Uma heurística baseada em *Iterated Local Search* para o Problema de Roteamento de Veículos com Múltiplos Depósitos

Mário Henrique de Paive Perché[1],
Anand Subramanian[1],
Pablo Luiz Araújo Munhoz[1],
Luiz Satoru Ochi[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
Rua Passo da Pátria, 154 – Bloco E, 3° andar – CEP 24.210-240 – Niterói (RJ), Brasil

{mperche,anand,pmunhoz,satoru}@ic.uff.br

Resumo. *Este trabalho aborda o Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD), uma generalização do Problema de Roteamento de Veículos (PRV), em que mais de um depósito é considerado. Cada veículo é associado a uma rota que inicia e termina em um mesmo depósito. O objetivo do PRVMD é minimizar a soma dos custos de viagem. Para resolver o PRVMD, foi proposto um algoritmo Multi-Start que faz uso do procedimento Variable Neighborhood Descent, com variações aleatórias das estruturas de vizinhança, integrado à metaheurística Iterated Local Search. O algoritmo proposto foi avaliado usando instâncias de referência e seu desempenho foi comparado com outras abordagens encontradas na literatura. Os resultados obtidos mostram que a abordagem desenvolvida é bastante competitiva.*

PALAVRAS CHAVE. *Problema de Roteamento de Veículos com Múltiplos Depósitos. Metaheurísticas. Iterated Local Search. MH - Metaheurísticas.*

Abstract. *This work deals with the Multi-Depot Vehicle Routing Problem (MDVRP), a generalization of the well-known Vehicle Routing Problem (VRP), in which more than one depot is considered. Each vehicle is associated with a route that starts and ends at the same depot. The objective of the MDVRP is to minimize the sum of the travel costs. In order to solve the MDVRP, we propose a Multi-Start algorithm which makes use of a Variable Neighborhood Descent, with random neighborhood ordering, integrated into a Iterated Local Search (ILS) framework. The proposed algorithm was evaluated using benchmark instances and its performance was compared with other solution approaches reported in the literature. The results obtained illustrates that the developed ILS approach is quite competitive.*

KEYWORDS. *Multi-Depot Vehicle Routing Problem. Metaheuristics. Iterated Local Search. MH - Metaheuristics.*

1 Introdução

Muitos trabalhos relacionados ao Problema de Roteamento de Veículos (PRV) têm sido publicados desde o trabalho de **Dantzig** e **Ramser** (1959). Segundo **Ballou** (2001), o custo de transporte corresponde de um a dois terços dos custos logísticos de uma empresa. Tal fato torna a busca por uma solução de boa qualidade ainda mais motivante. O PRV possui diversas variantes, onde são considerados tipos diferentes de veículos, tempo limite de entrega ou coleta dos produtos, mais de um depósito, entre outras. Essa última variante citada, abordada neste trabalho, é denominada Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD). Esse problema pode ser exemplificado com o caso em que uma empresa possui vários depósitos, que juntos, devem atender a demanda total de clientes, uma vez que não há restrição em relação a qual depósito atenderá a demanda de um cliente específico. Porém, se os clientes estão vinculados a um depósito, o problema de distribuição deve ser modelado como um conjunto de PRVs independentes.

Formalmente, o PRV pode ser definido com base em um grafo $G = (V, E)$, onde $V = \{v_1, v_2, ..., v_m\} \cup V_0$ é um conjunto de vértices e $E = \{e_1, e_2, ..., e_n\}$ um conjunto de arestas. Cada vértice $v_i \in V - V_0$ representa um cliente a ser atendido, sendo que $v_0 \in V_0$ representa o depósito. Por sua vez, cada aresta $(i, j) \in E$ é associada a um custo não-negativo c_{ij}, normalmente a distância entre 2 vértices. É importante ressaltar que todas as rotas têm o depósito (v_0) como ponto de partida e de chegada além de incluir um subconjunto de arestas de E'. Cada cliente tem uma demanda $q \geq 0$ a ser atendida por algum dos r veículos inicialmente estacionados no depósito e só pode ser visitado apenas uma vez. A frota de veículos é homogênea, ou seja, uma constante Q define a capacidade dos veículos. O PRV consiste em determinar um conjunto de rotas de modo a minimizar a soma dos custos atribuídos às arestas de E'. Além disso, ressalta-se que para cada rota, a capacidade Q do veículo associado deve ser respeitada. O PRVMD é uma generalização do PRV, onde o conjunto de vértices V pode ser definido por $V = \{v_1, v_2, ..., v_m\} \cup V_0$ onde $V_0 = \{v_{01}, v_{02}, ..., v_{0d}\}$ são os depósitos. Uma rota i pode ser definida por $R_i = \{d, v_1, v_2, ..., v_n, d\}$, com $d \in V_0$ e $n \leq m$. O custo de uma rota pode ser calculado como no PRV clássico.

