



SPOLM 2007

ISSN 2175-6295

Rio de Janeiro- Brasil, 08 e 09 novembro de 2007.

HEURÍSTICA VNS PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA E ENTREGA SIMULTÂNEA

Lia Mara Borges de Freitas

Universidade Candido Mendes – Campos
liamarabf@yahoo.com.br

José Elias Cláudio Arroyo

Departamento de Informática – Universidade Federal de Viçosa
jarroyo@dpi.ufv.br

Fermín Alfredo Tang Montané

Universidade Candido Mendes – Campos
tang@ucam-campos.br

Resumo

Neste artigo é proposto uma heurística VNS (*Variable Neighborhood Search*) 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. Para gerar soluções iniciais do problema foram testadas duas heurísticas construtivas e como mecanismo de busca local é usada a heurística VND (*Variable Neighborhood Descent*) que explora sete estruturas de vizinhanças. A heurística VNS é testada em problemas com $50 \leq n \leq 199$ clientes. A qualidade das soluções obtidas é avaliada pela comparação com as soluções obtidas por três métodos propostos na literatura.

Palavras Chave: Heurísticas; VNS; problema de roteamento de veículos.

Abstract

In this article is proposed a VNS (*Variable Neighborhood Search*) heuristic to solve approximately 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. To generate initial solutions we tested two constructive heuristics and we used a local search mechanism based on the Variable Neighborhood Descent (VND) heuristic that explores seven neighborhood structures. The proposed heuristic is tested on problems with $50 \leq n \leq 199$ clients. The quality of the obtained solutions is evaluated by the comparison with the solutions obtained by three methods from the literature.

Keywords: Heuristics; VNS; Vehicle routing problem; Metaheuristics.

1. INTRODUÇÃO

O Problema de Roteamento de Veículos com Coleta e Entrega Simultânea (PRVCES) é uma variante do clássico problema de roteamento de veículos. Este problema existe em diferentes empresas que realizam transportes de mercadoria/passageiros. Por exemplo, a Petróleo Brasileiro S/A (PETROBRAS) da bacia de Campos, hoje a maior empresa brasileira que atua na exploração e produção de petróleo e seus derivados no Brasil e no exterior, realiza o abastecimento das unidades marítimas ou plataformas a partir do único porto de Imbetiba e mediante uma frota de embarcações. Através de estudos realizados pela empresa, constatou-se que algumas tarefas poderiam ser otimizadas. Dentre essas tarefas, destaca-se a otimização das rotas que as embarcações devem percorrer partindo do porto de Imbetiba para atender a demanda de materiais das plataformas.

Esta otimização pode vir a diminuir, em muito, o tempo gasto no trajeto dos materiais para as plataformas e porto, o que pode representar uma economia significativa para a empresa.

Os pedidos solicitados pelas plataformas podem ser os seguintes:

- Somente entrega de materiais – os pedidos feitos para a embarcação são somente para desembarque de materiais nas plataformas.
- Coleta e Entrega de material – os pedidos solicitados para a embarcação são embarque e/ou desembarque nas plataformas, podendo ocorrer de maneira simultânea ou não.
- Somente recolhimento de material – esse caso é o menos utilizado, porém dependendo do tipo de material a ser recolhido das plataformas, será necessária a alocação de uma embarcação somente para o recolhimento de tais materiais.

O presente trabalho aborda o caso de coleta e entrega simultânea de materiais, onde a demanda de uma plataforma (cliente) é composta de cargas de coleta e entrega. Os pedidos de entrega são atendidos pelas embarcações (veículos) a partir do porto de Imbetiba (depósito), enquanto os pedidos de coleta são conduzidos até o porto no final do percurso. A frota de embarcações é heterogênea, ou seja, as embarcações são de diferentes tipos.

O PRVCES é um problema NP-difícil, ou seja, este tipo de problema, em geral, é intratável por natureza. A solução de instâncias do problema de grande porte através de algoritmos exatos é inviável (Min, 1989). Portanto, na literatura tem sido usados métodos baseados em metaheurísticas para solucionar tais problemas (Montané & Galvão, 2006; Vianna *et al.*, 1999). As metaheurísticas são métodos que visam encontrar soluções aproximadas através de procedimentos de busca em vizinhanças que objetivam estender o processo de busca para além de um mínimo local.

