoritmo da heurística VND usada na fase de busca local da metaheurística GRASP-VND implementada para os problemas PRVCES-Ho e PRVCES-He.

Procedimento VND(s)	// s solução corrente
início	
1. $r \leftarrow 4$;	// número de vizinhanças
2. $k \leftarrow 1$;	// tipo da vizinhança corrente
3. enquanto ($k \leq r$) faça	
4. Encontre o melhor vizinho $s' \in N_k(s)$;	
5. se $f(s') < f(s)$ então	
6. $s \leftarrow s'$;	
7. $k \leftarrow 1$;	// primeira vizinhança
8. senão $k \leftarrow k + 1$;	// próxima vizinhança
9. fim-se ;	
10. fim-enquanto ;	
11. retorne s;	
fim VND ;	

Figura 5 – Algoritmo da heurística VND

Conforme Freitas *et. al.* (2007) este trabalho também implementa sete estruturas de vizinhança que podem ser utilizadas na fase de busca local (VND). Porém, as estruturas de vizinhanças são utilizadas de forma diferente daquela proposta por Freitas *et. al.* (2007), neste trabalho adota-se uma relação de precedência entre as estruturas de vizinhanças e utiliza-se um número menor de estruturas na fase de busca local (VND), pois utilizar muitas vizinhanças pode acarretar em acréscimo de tempo sem melhoria.

A relação de precedência dos movimentos foi estipulada de acordo com a abrangência de cada estrutura de vizinhança, onde as vizinhanças de menor abrangência devem ser aplicadas primeiro (Hansen & Mladenovic, 2003). A abrangência das vizinhanças foi dada através de testes realizados em uma instância de problema da literatura, onde para cada vizinhança enumerou-se o número de melhorias realizadas pela mesma em uma dada solução inicial e o número de soluções válidas encontradas. Desta forma as sete estruturas de vizinhança implementadas obtiverem a seguinte ordem de precedência: MCC (Movimento de Cruzamento Composto), MUSI (Movimento União de Seguintes Iguais), MCT (Movimento de Cruzamento Tradicional), 2-OPT, MT (Movimento de Troca), MI (Movimento de Intercâmbio) e MR (Movimento de Realocação) respectivamente.

Considerando que o uso de muitas estruturas de vizinhança pode acarretar em acréscimo de tempo sem melhoria, das sete estruturas de vizinhança implementadas, utilizamos no VND apenas quatro estruturas N_k , sendo $k = 1, 2, 3, 4$ de acordo com a seguinte relação de precedência dada aos movimentos: MCT, MI, MR e 2-OPT, respectivamente.

Deixou-se o movimento 2-OPT para ser aplicado no fim, não pelo motivo de ser o movimento mais abrangente e sim por que considera-se o mesmo como um movimento de intensificação ou melhoria. Cada estrutura de vizinhança utilizada no VND avalia todos os vizinhos em busca do melhor entre eles.

4. Metaheurística VNS-VND

A metaheurística de Busca em Vizinhança Variável, conhecido como VNS (*Variable Neighborhood Search*) foi proposta por Mladenovic e Hansen (1997). Esta metaheurística perturba sistematicamente a solução corrente utilizando para isso diferentes estruturas de vizinhança, o que pode resultar na degradação da solução. Este processo funciona como um mecanismo de diversificação da busca. Após o processo de perturbação da

solução é executado um processo de busca local que é tipicamente um mecanismo de intensificação da busca. Sempre que a busca local não consegue melhorar a melhor solução altera-se o mecanismo de perturbação utilizando outra estrutura de vizinhança. Pode-se considerar que a metaheurística VNS é uma extensão do método *Variable Neighborhood Descent* (VND) (Mladenovic & Hansen, 1997) descrito na seção 3.2, que admite soluções que pioram a solução corrente, permitindo desta forma escapar de ótimos locais. O VNS é baseado na exploração de um conjunto de K_{max} vizinhanças pré-definidas $\{N_1, \dots, N_{K_{max}}\}$.