O PRVMD pode ser reduzido para o PRV para o caso em que exista apenas um depósito, logo, a sua complexidade é também definida como NP-Difícil. A utilização de métodos exatos para resolução dessa classe de problemas tende a se tornar ineficiente à medida em que a dimensão do problema cresce. Uma vez que o tempo computacional para resolver o problema via um método exato pode ser bastante elevado, opta-se por utilizar procedimentos heurísticos, que embora não garatam uma solução ótima para o problema são capazes de resolvê-lo em um tempo computacional relativamente inferior ao demandado pelos métodos exatos.

Um algoritmo *multi-start* que consiste em utilizar o procedimento *Variable Neighborhood Descent*, com variação aleatória das estruturas de vizinhança, integrada à metaheurística *Iterated Local Search* é apresentado neste trabalho com o intuito de resolver o PRVMD. São descritas as estruturas de vizinhança utilizadas bem como a construção da solução inicial. Foram utilizadas instâncias encontradas na literatura, sendo os resultados obtidos pelo algoritmo proposto, comparados com os melhores resultados encontrados até então.

O presente trabalho está dividido em 5 seções, incluindo esta introdução onde o

PRVMD é contextualizado. Na Seção 2 é apresentada uma breve descrição de trabalhos encontrados na literatura que se relacionam com o desenvolvido nesse projeto. Na Seção 3 a metodologia utilizada neste trabalho é demonstrada. Na Seção 4 os resultados são apresentados e analisados, e, por fim, na Seção 5 são feitas conclusões sobre a metodologia adotada.

2 Trabalhos Relacionados

Devido à importância prática dos problemas de roteamento e o desafio de resolvê-los eficientemente, várias técnicas de solução têm sido relatadas na literatura. Em virtude de o PRVMD ser um problema altamente combinatório e pertencer à classe NP-difícil, métodos exatos comumente não são capazes de resolvê-los em tempo computacional viável, justificando sua abordagem por metodologias heurísticas, que não garantem uma solução ótima, mas geram soluções de boa qualidade.

Tillman (1969) foi o primeiro a abordar o PRVMD, quando propôs um algoritmo baseado na heurística de economias de **Clarke** e **Wright** (1964). Algoritmos exatos, baseado no método *Branch-and-Bound*, foram propostos em 1980 por **Laporte** et al. (1984) e **Laporte** et al. (1988), onde o primeiro foi capaz de resolver casos com até 50 clientes e 8 depósitos, enquanto o último conseguiu resolver problemas com até 80 clientes e 8 depósitos. Recentemente, uma abordagem geral exata, proposta por **Baldacci** e **Mingozzi** (2009), foi concebida para lidar com PRVs diversos, incluindo o PRVMD.

Alguns algoritmos para o PRVMD, baseados em heurísticas clássicas do PRV, foram propostas entre o início dos anos 70 e início de 1980. **Tillman** e **Hering** (1971) introduziram uma abordagem *look ahead* à heurística de **Tillman** (1969). **Tillman** e **Cain** (1972) também extenderam o algoritmo de **Tillman** (1969) incluindo um procedimento de enumeração parcial para computar as melhorias ao unir clientes a uma mesma rota. **Wren** e **Holliday** (1972) apresentaram um algoritmo que emprega um procedimento de varredura para gerar as rotas, que por sua vez, são refinadas por um conjunto de heurísticas. **Gillett** e **Johnson** (1976) aplicaram o algoritmo clássico de varredura de **Gillett** e **Miller** (1974) para construir uma solução inicial que é posteriormente refinada por um processo que reatribui clientes à diferentes depósitos. **Golden** et al. (1977) propuseram duas abordagens heurísticas, onde a primeira é baseada no método das economias, enquanto a segunda é um procedimento ”primeiro agrupar, depois rotear”. **Raft** (1982) sugeriu um algoritmo modular que decompõe o problema em subproblemas resolvendo-os separadamente e então agrupa as soluções por meio de um processo iterativo.

Como algumas outras variantes do PRV, algoritmos para o PRVMD inspirados em metaheurísticas começaram a surgir na década de 1990. **Chao** et al. (1993) desenvolveram um algoritmo que faz uso de um procedimento *record-to-record* para melhorar a solução inicial gerada utilizando a heurística de economias de **Golden** et al. (1977). **Renaud** et al. (1996) implementaram uma heurística Busca Tabu composta por três fases: a melhoria rápida, a intensificação e diversificação. **Cordeau** et al. (1997) sugeriram um algoritmo Busca Tabu combinado com um mecanismo de GENI, que consiste em um procedimento de inserção geral originalmente desenvolvido pela **Gendreau** et al. (1992) para o Problema do Caixeiro Viajante (PCV).

Pisinger e **Ropke** (2007) desenvolveram o algoritmo *Adaptive Large Neighborhood Search* (ALNS), que consiste em um conjunto de heurísticas de remoção e

inserção, escolhidas em cada iteração via método de seleção da roleta com base no desempenho anterior dessas heurísticas. Esse algoritmo foi aplicado em algumas variantes do PRV, incluindo o PRVMD.