A literatura apresenta um grande número de metaheurísticas para problemas de otimização, tais como: Busca Tabu, *Simulated Annealing*, Algoritmos Genéticos, VNS, VNS, dentre outras.

Este trabalho propõe solucionar o PRVCES através do uso da heurística VNS, tendo por objetivo minimizar a distância percorrida pelos veículos. Os resultados obtidos até o momento demonstram a eficiência da heurística proposta quando comparada com os melhores resultados da literatura para o problema de roteamento de veículo com coleta e entrega simultânea (Dethloff, 2001; Montané & Galvão, 2006).

O presente artigo encontra-se organizado da seguinte forma: na Seção 2 é apresentado, em detalhes, o problema abordado e a literatura relacionada. Na Seção 3 descreve-se a heurística VNS implementada neste trabalho. Os resultados computacionais são apresentados na Seção 4 e a Seção 5 apresenta as conclusões finais.

2. PROBLEMA ABORDADO

Existe uma grande variedade de Problemas de Roteamento de Veículos. O problema clássico de roteamento de veículos (PRV) é definido da seguinte maneira: veículos com capacidade fixa Q devem sair do depósito ($i = 0$) e atender as demandas q_i ($i=1, \dots, n$) de n clientes. Conhecendo a distância d_{ij} entre os clientes i e j ($i, j = 0, \dots, n$), o objetivo é minimizar a distância total percorrida pelos veículos, de tal maneira que apenas um veículo atenda a

demanda de um dado cliente e que a capacidade Q dos veículos não seja violada. Este problema considera apenas um tipo de demanda, entrega ou coleta.

O PRVCES é uma extensão deste tipo de problema e pode ser definido como segue: dado um conjunto de n clientes, cada qual com uma demanda de coleta C_i e de entrega E_i ; e um depósito com um conjunto de m veículos com capacidade Q . Desta forma temos:

- n : número de clientes que podem demandar entrega e/ou coleta de materiais.
- Q : Capacidade de cada veículo.
- C_i : Demanda de coleta do cliente i .
- E_i : Demanda de entrega do cliente i .

Cada veículo deve sair do depósito ($i = 0$) e atender demandas C_i e E_i ($i=1, \dots, n$) dos clientes de acordo com sua capacidade e conhecendo a distância d_{ij} entre os clientes i e j ($i, j = 0, \dots, n$).

O objetivo é determinar o conjunto de rotas que deverão ser percorridas pelos veículos, minimizando o custo total transporte, dado pela distância total das rotas, e atendendo as seguintes condições:

- Cada rota começa e termina no depósito.
- Todo cliente i é visitado somente uma vez por somente um veículo.
- A demanda total de qualquer rota não deve superar a capacidade do veículo que a atende.

Uma característica importante do problema é que a carga de um veículo em qualquer rota é uma mistura de pedidos de coleta e entrega.

Apresenta-se uma breve revisão da literatura relacionada ao problema abordado.

O Problema de Roteamento de Veículos com Pedidos de Coleta e Entrega (PRVCE) considera que cada cliente faz pedidos de coleta e/ou entrega, mas não ambos simultaneamente. Este problema tem recebido bastante atenção na literatura (Mosheiov, 1994; Mosheiov, 1998; Salhi e Nagy, 1999).

No caso do PRVCES que é uma generalização do problema anterior a literatura é escassa. Entre os trabalhos existentes na literatura pode-se citar: Min (1989) estudou sobre esta classe de problemas no contexto de um sistema de distribuição para bibliotecas públicas, propondo uma heurística clássica de agrupar primeiro e rotear depois; Dethloff (2001) propôs uma heurística de construção com base no princípio da inserção mais barata; Nagy e Salhi (2005) propuseram uma heurística de busca local que admite soluções com certo grau de inviabilidade; Montané e Galvão (2006) desenvolveu um algoritmo de busca tabu para o problema; e Angelelli e Mansini (2002) desenvolveu um algoritmo exato, usando um algoritmo de *branch-and-price* para o problema com janelas de tempo.