Neste trabalho implementa-se uma variante da heurística VNS, conhecida como VNS-VND, que utiliza o método VND como procedimento de busca local. A metaheurística VNS-VND inicia com uma solução s obtida por uma heurística construtiva para o PRVCES, com base no princípio de particionamento do *tour*. A solução s é melhorada por uma busca local a qual é baseada na heurística VND, descrita anteriormente na Subseção 3.2.

A Figura 6 mostra o algoritmo na metaheurística VNS-VND proposta neste trabalho.

```
Procedimento VNS-VND  
inicio  
1.  $s \leftarrow \text{PART-Tour}()$ ; // solução inicial  
2.  $I_{max} \leftarrow 0$ ; // número máximo de iterações  
3. enquanto  $I_{max} < 80$  faça // critério de parada  
4.  $k \leftarrow 1$ ; // tipo de estrutura de vizinhança  
5. enquanto  $(k \leq r)$  faça  
6.  $s' \leftarrow$  Gere um vizinho qualquer de  $N_k(s)$ ; // Etapa de diversificação  
7.  $s'' \leftarrow \text{VND}(s')$ ; // Etapa de busca local  
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.  $I_{max} \leftarrow I_{max} + 1$ ;  
15. fim-enquanto;  
16. retorne  $s$ ;  
fim VNS-VND;
```

Figura 6 - Metaheurística VNS-VND aplicada ao PRVCES.

É importante observar que a solução s' gerada (*vide* Figura 6, Passo 6) é um vizinho qualquer da vizinhança $N_k(s)$, ou seja, a solução s' não necessariamente é melhor que a solução inicial s . Esta etapa do VNS realiza uma tarefa de diversificação da busca, com ela é possível evitar redundância de solução, com isto tende-se a aumentar a probabilidade de se escapar de ótimos locais. Nesta etapa de diversificação são utilizadas três estruturas de vizinhança N_k , para $k = 1, 2, 3$, são elas: Realocação (MR), União de Seguintes Iguais (MUSI) e 2-OPT, respectivamente, conforme movimentos para gerar vizinhanças descritos em Freitas *et. al.* (2007). A etapa de busca local (VND) da metaheurística VNS-VND utilizou as mesmas vizinhanças: MCT, MI, MR e 2-OPT utilizadas pelo VND da metaheurística GRASP-VND, mostrada anteriormente na Subseção 3.2. O Critério de parada da VNS-VND é um número máximo de iterações ($I_{max} = 80$).

5. Resultados computacionais

Nesta seção analisa-se o comportamento e desempenho das metaheurísticas VNS-VND e GRASP-VND propostas no presente trabalho. Estes algoritmos foram codificados em Linguagem C++, usando o compilador C++ Builder 6.0 e executados em um computador AMD Turion 64, 797 MHz, com 1 GB de RAM e sistema operacional Microsoft Windows XP. Os algoritmos VNS-VND e GRASP-VND são testado na resolução dos problemas PRVCES-Ho e PRVCES-He. No caso do PRVCES-Ho, utilizam-se dois conjuntos de dados: Dethloff (2001) que contém 40 instâncias de 50 clientes; e Salhi e Nagy (1999) com 14 instâncias entre 50 e 199 clientes. No caso do PRVCES-He, não foi possível encontrar na literatura conjuntos de dados teste para este problema específico. Por este motivo, foi gerado um conjunto de dados próprio. Os conjuntos de dados de Salhi e Nagy (1999) e Dethloff (2001) foram adaptados acrescentando-se dois tipos de veículos (tipo B e C) adicionais de maior capacidade que o tipo original (tipo A) considerado nas instâncias para o problema PRVCES-Ho. As

capacidades dos veículos, tipo B e C, foram definidas como sendo 20% e 44% maior que a capacidade do veículo tipo A, respectivamente.