3 Metodologia

Nesta seção é apresentada a metodologia utilizada para resolver o PRVMD.

3.1 Representação de uma solução

Uma solução para o PRVMD é representada por um vetor de rotas. Além de ser um dado de entrada para o problema, o número de veículos (r) é o mesmo para todos os depósitos. Cada rota é associada a um veículo, podendo conter 0, 1 ou mais clientes. Admitindo que o número de depósitos seja D, o total de rotas é dado por $r \times D$. Portanto, a cada r posições do vetor solução tem-se um depósito. A Figura 1 exemplifica uma configuração possível para uma solução com quatro depósitos e dois veículos disponíveis por depósito.

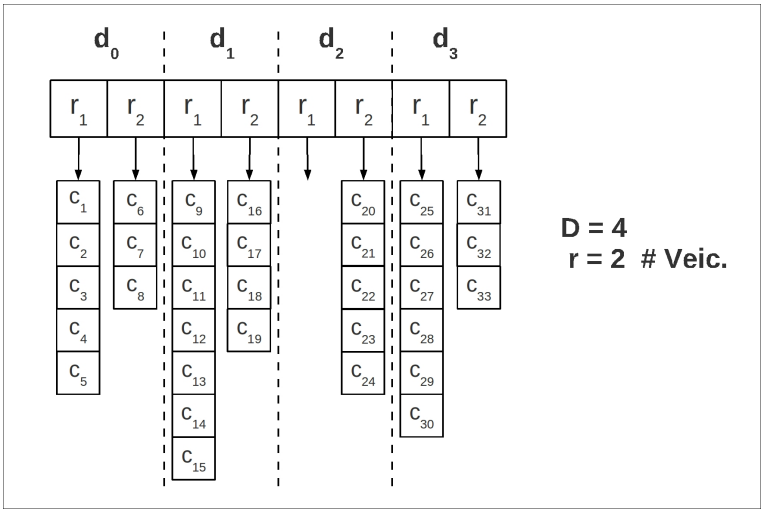

Figura 1. Representação de uma solução

3.2 Avaliação de uma solução

Uma solução do PRVMD é avaliada por uma função f apresentada na Equação (1) dada pela distância total percorrida pelos veículos de todos os depósitos.

$$f = \sum_{i,j \in (V \cup V_0)} c_{ij} \times x_{ij} \tag{1}$$

onde c_{ij} é a custo de viagem entre os cliente i e j quando $i, j \in V$; um cliente i e um depósito j quando $i \in V, j \in V_0$; e um depósito i e um cliente j quando $i \in V_0, j \in V$. A variável x_{ij} assume valor 1 se a aresta $(i, j) \in E$ estiver na solução, e assume o valor 0, caso contrário.

3.3 Geração de uma solução inicial

Seja r o número de veículos (ou rotas). Cada rota é preenchida com um cliente k, aleatoriamente selecionado de uma lista de candidatos (LC). Em seguida, os demais

clientes são avaliados de acordo com a Equação (2). Note que todas as vezes que um cliente é adicionado à uma solução parcial, a LC é atualizada e o custo de inserção dos demais clientes é recalculado.

$$g(k) = (c_{ik} + c_{kj} - c_{ij}) - \gamma(c_{0k} + c_{k0}) \tag{2}$$

O custo (c) de inserir um cliente $k \in LC$ em uma rota usanto critério de inserção mais barata modificado é expressado pela Equação (2). O valor de $g(k)$ é obtido pela soma de duas parcelas. A primeira representa o custo de inserir o cliente k entre todos os pares de clientes adjacentes i e j enquanto a segunda parcela corresponde a uma penalidade utilizada para evitar a inserção tardia de clientes muito distantes do depósito. O custo de ida e volta do depósito é ponderado por um fator γ, que é escolhido aleatoriamente dentro do intervalo $\{0.00, 0.05, 0.10, ..., 1.65, 1.70\}$. O cliente k que possuir o menor custo de inserção associado é então adicionado à solução. O procedimento guloso de construção termina quando todos os clientes forem adicionados à solução.

3.4 Estruturas de Vizinhança

Existem dois tipos de estruturas de vizinhança, a saber: Inter-Rotas e Intra-Rota. Quando há troca/realocação de um ou mais clientes pertencentes a uma rota r_1 para uma rota r_2 ou vice-versa, sendo $r_1 \neq r_2$, esse movimento é classifcado como Inter-Rotas. Após a execução do movimento, a posição inicial de um cliente $c_1 \in r_1$ pode, ou não, ser preservada em r_2. No caso em que os movimentos citados acima são realizados em uma mesma rota, alterando apenas a ordem de atendimento dos clientes, são classificados como Intra-Rota.