Um caso particular de coleta e entrega simultânea com frota heterogênea é abordado em (Galvão & Guimarães, 1990), este trabalho considera restrições sobre o número de clientes em cada rota, assim como restrições sobre a distância máxima entre dois clientes adjacentes.

3. HEURÍSTICA VNS PARA PRVCES

O método VNS é uma heurística moderna proposta por Mladenovic e Hansen (1997) e é baseada na exploração de um conjunto de k_{max} vizinhanças pré-definidas, $\{N_1, \dots, N_{k_{max}}\}$. Esta heurística gera um vizinho $x' \in N_k(x)$ da solução corrente x , executa uma busca local a partir desta solução obtendo uma solução ótima local x'' . Se esta solução x'' melhora a solução corrente x , o processo reinicia a partir desta solução (i.e. $x = x''$). Caso contrário, a estrutura de vizinhança é alterada de forma sistemática. A heurística VNS e suas extensões têm sido aplicadas satisfatoriamente para resolver vários problemas de otimização combinatória.

Neste artigo, a heurística VNS inicia com uma solução x obtida por uma heurística construtiva para o PRVCES. A solução x é melhorada por uma busca local a qual é baseada na heurística VND. Na Figura 1 é apresentado o pseudocódigo a heurística VNS.

```

inicio
1.  $x \leftarrow \text{Heurística-Contrutiva}()$ ;
2. Enquanto o critério de parada não seja satisfeito faça
3.    $k \leftarrow 1$ ; //Tipo de estrutura de vizinhança
4.   Enquanto ( $k \leq r$ ) faça
5.     Gere um vizinho qualquer  $x' \in N_k(x)$ ; //vizinhança  $k$ 
6.      $x'' \leftarrow \text{VND}(x')$ ; //Busca Local
7.     se  $f(x'') < f(x)$  então
8.        $x \leftarrow x''$ ;
9.        $k \leftarrow 1$ ;
10.    senão  $k \leftarrow k + 1$ ;
11.  fim-enquanto;
12. fim-enquanto;
13. Retorne  $x$ ;
fim VNS;

```

Figura 1: Algoritmo VNS.

As heurísticas construtivas testadas, para gerar a solução inicial para o PRVCES, são descritas na subseção 3.1. A busca local VND e as estruturas de vizinhas usadas são descritos na subseção 3.2.

3.1 HEURÍSTICAS CONSTRUTIVAS

Heurísticas construtivas são procedimentos que, a partir de uma regra de prioridade, constroem passo a passo uma solução viável do problema. A utilização destas heurísticas se dá em problemas em que o tempo computacional é muito curto ou como ponto de partida para algoritmos de melhoria.

As heurísticas de construção implementadas neste trabalho são Varredura (Gillett e Miller, 1974) e Particionamento de Tour (Beasley, 1983) que estão descritas nas subseções 3.1.1 e 3.1.2, respectivamente.

3.1.1. Heurística de Varredura

A heurística de Varredura procura obter uma solução em duas etapas distintas:

- Primeiramente agrupam-se, de maneira seqüencial, os pontos de demanda segundo o critério de proximidade (os clientes são agrupados com base em suas coordenadas polares) e satisfazendo a capacidade de cada veículo, formando assim *clusters*. O conjunto de *clusters* é formado inicialmente girando um raio centrado no depósito (Laporte *et al.*, 2000).
- Na segunda etapa é solucionado cada *cluster* independentemente criando uma rota para estes clientes (Goldbarg e Luna, 2000). Cada *cluster* é transformado em uma rota válida para o problema.

A seguir são apresentados os passos do algoritmo da varredura aqui implementado:

Passo 1. Calcular os ângulos e os raios entre os clientes i ($i=1, \dots, n$) de acordo com suas coordenadas polares, considerando que o sistema de coordenadas tem o depósito como origem e um cliente base escolhido aleatoriamente.

Passo 2. Para cada quadrante, que é delimitado pelas coordenadas do depósito, ordenar os clientes com base em suas coordenadas polares de forma que os clientes com menor ângulo, com relação ao cliente base (que inicialmente é o cliente 1), correspondam às primeiras posições. Em caso de empate entre dois clientes escolhe-se o cliente com menor raio. Os clientes ordenados no primeiro quadrante adquirem as primeiras posições e assim consecutivamente até o quarto quadrante.