Para o PRVCES-Ho, a qualidade dos resultados obtidos pelos algoritmos VNS-VND e GRASP-VND foi avaliada pela comparação com os algoritmos de Dethloff (2001), Salhi e Nagy (1999) e Montané e Galvão (2006). No caso do PRVCES-He os resultados dos algoritmos VNS-VND e GRASP-VND foram comparados entre si. Não foi possível avaliar a qualidade destes resultados por não terem sido encontrados na literatura algoritmos para este problema específico.

Foi utilizado um conjunto de 108 problemas teste para avaliar os algoritmos propostos: 40 de Dethloff (2001), 14 de Salhi e Nagy (1999) e mais 54 problemas adaptados.

5.1 Comparação dos resultados para o PRVCES-Ho

A Tabela 1 mostra a comparação entre resultados obtidos pelas metaheurística VNS-VND e GRASP-VND ao resolverem as instâncias de Dethloff (2001). Neste conjunto de dados, observa-se que ambas as heurísticas propostas produziram resultados de boa qualidade e bastantes competitivos. Vale observar, no entanto, que o VNS-VND teve em média uma ligeira vantagem em termos de qualidade da solução produzida, que foi 0,03% melhor que o GRASP-VND. Mais importante ainda é o fato de que embora a diferença percentual seja pequena, o VNS-VND produziu soluções melhores em 19 das 40 instâncias testadas com relação ao GRASP-VND, e obteve empate em outras três instâncias. Já o tempo computacional foi superior no caso de VNS-VND.

Tabela 1 – Comparação entre as metaheurísticas VNS-VND e GRASP-VND para o PRVCES-Ho: Instâncias de Dethloff (2001).

<i>n</i> = 50	VNS-VND			GRASP-VND		
	Média de 5 execuções			Média de 5 execuções		
	NR	FO. Sol	TS	NR	FO. Sol	TS
G-SCA3	3,9	679,70	10,60	4	680,50	2,90
G-SCA8	9,4	1046,40	4,30	9,5	1045,80	1,10
G-CON3	3,9	567,60	10,10	4	569,10	3,10
G-CON8	9,2	790,20	4,00	9,4	789,60	1,70
Média Geral		770,97			771,23	

A Tabela 2 mostra a comparação entre resultados obtidos pela heurística VNS-VND e os algoritmos de Dethloff(2001) e Montané e Galvão (2006) ao resolverem as instâncias de Dethloff (2001). A metaheurística VNS-VND é em média 8,51% melhor que o método de Dethloff (2001) e com relação aos melhores resultados da literatura (Montané e Galvão, 2006) a VNS-VND produziu resultados que em média foram 0,88% piores. No entanto, a VNS-VND foi capaz de calcular 12 novas melhores soluções e 1 empate. Com relação aos tempos computacionais, embora se trate de máquinas diferentes, o VNS-VND exigiu maior tempo computacional.

Tabela 2 – Desempenho do VNS-VND em relação a algoritmos da literatura para o PRVCES-Ho: Instâncias de Dethloff (2001).

<i>n</i> = 50	Dethloff (2001)			Montané -Galvão, (2006)			VNS-VND Média de 5 execuções		
	NR	FO. Sol	TS	NR	FO. Sol	TS	NR	FO. Sol	TS
G-SCA3	-	746,57	-	4	674,16	3,40	3,9	679,70	10,60
G-SCA8	-	1166,43	-	9,2	1044,35	4,11	9,4	1046,40	4,30
G-CON3	-	597,27	-	4	564,17	3,44	3,9	567,60	10,10
G-CON8	-	860,59	-	9,1	774,31	3,96	9,2	790,20	4,00
Média Geral		842,72			764,25			770,97	

A Tabela 3 mostra a comparação entre resultados obtidos pelas metaheurísticas GRASP-VND e VNS-VND ao resolverem as instâncias de Salhi e Nagy (2001). Neste conjunto de dados de maior porte, a heurística GRASP-VND perde ainda mais em qualidade quando comparada com o VNS-VND. O VNS-VND foi em média 2,27% melhor que o GRASP-VND. Observa-se, porém, um aumento significativo no tempo computacional do VNS-VND.