Para a exploração do espaço de soluções foram utilizadas nove estruturas de vizinhança do PRV clássico, admitindo somente movimentos viáveis, isto é, que não violam a capacidade máxima do veículo. Dentre elas, seis são do tipo Inter-Rotas: *Shift(1,0)* - um cliente c é transferido de uma rota r_1 para uma rota r_2; *Swap(1,1)* - permutação entre um cliente c_1 pertencente a uma rota r_1 e um cliente c_2 pertencente a uma rota r_2; *Shift(2,0)* - dois clientes adjacentes, c_1 e c_2, são transferidos de uma rota r_1 para uma rota r_2; *Swap(2,1)* - permutação entre dois clientes adjacentes, c_1 e c_2, pertencentes a uma rota r_1 e um cliente c_3 pertencente a uma rota r_2; *Swap(2,2)* - permutação entre dois clientes adjacentes, c_1 e c_2, pertencentes a uma rota r_1 com outros dois clientes adjacentes, c_3 e c_4, pertencentes a uma rota r_2; e, por fim, *Cross* - a aresta entre dois clientes adjacentes, c_1 e c_2, pertencentes a uma rota r_1 e outra aresta entre dois clientes adjacentes, c_3 e c_4, pertencentes a uma rota r_2 são removidos. Em seguida, duas novas arestas são inseridas, uma conectando o cliente c_1 ao cliente c_4, e outra conectando o cliente c_3 ao cliente c_2. E três são do tipo Intra-Rotas: *Or-opt* - um, dois ou três clientes adjacentes são removidos e inseridos em algum outro ponto da rota; *2-opt* - duas arestas não adjacentes são removidas, e outras duas são adicionadas de forma que seja gerada uma nova rota; e *Exchange* - permutação entre dois clientes, versão intra-rota da vizinhança *Swap(1,1)*. Um melhor entendimento de todas estruturas de vizinhança detalhadas acima é apresentado em **Subramanian** et al. (2010).

Como uma solução para o PRVMD não distingue as rotas de acordo com o respectivo depósito, as estruturas de vizinhança citadas acima são capazes de trocar/realocar um ou mais clientes entre rotas de depósitos diferentes. Porém, a troca/realocação de vários

clientes exige que os respectivos movimentos sejam aplicados uma série de vezes, o que na maioria das vezes não acontece. Logo, a varredura do espaço de busca fica comprometida. Com o intuito de percorrer melhor o espaço de busca, além dessas estruturas de vizinhança utilizam-se as 2 novas estruturas Inter-Rotas propostas (*ShiftDepot* e *SwapDepot*).

3.4.1 Estrutura de Vizinhança *ShiftDepot*

Uma rota r_i pertencente a um depósito d_k é transferida para um outro depósito d_l, desde que o depósito d_l possua, no mínimo, um veículo livre. A Figura 2 ilustra a aplicação do *ShiftDepot*, onde a rota r_1 pertencente ao depósito d_1 é realocada para o depósito d_0. Como o depósito d_0 possui uma rota livre, esse movimento é viável.

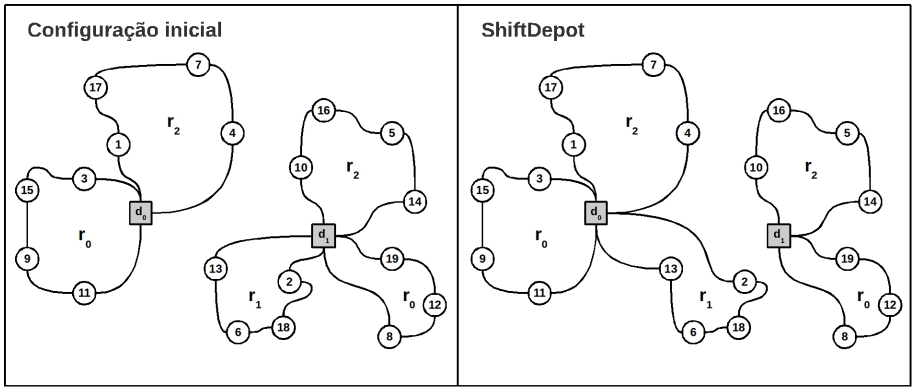

Figura 2. Estrutura de vizinhança *ShiftDepot*

3.4.2 Estrutura de Vizinhança *SwapDepot*

As rotas r_i e r_j pertencentes aos depósitos d_k e d_l, respectivamente, são trocadas de depósito. Na Figura 3 as rotas r_2 e r_1, pertencentes ao depósito d_0 e d_1, respectivamente, são trocadas. Observa-se que essa transição entre depósitos é sempre viável.