Passo 3. Construir uma nova rota, seguindo os seguintes passos:

- (a) Incluir um novo cliente no *cluster* enquanto a soma das demandas dos clientes deste *cluster* não ultrapasse a capacidade do veículo atribuído a ele, ou seja, a soma das demandas de coleta tem que ser menor ou igual à capacidade

do veículo e a soma das demandas de entrega tem que ser menor ou igual à capacidade do veículo.

(b) Resolver um PCVCES com os clientes do *cluster* atual, obtendo-se a rota viável.

Passo 4. Repetir o Passo 3, caso existam clientes que não foram roteados. Caso contrário, guardar a solução atual e ir ao passo 5.

Passo 5. Repetir os passos 3 e 4 para uma nova ordenação de clientes (tendo por base sucessivamente os clientes $2,3,\dots,n$).

Passo 6. O algoritmo termina após considerar as n ordenações possíveis.

3.1.2. Heurística de Particionamento de Tour

Assim como a Varredura, a heurística de Particionamento de Tour também procura obter a solução em duas etapas distintas: grupamento e roteamento.

O grupamento dos clientes é realizado a partir de um *tour* de caixeiro viajante (de menor distância). Depois de constituídos os grupos de clientes o roteamento de cada grupo é realizado resolvendo-se subproblemas do caixeiro viajante capacitados com serviço de coleta e entrega simultâneo, de maneira que a restrição de capacidade seja satisfeita. Este procedimento é repetido para diferentes nós iniciais com a finalidade de melhorar a qualidade da solução obtida.

A heurística de Particionamento de Tour é descritas através dos seguintes passos:

Passo 1. Resolver um PCV considerando todos os clientes e o depósito.

Passo 2. Particionar, de maneira sequencial, o *tour* obtido no Passo 1, começando em um cliente arbitrário e visando a formação de grupos de clientes tal que cada grupo satisfaça a restrição de capacidade do veículo. A soma das demandas dos clientes do grupo tem que ser menor ou igual à capacidade do veículo estabelecida ao grupo.

Passo 3. Para cada grupo de clientes formado no passo 2, resolve-se um problema do caixeiro viajante capacitados com serviço de coleta e entrega simultâneos (PCVCES). As rotas viáveis obtidas em cada grupo constituem uma solução viável para o PRVCES.

Passo 4. Repetir os passos 2 e 3 começando o processo em um cliente diferente aos já considerados. Compara-se a solução com a melhor solução encontrada até o momento e guarda-se a melhor.

3.2. BUSCA LOCAL BASEADA 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 isto 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. No presente trabalho é implementada uma heurística VND convencional (*vide* Mladenovic & Hansen, 1997) na qual são utilizados os seguintes movimentos para gerar as vizinhanças:

- *Movimento 2-OPT*: O movimento 2-Opt consiste na troca de dois arcos, não consecutivos e pertencentes a uma mesma rota por outros dois arcos que permitem restabelecer a conectividade da rota.
- *Movimento de troca (MT)*: Este movimento consiste em trocar a posição de um par de clientes na mesma rota.

- *Movimento de realocação (MR)*: Este movimento consiste em remover um cliente de uma rota e inseri-lo em outra rota diferente, podendo reduzir, em alguns casos, o número de rotas existentes.
- *Movimento de intercâmbio (MI)*: Um movimento de intercâmbio consiste na troca de clientes entre duas rotas, ou seja, um cliente pertencente a uma rota é trocado com outro de outra rota.
- *Movimento de cruzamento tradicional (MCT)*: Este movimento, descrito em (Kindervater & Savelsbergh, 1997), consiste na troca de seqüências ordenadas de clientes (um ou mais clientes) entre duas rotas. Para isso, cada rota original é dividida em dois segmentos S1 e S2 (inicial e final), e trocam-se os segmentos finais entre as duas rotas (*vide* Figura 2).