Tabela 3 – Comparação entre as metaheurísticas VNS-VND e GRASP-VND para o PRVCES-Ho: Instâncias de Salhi e Nagy (1999).

<i>n</i>	VNS-VND			GRASP-VND		
	Média das 5 execuções			Média das 5 execuções		
	NR	FO. Sol	TS	NR	FO. Sol	TS
G-50	3	484	5,55	3	478	1,67
G-75	6	702	15,92	7	712	6,95
G-100	5,5	706	69,05	5,5	715	30,87
G-120	4	865	142,04	5	889	51,70
G-150	7	882	190,77	8	923	105,26
G-199	11	1090	392,63	11	1128	172,66
Média Geral		776			794	

A Tabela 4 mostra a comparação entre resultados obtidos pela metaheurística VNS-VND e os algoritmos Salhi e Nagy (1999), Dethloff (2001) e de Montané e Galvão (2006) ao resolverem as instâncias de Salhi e Nagy (2001). Neste conjunto de dados de maior porte, a metaheurística VNS-VND foi em média 27,20% melhor que o método de Salhi e Nagy(1999) e 14,54% melhor que o algoritmo de Dethloff (2001). Com relação aos melhores resultados da literatura (Montané e Galvão, 2006) a VNS-VND produziu resultados que em média foram 0,26% melhores, sendo capaz de calcular 5 novas melhores soluções e 2 empates no conjunto de 14 problemas testados. Com relação aos tempos computacionais, embora se trate de máquinas diferentes, é evidente o maior esforço computacional exigido pela heurística VNS-VND.

Tabela 4 – Desempenho do VNS-VND em relação a algoritmos da literatura para o PRVCES-Ho: Instâncias de Salhi e Nagy (1999).

<i>n</i>	Salhi-Nagy (1999)			Dethloff (2001)		Montané e Galvão (2006)			VNS-VND		
	NR	FO. Sol	TS	NR	FO. Sol	NR	FO. Sol	TS	Média da 5 execuções		
	NR	FO. Sol	TS	NR	FO. Sol	NR	FO. Sol	TS	NR	FO. Sol	TS
G-50	6	602	3,00	3	501	3	471	4,05	3	484	5,55
G-75	12	914	1,50	7	782	7	698	7,26	6	702	15,92
G-100	10	888	3,58	5	831	6	701	12,02	6	706	69,05
G-120	11	1500	3,50	4	1015	5	905	18,11	4	865	142,04
G-150	15	1178	4,30	7	1050	7	878	29,07	7	878	206,34
G-199	19	1493	12,85	11	1348	11	1091	53,86	11	1090	392,63
Média Geral	-	1066	-	-	908	-	778	-	-	776	-

5.2 Comparações dos resultados para o PRVCES-He

Na Tabela 5 é possível observar que a metaheurística VNS-VND quando aplicada às instâncias de Dethloff (2001) para o caso PRVCES-He obtém soluções de melhor qualidade, sendo em média 0,89% melhor que o GRASP-VND. A VNS-VND foi consistentemente superior ao GRASP-VND em quase todos os casos, encontrando 34 melhores soluções e 2 empates no conjunto de 40 problemas teste. O VNS-VND continua exigindo maior esforço computacional que o GRASP. É importante mencionar que a frota de veículos calculadas pelas metaheurísticas VNS-VND e GRASP-VND ao resolver as instâncias de Dethloff (2001) foi praticamente a mesma em todos as instâncias. Em apenas 7 casos, a heurística VNS-VND utilizou um veículo a menos.

Tabela 5 – Comparação entre as metaheurísticas VNS-VND e GRASP-VND para o PRVCES-He: Instâncias de Dethloff modificadas.