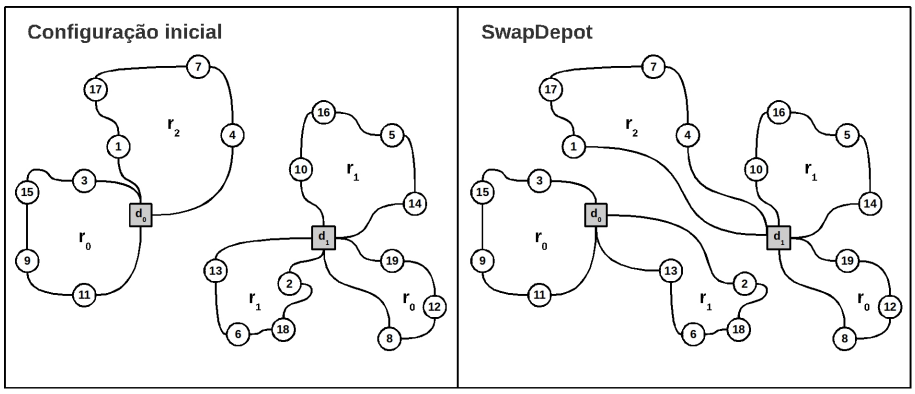

Figura 3. Estrutura de vizinhança *SwapDepot*

3.5 Algoritmo Proposto - ILS-RVND

O algoritmo proposto baseia-se em metaheurísticas, as quais, ao contrário dos procedimentos heurísticos clássicos, possuem mecanismos que possibilitam escapar dos

ótimos locais, ainda distantes de um ótimo global. Mais especificamente, propõe-se um algoritmo *multi-start* que consiste em utilizar o procedimento *Variable Neighborhood Descent* (VND) (**Mladenović** e **Hansen**, 1997), com variação aleatória das estruturas de vizinhança (RVND), integrada à metaheurística *Iterated Local Search* (ILS) (**Lourenço** et al., 2003), denominado ILS-RVND.

O pseudocódigo do ILS-RVND é apresentado na Figura 4. A heurística *multi-start* (linhas 3-19), é executada $iterMax$ vezes e faz o uso de um procedimento aleatório para construir a solução inicial. Desse modo, a diversidade de soluções iniciais geradas é garantida. Inicia-se então, o procedimento ILS (linhas 7-15). Como método de busca local do ILS utiliza-se o algoritmo RVND (linha 8), detalhado na subseção 3.5.1. Já na fase de diversificação do ILS, escolhe-se um tipo de perturbação a ser utilizada, dado um conjunto de perturbações (linha 13), a saber: *MultiSwap(1,1)* e *MultiShift(1,1)*. Definidas as fases de intensificação e diversificação o ILS é executado $iterMaxIls$ vezes.

Algoritmo 1: ILS-RVND($iterMax$, $iterMaxIls$)

```
1  início
2      s* = ∞;
3      para (i = 1, 2, ..., iterMax) faça
4          s ← ConstroiSI();
5          s' ← s;
6          iterIls ← 0;
7          enquanto (iterIls ≤ iterMaxIls) faça
8              s ← RVND(N(.), f(.));
9              se (f(s) < f(s')) então
10                 s' ← s;
11                 iterIls ← 0;
12             fim
13             s ← Perturbacao(s');
14             iterIls ← iterIls + 1;
15         fim
16         se (f(s') < f(s*)) então
17             s* ← s';
18         fim
19     fim
20     retorna s*;
21 fim
```

Figura 4. Algoritmo ILS-RVND

No mecanismo *MultiSwap(1,1)*, múltiplas aplicações randômicas do *Swap*$(1, 1)$ são feitas. Já no *MultiShift(1,1)*, múltiplas aplicações randômicas do *Shift(1,1)* são feitas. O movimento *Shift(1,1)* consiste em transferir um cliente k de uma rota r_1 para uma rota r_2 ao mesmo tempo em que um cliente l da rota r_2 é transferido para a rota r_1. Essa perturbação é mais "agressiva" se comparada à *MultiSwap(1,1)*, possibilitando um grande número de movimentos.

Salienta-se que ao obter uma solução de melhora (condição da linha 9 satisfeita) o ILS é reiniciado, ou seja, $iterIls$ é zerado (linha 11). Dando sequência ao método proposto, a solução obtida pelo ILS é comparada com a melhor solução encontrada até então (linha 16). Se a condição for satisfeita, atualiza-se a melhor solução (s^*). O algoritmo descrito até aqui integra a heurística *multi-start*, executada $iterMax$ vezes. Após a primeira iteração do algoritmo (ILS-RVND), utiliza-se a melhor solução s^* encontrada pelo ILS como guia,

ou seja, nas próximas iterações da heurística *multi-start* busca-se uma solução melhor do que a melhor solução encontrada nas n iterações anteriores.

3.5.1 Algoritmo RVND

O algoritmo RVND, apresentado na Figura 5, inicia armazenando as estruturas Inter-Rotas em uma lista de estruturas de vizinhança ($listaEstrtVizinhanca$). A seguir, a cada iteração do algoritmo, uma estrutura de vizinhança pertencente à $listaEstrtVizinhanca$ é selecionada de maneira aleatória (linha 4). Dada uma solução s, realiza-se uma busca local com base na estrutura de vizinhança escolhida. Quando a solução s' encontrada pela busca local é melhor do que s, uma estrutura de vizinhança Intra-Rota é aplicada à s' e a $listaEstrtVizinhanca$ é atualizada com todas as estruturas de vizinhança disponíveis (linhas 6-9). Caso contrário, a estrutura de vizinhança é removida da $listaEstrtVizinhanca$ (linha 11). Repetem-se os passos descritos entre as linhas 4 e 12 até que $listaEstrtVizinhanca$ não tenha mais nenhum elemento.