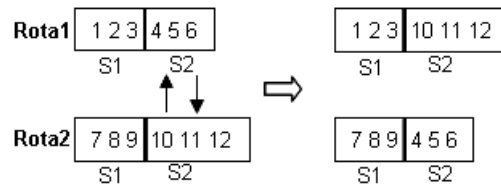


Figura 2 - Movimento de Cruzamento

- *Movimento de cruzamento composto (MCC)*: É um novo movimento proposto no presente trabalho como extensão do movimento de cruzamento tradicional. Cada rota é dividida em dois segmentos S1 e S2 (inicial e final). Em uma primeira etapa troca-se o segmento S1 com o segmento S2 em cada rota. Logo após, aplica-se o cruzamento tradicional (*vide* Figura 3).

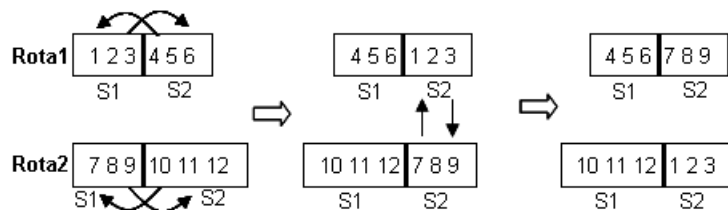


Figura 3 - Movimento de Cruzamento

- *Movimento união de segmentos iguais (MUSI)*: É um outro movimento de cruzamento entre duas rotas proposto neste trabalho. Após a divisão de cada rota em dois segmentos S1 e S2 (inicial e final), uma nova rota é formada unindo-se ambos os segmentos S1 (iniciais), enquanto que outra nova rota é formada unindo-se ambos os segmentos S2 (finais). Um exemplo deste movimento é mostrado na Figura 4.

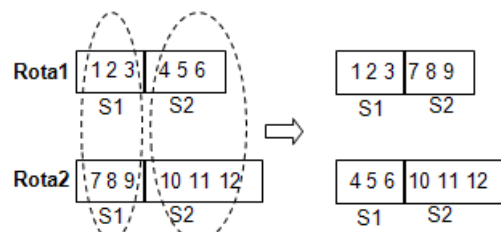


Figura 4 - Movimento de Cruzamento

Os movimentos descritos acima geram 7 estruturas de vizinhança diferentes. As estruturas de vizinhanças são denotadas por N_1, N_2, \dots, N_7 de acordo com a seguinte relação de precedência dada aos movimentos que as definem: *MCT, MCC, MI, MR, 2-OPT, MT* e *MUSI*, respectivamente. Assim, o movimento *MCT* é o mais freqüentemente usado, enquanto que o movimento *MUSI* é o menos freqüente. Em cada estrutura de vizinhança N_k são avaliados todos os vizinhos em busca do melhor entre eles.

Na Figura 5 é mostrado o pseudocódigo da heurística VND usada na fase de busca local da heurística VNS.

Procedimento VND(s)	
Início	
1.	Seja s_0 uma Solução Inicial
2.	$s \leftarrow s_0$; // Solução corrente
3.	$r \leftarrow 7$; // Número de vizinhanças consideradas
4.	$k \leftarrow 1$; // Tipo da vizinhança corrente
5.	enquanto ($k \leq r$) faça
6.	Encontre o melhor vizinho $s' \in N_k(s)$;
7.	se $f(s') < f(s)$
8.	então $s \leftarrow s'$;
9.	$k \leftarrow 1$; // primeira vizinhança
10.	senão $k \leftarrow k + 1$; // próxima vizinhança
11.	fim-se;
12.	fim-enquanto;
13.	Retorne s ;
Fim VND;	

Figura 5: Heurística VND – Busca Local do VNS

4. RESULTADOS COMPUTACIONAIS

Nesta seção analisa-se o comportamento e desempenho da heurística VNS proposta neste trabalho. Este algoritmo foi codificado em Linguagem C, usando o compilador C++ Builder 6.0 e executado em uma máquina AMD Athlon(tm) 64, 2.20 GHz, 1 GB de RAM com sistema operacional Microsoft Windows XP.