<i>n</i> = 50	VNS-VND			GRASP-VND		
	Média de 5 execuções			Média de 5 execuções		
	FV	FO. Sol	TS	FV	FO. Sol	TS
G-SCA3	1C 1B 1A	632,08	10,70	1C 1B 1A	635,60	3,43

Os resultados da Tabela 6 reiteram a superioridade de heurística VNS-VND sobre a heurística GRASP-VND quando aplicadas em instâncias de maior porte (instâncias de Salhi e Nagy, 1999) para o caso PRVCES-He. Neste caso a metaheurística VNS-VND foi em média 2,09% melhor que a GRASP-VND, e encontrou as melhores soluções em todos as 14 instâncias. No entanto, vale observar que o tempo computacional requerido pela metaheurística VNS-VND é em média 73,25% superior ao requerido pela metaheurística GRASP-VND.

Tabela 6 – Comparação entre as metaheurísticas VNS-VND e GRASP-VND para o PRVCES-He: Instâncias de Salhi e Nagy modificadas

<i>n</i>	VNS-VND			GRASP-VND		
	Média de 5 execuções			Média de 5 execuções		
	FV	FO. Sol	TS	FV	FO. Sol	TS
G-50	1C 1B 1A	461,30	11,48	1C 1B 1A	468,70	2,67
G-75	2C 2B 1A	635,40	27,78	2C 2B 1A	646,00	11,46
G-100	2C 2B 1A	661,15	80,51	2C 2B 1A	668,70	41,33
G-120	1C 1B 1A	757,10	165,35	1C 1B 1A	763,70	132,90
G-150	2C 2B 2A	817,00	248,22	2C 2B 2A	846,70	83,81
G-199	3C 3B 2A	985,80	452,66	3C 3B 3A	1022,60	302,09
Média Geral		711,27			726,44	

6. Conclusões

Neste artigo foram propostas as metaheurísticas VNS-VND e GRASP-VND para gerar boas soluções para os problemas de Roteamento de Veículos com Coleta e Entrega Simultânea com Frota Homogênea (PRVCES-Ho) e Frota Heterogênea (PRVCES-He). O objetivo é determinar o conjunto de rotas que deverão ser percorridas pelos veículos minimizando os custos de transporte, dados pela distância total das rotas. As heurísticas propostas foram testadas em um conjunto de 108 instâncias teste, com $50 \leq n \leq 199$ clientes. A qualidade das soluções obtidas foi avaliada pela comparação com as soluções obtidas por algoritmos propostos na literatura.

Em geral, para todas as instâncias testadas e para ambos os problemas de coleta e entrega simultânea, a metaheurística VNS-VND obteve resultados bastante competitivos e superiores aos resultados da metaheurística GRASP-VND. Isto se explica pelo fato da metaheurística VNS-VND utilizar, em sua fase de diversificação, estruturas de vizinhança diferentes daquelas utilizadas na fase de busca local (VND). A metaheurística VNS-VND, no entanto, exige um maior esforço computacional. Quando comparada com os melhores resultados da literatura (Montané e Galvão, 2006), a metaheurística VNS-VND produziu soluções de boa qualidade, sendo capaz de produzir 17 novas melhores soluções e 3 empates, de um total de 54 instâncias teste. Uma grande parte das melhores soluções encontradas pela VNS-VND foi para problemas de porte médio, tamanho maior que 100 clientes.