Algoritmo 2: RVND($N(.)$, $f(.)$), s

```
1  início
2      Inicializa listaEstrtVizinhanca;
3      enquanto (listaEstrtVizinhanca ≠ ∅) faça
4          N^(α) ← Selecione um elemento de listaEstrtVizinhanca;
5          s' ← MelhorVizinho(s, N^(α));
6          se (f(s') < f(s)) então
7              s ← s';
8              s ← IntraRota(s);
9          fim
10         senão
11             Remove N^(α) de listaEstrtVizinhanca;
12         fim
13     fim
14     retorna s
15 fim
```

Figura 5. Algoritmo RVND

4 Resultados Computacionais e Análises

Esta seção apresenta os resultados do ILS-RVND descrito na Seção 3.5. O algoritmo descrito neste trabalho foi implementado em C++. Os testes foram realizados em um computador i7 com 8GB de RAM e sistema operacional Ubuntu 9.10, kernel 2.6.18. Apesar da existência de 4 núcleos, o algoritmo não explora esta capacidade de multiprocessamento.

Foram consideradas 11 instâncias utilizadas por **Cordeau** et al. (1997) envolvendo até 360 clientes e 9 depósitos, onde não há restrição quanto à duração máxima de uma rota.

As instâncias podem ser obtidas no site *http://neo.lcc.uma.es/radi-aeb/WebVRP/*, e os limites superiores foram encontrados em **Pisinger** e **Ropke** (2007). **Baldacci** e **Mingozzi** (2009), através de uma abordagem exata encontrou algumas soluções ótimas das instâncias consideradas no teste.

A Tabela 1 compara as duas implementações do ILS-RVND, sendo que ILS-RVND II diferencia-se do ILS-RVND I por fazer o uso das duas estruturas de vizinhança *ShiftDepot* e *SwapDepot*. A primeira coluna indica o nome da instância; S^* é o valor da melhor solução encontrada pelo algoritmo; M_s representa a média dos valores das soluções encontradas; e T_m, o tempo computacional médio, em segundos, despendido para encontrar as soluções. Tanto a média das soluções encontradas, quanto o tempo computacional médio foram calculados após a realização de 30 execuções de cada um dos algoritmos para cada instância.

Tabela 1. ILS-RVND I $\times$ ILS-RVND II

	ILSRVND I			ILSRVND II		
	S^*	M_s	T_m	S^*	M_s	T_m
	576,87	576,87	2,36	576,87	576,87	2,48
p02	473,53	473,61	1,78	473,53	473,53	1,70
p03	641,19	641,19	5,55	641,19	641,19	5,14
p04	1001,59	1004,64	11,08	**1001,04**	**1004,37**	10,62
p05	750,03	751,61	10,76	750,03	**751,36**	9,79
p06	876,50	879,85	12,17	876,50	**879,11**	11,84
p07	881,97	**883,78**	10,25	881,97	884,14	9,83
p12	1318,95	1318,95	4,43	1318,95	1318,95	4,26
p15	2505,42	2506,78	37,36	2505,42	**2505,64**	33,97
p18	3702,85	3719,91	130,70	3702,85	**3717,00**	116,62
p21	5490,55	5516,60	479,62	**5474,84**	**5502,89**	424,55

Observa-se na Tabela 1, que em nove das onze instâncias consideradas, houve empate em relação à melhor solução encontrada, sendo que o ILS-RVND II superou o ILS-RVND I nas demais instâncias. Já em relação à média das soluções encontradas, o ILS-RVND II obteve melhores resultados em seis instâncias, empatando em quatro e perdendo, com apenas 0.04% de diferença, em uma delas. Esses resultados mostram que a inserção das duas estruturas de vizinhança, específicas para o PRVMD, tornaram o algoritmo mais robusto.

Na Tabela 2, a seguir, são apresentados os resultados do algoritmo ILS-RVND II comparados com os resultados da literatura de **Baldacci** e **Mingozzi** (2009), **Cordeau** et al. (1997) e **Pisinger** e **Ropke** (2007). Nesta tabela, a primeira coluna indica o nome da instância; S_{UB} representada os melhores resultados encontrados na literatura (*Upper Bound*) para o PRVMD antes da publicação do trabalho de **Pisinger** e **Ropke** (2007); Z representa os valores ótimos, ou os melhores valores encontrados por **Baldacci** e **Mingozzi** (2009); S^* representa o valor da melhor solução encontrada por cada algoritmo; M_s, a média das soluções encontradas; e T_m, o tempo computacional médio, em segundos, despendido para encontrar as soluções. As colunas GAP e Imp. apresentam os resultados obtidos por duas medidas de desempenho, definidas pelas Equações (3) e (4), respectivamente:

$$\text{GAP} = \frac{(Z - M_s^{ILS-RVNDII})}{Z} \tag{3}$$

$$\text{Imp.} = \frac{(s_{Lit} - s^*_{ILS-RVNDII})}{s_{Lit}} \tag{4}$$

onde Z é a solução encontrada pelo método exato (quando uma solução não é apresentada,

utiliza-se o melhor resultado encontrado na literatura); $M_s^{ILS-RVNDII}$ representa a média das soluções encontradas pelo algoritmo ILS-RVND II; s_{Lit} representa a melhor solução da literatura incluindo os resultados encontrados por **Pisinger** e **Ropke** (2007); e, por fim, $s^*_{ILS-RVNDII}$ representa a melhor solução encontrada pelo algoritmo ILS-RVND II.