A heurística VNS é avaliada utilizando dois conjuntos de dados propostos para o PRVCS por Salhi e Nagy (1999) e Dethloff (2001), respectivamente. O primeiro conjunto compreende 14 instâncias entre $n = 50$ e $n = 199$ clientes, enquanto que o segundo conjunto compreende 40 instâncias com $n = 50$ clientes. A qualidade dos resultados obtidos pela heurística VNS foi avaliada pela comparação com os métodos de Salhi e Nagy (1999), Dethloff (2001) e Montané e Galvão (2006).

Ao realizar uma comparação entre as duas heurísticas construtivas implementadas, Varredura e Particionamento de Tour, para as 14 instâncias de Salhi e Nagy (1999) foi observado que em média a heurística de Particionamento de Tour é 78% melhor que a Varredura. Por este motivo, escolheu-se a heurística de Particionamento para gerar as soluções iniciais para a heurística VNS.

A Tabela 1 mostra a comparação dos resultados obtidos para o conjunto de instâncias do Dethloff (2001). Nesta Tabela compara-se a heurística VNS com os métodos propostos por Dethloff (2001) e Montané e Galvão (2006). Para este conjunto de instâncias, a heurística VNS proposta produziu resultados de boa qualidade e bastantes competitivos. Em média, a VNS foi 6,10% melhor que o método proposto por Dethloff (2001) e apenas 3,92% pior que o método de Montané e Galvão (2006). É importante mencionar que a heurística VNS foi capaz de encontrar 5 novas melhores soluções as quais não foram obtidas pela método proposto por Montané e Galvão (2006).

Dethloff (2001) não disponibiliza o número de rotas (NR) de cada solução e os tempos computacionais (TS) gastos na execução do seu algoritmo. Vale ressaltar que os tempos computacionais da heurística VNS foram bem menores que os tempos da Busca Tabu de Montané e Galvão (2006).

Tabela 1 – Comparação de métodos para resolver as instâncias de Dethloff (2001).

n	Instâncias	Dethloff			Montané e Galvão			VNS		
		NR	FO. Sol	TS	NR	FO. Sol	TS	NR	FO. Sol	TS
50	SCA3-0	-	689,00	-	4	640,55	3,37	4	645	0,59
50	SCA3-1	-	765,60	-	4	697,84	3,25	4	704	0,48
50	SCA3-2	-	742,80	-	4	659,34	3,52	4	660	1,25
50	SCA3-3	-	737,20	-	4	680,04	3,31	4	693	1,01
50	SCA3-4	-	747,10	-	4	690,50	3,43	4	692	1,23
50	SCA3-5	-	784,40	-	4	659,90	3,67	4	683	0,92
50	SCA3-6	-	720,40	-	4	653,81	3,35	4	674	0,82

50	SCA3-7	-	707,90	-	4	659,17	3,33	4	732	1,03
50	SCA3-8	-	807,20	-	4	719,47	3,40	3	769	0,78
50	SCA3-9	-	764,10	-	4	681,00	3,41	4	690	0,62
50	SCA8-0	-	1132,90	-	9	981,47	4,14	9	1033	0,73
50	SCA8-1	-	1150,90	-	9	1077,44	4,27	10	1142	0,73
50	SCA 8-2	-	1100,80	-	10	1050,98	4,20	10	1048	1,12
50	SCA 8-3	-	1115,60	-	9	983,34	4,17	10	1034	0,51
50	SCA 8-4	-	1235,40	-	9	1073,46	4,13	10	1102	0,57
50	SCA 8-5	-	1231,60	-	9	1047,24	4,02	9	997	0,95
50	SCA 8-6	-	1062,50	-	9	995,59	3,85	9	998	0,79
50	SCA 8-7	-	1217,40	-	10	1068,56	4,22	10	1148	0,45
50	SCA 8-8	-	1231,60	-	9	1080,58	3,85	9	1094	0,71
50	SCA 8-9	-	1185,60	-	9	1084,80	4,20	10	1090	1,04
50	CON3-0	-	672,40	-	4	631,39	3,64	4	649	0,59
50	CON3-1	-	570,60	-	4	554,47	3,31	4	572	0,53
50	CON3-2	-	534,80	-	4	522,86	3,45	4	542	0,64
50	CON3-3	-	656,90	-	4	591,19	3,28	4	600	0,86
50	CON3-4	-	640,20	-	4	591,12	3,47	4	642	0,43
50	CON3-5	-	604,70	-	4	563,70	3,38	3	640	0,57
50	CON3-6	-	521,30	-	4	506,19	3,32	3	550	0,17
50	CON3-7	-	602,80	-	4	577,68	3,51	3	751	0,2
50	CON3-8	-	556,20	-	4	523,05	3,66	4	526	0,64
50	CON3-9	-	612,80	-	4	580,05	3,36	4	600	1,15
50	CON8-0	-	967,30	-	9	860,48	4,19	9	919	1,15
50	CON8-1	-	828,70	-	9	740,85	3,89	9	832	0,34
50	CON8-2	-	770,20	-	9	723,32	3,76	9	717	0,25
50	CON8-3	-	906,70	-	10	811,23	4,12	10	856	0,93
50	CON8-4	-	876,80	-	9	772,25	3,75	10	831	0,17
50	CON8-5	-	866,90	-	9	756,91	3,99	9	772	0,34
50	CON8-6	-	749,10	-	9	678,92	4,04	9	673	1,03
50	CON8-7	-	929,80	-	9	814,50	4,00	9	894	0,78
50	CON8-8	-	833,10	-	9	775,59	3,74	9	735	0,9
50	CON8-9	-	877,30	-	9	809,00	4,13	9	841	0,62
Média			842,72			764,25			794,25	