Referências Bibliográficas

- Angelelli E. e Mansini R.** (2002), A Branch and Price Algorithm for a Simultaneous Pick-up and Delivery Problem. *Rapporto Tecnico 2002-01-24 del Dipartimento di Elettronica per l'Automazione, Università degli Studi di Brescia, Brescia.*
- Beasley, J.E.** (1983), Route First-Cluster Second Methods for Vehicle Routing. *Omega, 11, 403-408.*
- Bianchessi, N. e Righini, G.** (2007), Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research, 34, 578-594.*
- Chen, J. F. e Wu, T. H.** (2006), Vehicle routing problem with simultaneous deliveries and pickups. *Journal of the Operational Research Society, 57, 579-587.*
- Dell'Amico, M., Righini, G. e Salanim, M.** (2006), A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science, 40, 235-247.*
- Dethloff, J.** (2001), Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR Spectrum, 23, 79-96.*
- Feo, T.A.; Resende, M.G.C.** (1995), Greedy randomized adaptive search procedures. *Journal of Global Optimization 6, 109-133.*
- Freitas, L. M. B.; Arroyo, J. E. C.; Montané, F. A. T.; Vianna, D. S.** (2007), Heurística GRASP para o problema de Roteamento de Veículos com Coleta e Entrega Simultânea. *XXXIX SBPO - Simpósio Brasileiro de Pesquisa Operacional, Fortaleza.*
- Freitas, L. M. B.; Arroyo, J. E. C.; Montané, F. A. T.** (2007), Heurística VNS para o problema de Roteamento de Veículos com Coleta e Entrega Simultânea. *X SPOLM – Simpósio de Pesquisa Operacional e Logística da Marinha, Rio de Janeiro.*
- Galvão, R.D.; Guimarães, J.** (1990), The control of helicopter operations in the Brazilian oil industry: Issues in the design and implementation of a computerized system. *European Journal of Operational Research, v. 49, n. 2, 266-270.*
- Gillet B.E. e Miller L.R.** (1974), A heuristic for the vehicle dispatch problem. *Operations Research, 22, 340-349.*
- Gökçe, E. I.** (2004), A revised ant colony system approach to vehicle routing problems. *Dissertação de Mestrado, Graduate School of Engineering and Natural Sciences, Sabanci University, Turquia.*
- Goldbarg, M.C. e Luna, H.P.L.** (2005), Otimização Combinatória e Programação Linear: Modelos e Algoritmos. *Campus.*
- Gribkovskaia, I., Halskau, Ø., Laporte, G. e Vlcek, M.** (2006), General solutions to the single vehicle routing problem with pickups and deliveries. *European Journal of Operational Research, In Press.*
- Hansen, P., Mladenovic, N.** (2003), A Tutorial on Variable Neighborhood Search. *Les Cahiers du Gerad, ISSN: 0711-2440.*
- Min, H.** (1989), The multiple vehicle routing problem with simultaneous delivery and pickup points. *Transportation Research A, 23, 377-386.*
- Mladenovic, N.; Hansen, P.** (1997), Variable neighborhood Search. *Computers and Operations Research, v. 24, 1097 – 1100.*
- Montané, F. A. T.** (2006), Um Algoritmo de Busca Tabu Adaptativo aplicado a Problemas de Roteamento de Veículos com Pedidos de Coleta e Entrega. *XXXVIII SBPO Simpósio Brasileiro de Pesquisa Operacional, Goiânia.*
- Montané, F.A.T. e Galvão, R.D** (2002), Vehicle routing problems with simultaneous pick-up and delivery service. *Journal of the Operational Research Society of India, 39, 19-33.*
- Montané, F.A.T. e Galvão, R.D** (2006), A Tabu Search Algorithm for the Vehicle Routing Problem with Simultaneous Pick-up and Delivery Service. *Computers & Operations Research, 33, 595-619.*
- Mosheiov, G.** (1998), Vehicle routing with pick-up and delivery: tour partitioning heuristics. *Computer and Industrial Engineering, 34, 669-684.*
- Nagy, G. e Salhi, S.** (2005), Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research, 162, 126-141.*
- Ropke, S.; Pisinger, D.** (2006), A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research, 171, 750-775.*



Salhi, S. e Nagy, G. (1999), A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society*, 50, 1034-1042.

Vural, A. V. (2003), A GA based meta-heuristic for capacitated vehicle routing problem with simultaneous pick-up and deliveries. *Dissertação de Mestrado, Graduate School of Engineering and Natural Sciences, Sabanci University, Turquia.*