Na primeira medida de desempenho é calculado o desvio das soluções médias do algoritmo proposto em relação a solução apresentada pelo método exato, desse modo, quanto menor o valor do desvio, menor a variabilidade das soluções médias finais e mais robusto é o algoritmo. Já na segunda medida de desempenho, verifica-se o quanto o algoritmo proposto superou o melhor resultado encontrado na literatura.

Além disso, a tabela pode ser dividida em três grupos, ALNS 25K, ALNS 50K e ILS-RVND II, que representam o algoritmo ALNS (proposto por **Pisinger** e **Ropke** (2007)) utilizando 25 mil iterações, o algoritmo ALNS utilizando 50 mil iterações e o algoritmo proposto nesse trabalho, respectivamente.

Tabela 2. ILS-RVND II $\times$ Resultados da Literatura para o PRVMD

			ALNS 25K			ALNS 50K			ILS-RVND II			GAP	Imp.
	S_{UB}	Z	S^*	M_s	T_m	S^*	M_s	T_m	S^*	M_s	T_m	(%)	(%)
p01	576,87	**576,87**	**576,87**	576,87	14	**576,87**	576,87	29	**576,87**	576,87	2,48	0,000	0,000
p02	473,53	**473,53**	**473,53**	473,53	14	**473,53**	473,53	28	**473,53**	473,53	1,70	0,000	0,000
p03	641,19	**640,65**	641,19	641,19	32	641,19	641,19	64	641,19	641,19	5,14	0,084	0,000
p04	1001,59	**999,21**	1001,59	1008,49	42	1001,04	1006,09	88	1001,04	1004,37	10,62	0,514	0,000
p05	751,86	751,26[a]	751,86	753,04	58	751,26	752,34	120	**750,03**	751,36	9,79	0,013	0,164
p06	880,42	**876,50**	880,42	884,36	47	876,7	883,01	93	**876,50**	879,11	11,84	0,297	0,023
p07	881,97	**881,97**	**881,97**	889,14	43	**881,97**	889,36	88	**881,97**	884,14	9,83	0,246	0,000
p12	1318,95	**1318,95**	**1318,95**	1319,7	38	**1318,95**	1319,13	75	**1318,95**	1318,95	4,26	0,000	0,000
p15	2505,42	**2505,42**[a]	**2505,42**	2517,96	125	**2505,42**	2519,64	253	**2505,42**	2505,64	33,97	0,009	0,000
p18	3727,58	-	3727,58	3751,85	209	**3702,85**	3736,53	419	**3702,85**	3717,00	116,62	0,381	0,000
p21	5519,47	-	5519,47	5576,82	293	**5474,84**	5501,58	582	**5474,84**	5502,89	424,55	0,510	0,000

Valores marcados com [a] representam soluções em que a otimalidade não foi provada.

Na Tabela 2, em relação às melhores soluções encontradas pelo ILS-RVND II, o algoritmo encontrou todas as melhores soluções da literatura, melhorando duas soluções se comparado com o algoritmo ALNS, sendo que em uma delas o ótimo foi encontrado, e perdendo em apenas duas instâncias para o método exato. Salienta-se ainda que na instância $p05$, o ILS-RVND II apresentou um resultado melhor do que o encontrado pelo método exato, que não conseguiu provar a otimalidade da solução.

Em relação ao GAP, pode-se observar que o desvio das soluções médias encontradas pelo ILS-RVND II foi muito baixo (menores que 0,514 em todos os casos), variando em menos de 0,02% em cinco instâncias, mostrando a robustez do algoritmo proposto. O valor apresentado na coluna Imp. demonstra a qualidade das melhores soluções encontradas pelo ILS-RVND II, melhorando as soluções apresentadas pela literatura em dois casos.

5 Conclusões

Este trabalho aborda o Problema de Roteamento de Veículos com Múltiplos Depósitos, sendo proposto um algoritmo denominado ILS-RVND, que combina a heurística *Multi-start*, utilizando o procedimento *Variable Neighborhood Descent*, com variação aleatória das estruturas de vizinhança, integrada à metaheurística *Iterated Local Search*. Para explorar o espaço de soluções, além das estruturas de vizinhança Inter-Rotas

e Intra-Rota comuns ao PRV, foram utilizadas duas estruturas específicas para o problema abordado.