n – Número de Clientes, **NR** – Número Rotas, **FO. Sol** – Valor da Função Objetivo da Solução, **TS** – Tempo em Segundos

A Tabela 2 mostra os resultados obtidos para o conjunto de problemas de maior porte gerados por Salhi e Nagy (1999). As soluções obtidas pela heurística VNS são comparadas com as soluções geradas pelos métodos de Salhi e Nagy (1999), Dethloff (2001) e Montané e Galvão (2006). Neste conjunto de problemas, novamente, a heurística VNS obteve soluções de boa qualidade. Observa-se que, em média, o VNS é 30,31% melhor que o método do Salhi e Nagy (1999) e 11% melhor que o método do Dethloff (2001). A heurística VNS proposta só perde para o algoritmo de Busca Tabu de Montané e Galvão (2006), sendo em média 5,14 % pior. Vale mencionar que para este conjunto de dados o tempo computacional também foi bastante menor que os tempos da Busca Tabu de Montané e Galvão (2006).

Tabela 2 – Comparação dos métodos para resolver as instâncias de Salhi e Nagy (1999).

<i>n</i>	Instâncias	Salhi e Nagy			Dethloff			Montané e Galvão			VNS		
		NR	FO. Sol	TS	NR	FO. Sol	TS	NR	FO. Sol	TS	NR	FO. Sol	TS
50	CMT1x	6	601	3	3	501	-	3	472	3,73	3	472	0,62
50	CMT1y	5	603	3	3	501	-	3	470	4,37	3	516	0,93
75	CMT2x	11	903	1,7	7	782	-	7	695	6,91	6	707	1,64
75	CMT2y	12	924	1,3	7	782	-	7	700	7,61	7	756	0,93
100	CMT3x	10	923	2,3	5	847	-	5	721	11,04	5	740	4,34
100	CMT3y	10	923	2,3	5	847	-	5	719	12,01	5	732	5,34
100	CMT12x	10	831	4,9	6	804	-	6	675	12,23	6	710	3,96
100	CMT12y	11	873	4,8	5	825	-	6	689	12,80	6	747	2,90

120	CMT11x	11	1500	3,4	4	959	-	4	900	18,17	5	933	17,23
120	CMT11y	11	1500	3,6	4	1070	-	5	910	18,04	5	948	2,78
150	CMT4x	15	1178	4,3	7	1050	-	7	880	24,60	8	938	7,15
150	CMT4y	15	1178	4,3	7	1050	-	7	878	29,07	8	942	9,14
199	CMT5x	19	1509	12,8	11	1348	-	11	1098	51,50	11	1177	11,68
199	CMT5y	19	1477	12,9	11	1348	-	10	1083	56,21	12	1131	25,75
Média		1066		908		778		818					

n – Número de Clientes, *NR* – Número Rotas, *FO. Sol* – Valor da Função Objetivo da Solução, *TS* – Tempo em Segundos

5. CONCLUSÕES

Neste artigo foi proposta uma heurística VNS para gerar soluções aproximadas boa qualidade para o problema de Roteamento de Veículos com Coleta e Entrega Simultânea (PRVCES). O objetivo é determinar o conjunto de rotas que deverão ser percorrido pelos veículos minimizando o custo total de transporte, dado pela distância total das rotas.