Inicialmente foram comparadas duas versões do algoritmo proposto, uma que não faz o uso das duas estruturas de vizinhança específicas e outra que faz. Nesta comparação verificou-se que a adição das duas novas estruturas diminuiu a variabilidade média das soluções encontradas.

A seguir, uma nova comparação foi feita utilizando o algoritmo proposto com as duas novas estruturas de vizinhança e os resultados encontrados na literatura. Os experimentos computacionais mostram que todas as soluções ótimas foram encontradas, exceto em duas instâncias, onde nenhum algoritmo heurístico presente na literatura também foi capaz alcançar. Além disso, em uma das instâncias em que o algoritmo exato não conseguiu provar o ótimo, uma solução de melhor qualidade foi encontrada. Observa-se também que a variabilidade média das soluções geradas foi de 0,514%, no máximo, mostrando a robustez do algoritmo proposto.

Agradecimentos

Os autores informam que este trabalho foi parcialmente financiado pelos seguintes órgãos de fomento: CAPES (PROCAD, PRO-ENG), CNPq (CT-INFO e UNIVERSAL), FAPERJ e FAPEMIG.

Referências

Baldacci, **R**. e **Mingozzi**, **A**. (2009). A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, v. 120, n. 2, p. 347–380.

Ballou, **R**. **H**. (2001). *Gerenciamento de cadeia de suprimentos:planejamento, organização e logística empresarial*. BOOKMAN, São Paulo, SP, quarta edição.

Chao, **I**. **M**.; **Golden**, **B**. **L**. e **Wasil**, **E**. (1993). A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions. *American Journal of Mathematical and Management Sciences*, v. 13, n. 3-4, p. 371–406.

Clarke, **G**. e **Wright**, **J**. **W**. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, v. 12, p. 568–581.

Cordeau, **J**. **F**.; **Gendreau**, **M**. e **Laporte**, **G**. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, v. 30, n. 2, p. 105–119.

Dantzig, **G**. **B**. e **Ramser**, **J**. **H**. (1959). The truck dispatching problem. *Management Science*, v. 6, n. 1, p. 80–91.

Gendreau, **M**.; **Hertz**, **A**. e **Laporte**, **G**. (1992). New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, v. 40, p. 1086–1094.

Gillett, **B**. **E**. e **Johnson**, **J**. **G**. (1976). Multi-terminal vehicle-dispatch algorithm. *Omega*, v. 4, p. 711–718.

Gillett, **B**. **E**. e **Miller**, **L**. **R**. (1974). A heuristic for the vehicle-dispatch problem. *Operations Research*, v. 21, p. 340–349.

Golden, **B**. **L**.; **Magnanti**, **T**. **L**. e **Nguyen**, **H**. **Q**. (1977). Implementing vehicle routing algorithms. *Networks*, v. 7, n. 2, p. 113–148.

Laporte, **G**.; **Nobert**, **Y**. e **Arpin**, **D**. (1984). Optimal solutions to capacitated multidepot vehicle routing problems. *Congressus Numerantium*, v. 44, p. 283–292.

Laporte, **G**.; **Nobert**, **Y**. e **Taillefer**, **S**. (1988). Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Science*, v. 22, n. 3, p. 161–172.

Lourenço, **H**. **R**.; **Martin**, **O**. **C**. e **Stützle**, **T**. (2003). Iterated local search. Glover, F. e Kochenberger, G. A., editors, *Handbook of Metaheuristics*, Capítulo 11, p. 321–353. Kluwer Academic Publishers.

Mladenović, **N**. e **Hansen**, **P**. (1997). Variable neighborhood search. *Computers and Operations Research*, v. 24, n. 11, p. 1097–1100.

Pisinger, **D**. e **Ropke**, **S**. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, v. 34, n. 8, p. 2403–2435.

Raft, **O**. **M**. (1982). A modular algorithm for an extended vehicle scheduling problem. *Computers & Operations Research*, v. 11, p. 67–76.

Renaud, **J**.; **Laporte**, **G**. e **Boctor**, **F**. **F**. (1996). A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*, v. 23, n. 3, p. 67–76.

Subramanian, **A**.; **Drummond**, **L**. **M**. **A**.; **Bentes**, **C**.; **Ochi**, **L**. **S**. e **Farias**, **R**. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, v. 37, n. 11, p. 1899 – 1911.

Tillman, **F**. **A**. (1969). The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, v. 3, p. 192–204.

Tillman, **F**. **A**. e **Cain**, **T**. **M**. (1972). An upperbound algorithm for the single and multiple terminal delivery problem. *Management Science*, v. 18, p. 664–682.

Tillman, **F**. **A**. e **Hering**, **R**. **W**. (1971). A study of a look-ahead procedure for solving the multiterminal delivery problem. *Transportation Science*, v. 5, p. 225–229.

Wren, **A**. e **Holliday**, **A**. (1972). Computer scheduling of vehicles from one or more depots to a number of delivery points. *Operations Research*, v. 23, p. 333–344.