A heurística foi testada em um conjunto de 54 instâncias teste, com $50 \leq n \leq 199$ clientes. A qualidade das soluções obtidas foi avaliada pela comparação com as soluções obtidas por três métodos propostos na literatura.

Baseado nos resultados obtidos, conclui-se que a heurística VNS produziu soluções de boa qualidade. A heurística obteve resultados bastante competitivos no conjunto de instâncias com até 50 clientes, onde em média de mostrou ser 6,10% melhor que método proposto por Dethloff (2001) e apenas 3,92% pior que o método de Montané e Galvão (2006). Já no conjunto de instâncias entre 50 e 199 clientes, em média, o VNS foi 30,31% melhor que o algoritmo do Salhi e Nagy (1999) e 11% melhor que o algoritmo do Dethloff (2001), perdendo somente para o algoritmo de Busca Tabu de Montané e Galvão (2006) em 5,14 %.

É importante observar que tanto para o conjunto de instâncias com até 50 clientes como no conjunto de instâncias entre 50 e 199 clientes o tempo computacional da heurística VNS foi bem menor que os tempos da Busca Tabu de Montané e Galvão (2006).

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ANGELELLI, 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.
- [2] BEASLEY, J.E. (1983) Route First-Cluster Second Methods for Vehicle Routing. *Omega*, v. 11, n. 4, 403-408.
- [3] DETHLOFF, J. (2001) Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR Spectrum*, v. 23, n. 1, 79-96.
- [4] FEO, T.A.; RESENDE, M.G.C. (1995) Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6, 109-133.
- [5] 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.
- [6] GILLET, B.E.; MILLER, L.R. (1974) A heuristic algorithm for the vehicle dispatch problem. *Operations Research*, v. 22, n. 2, 340-349.
- [7] GOLDBARG, M.C.; LUNA, H.P. (2000) Otimização Combinatória e Programação Linear: Modelos e Algoritmos. *Editora Campus*
- [8] KINDERVATER, G.A.P. E SAVELSBERGH, M.W.P. (1997) Vehicle Routing: Handling Edge Exchanges, em E.H.L. Aarts, J.K. Lenstra (Eds.), *Local Search in Combinatorial Optimization*, Wiley, Chichester, 337-360.
- [9] MIN, H. (1989) The multiple vehicle routing problem with simultaneous delivery and pickup points. *Transportation Research A*, v. 23, n. 5, 377-386.
- [10] MLADENOVIC, N.; HANSEN, P. (1997) Variable neighborhood Search. *Computers and Operations Research*, v. 24, 1097 – 1100.

- [11] MONTANÉ, F. A.T.; GALVÃO, R. D. (2006) A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers and Operations Research*, v. 33, n. 3, 595-619.
- [12] MOSHEIOV, G. (1994) The traveling salesman problem with pick-up and delivery. *European Journal of Operational Research*, v. 79, n. 2, 299-310.
- [13] MOSHEIOV, G. (1998) Vehicle routing with pick-up and delivery: Tour partitioning Heuristics. *Computers and Industrial Engineering*, v. 34, n. 3, 669-684.
- [14] NAGY, G.; SALHI, S. (2004) Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. Article in Press *European Journal of Operational Research*, Available online at www.sciencedirect.com.
- [15] SALHI, S.; NAGY, G. (1999) A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society*, v. 50, 1034-1042.
- [16] VIANNA, D. S.; OCHI, L. S.; DRUMMOND, L. M. A. (1999) A parallel hybrid evolutionary metaheuristic for the period vehicle routing problem with heterogeneous fleet. *Lecture Notes in Computer Science*, 1388, 216-225.