

# UM NOVO ALGORITMO HEURÍSTICO PARA RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA E ENTREGA SIMULTÂNEA

**Thaís Cotta Barbosa da Silva<sup>1</sup>, Raphael Carlos Cruz<sup>1</sup>, Marcio Tadayuki Mine<sup>2</sup>,  
Marcone Jamilson Freitas Souza<sup>1</sup>, Ernesto del Rosario Santibanez<sup>1</sup>**

<sup>1</sup>Departamento de Ciência da Computação - Universidade Federal de Ouro Preto (UFOP)  
Campus Universitário, Morro do Cruzeiro, CEP 35.400-000, Ouro Preto (MG), Brasil  
thais\_cotta@yahoo.com.br, raphaelcarlos25@yahoo.com.br  
marcone@iceb.ufop.br, santibanez.ernesto@gmail.com

<sup>2</sup>GAPSO

Rua Lauro Müller, 1160, sala 3402, 22.290-160, Rio de Janeiro (RJ), Brasil  
marciomine@gmail.com

## Resumo

Este trabalho tem seu foco no Problema de Roteamento de Veículos com Coleta e Entrega Simultânea (PRVCES). Dada sua dificuldade de solução na otimalidade, é proposto um algoritmo heurístico, nomeado GENILS-TS, que combina seis procedimentos heurísticos: Inserção Mais Barata, Inserção Mais Barata com Múltiplas Rotas, GENIUS, *Iterated Local Search* (ILS), Descida em Vizinhança Variável (VND) e Busca Tabu (TS). Os três primeiros procedimentos visam à obtenção de uma solução inicial, enquanto os procedimentos VND e Busca Tabu são usados como métodos de busca local para o ILS. A Busca Tabu somente é acionada após certo número de iterações sem sucesso do VND. O algoritmo proposto foi testado em problemas-teste disponíveis na literatura e se mostrou capaz de gerar soluções de qualidade.

**Palavras-Chaves:** Problema de Roteamento de Veículos com Coleta e Entrega Simultânea, *Iterated Local Search*, Descida em Vizinhança Variável, GENIUS, Inserção Mais Barata, Busca Tabu.

## Abstract

This work addresses the Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD). Due to its complexity, we propose a heuristic algorithm for solving it, so-called GENILS-TS. This algorithm combines six heuristic procedures Cheapest Insertion, Cheapest Insertion with multiple routes, GENIUS, Iterated Local Search (ILS), Variable Neighborhood Descent (VND) and Tabu Search (TS). The first three procedures aim to obtain an initial solution, and the VND and TS are used as local search methods for ILS. TS is called after some iterations without success by VND. The algorithm was tested on benchmark instances taken from the literature and it was able to generate good quality solutions.

**Keywords:** Vehicle Routing Problem with Simultaneous Pickup and Delivery, Iterated Local Search, Variable Neighborhood Descent, GENIUS, Cheapest Insertion, Tabu Search.

## 1. INTRODUÇÃO

O Problema de Roteamento de Veículos (PRV), do inglês *Vehicle Routing Problem* (VRP), foi proposto por Dantzig and Ramser (1959) e pode ser definido como segue. Dado um conjunto  $N$  de clientes, cada qual associado a uma demanda  $d_i$ , e uma frota homogênea de veículos de capacidade  $Q$ , o objetivo é obter um conjunto de rotas a serem percorridas pelos veículos cujo custo seja mínimo, levando em consideração o atendimento completo da demanda dos clientes.

Uma importante variação do PRV, objeto de estudo deste trabalho, é o Problema de Roteamento de Veículos com Coleta e Entrega Simultânea (PRVCES). Proposto por Min (1989), este problema se diferencia do PRV clássico por ter associado aos clientes não só uma demanda  $d_i$  para cada cliente, mas também uma quantidade  $p_i$  de produtos a serem totalmente coletados, sendo que ambas as operações devem ser realizadas simultaneamente.

O PRVCES tem sido bastante estudado desde 1989. Alguns dos principais algoritmos de solução para este problema são brevemente comentados a seguir.

Wassan *et al.* (2007) propuseram uma Busca Tabu Reativa para resolvê-lo. Zachariadis *et al.* (2009) usaram uma técnica híbrida, unindo as metaheurísticas Busca Tabu e *Guided Local Search* (Voudouris and Tsang, 1996). Posteriormente, os mesmos autores propuseram um algoritmo evolucionário (Zachariadis *et al.*, 2010), que utiliza uma memória adaptativa para guardar as informações das soluções de alta qualidade obtidas durante a busca. Essas informações são usadas para gerar novas soluções em regiões que possuem grande chance de trazer melhores resultados, sendo posteriormente, melhoradas pela Busca Tabu. Mine *et al.* (2010) propuseram um algoritmo, nomeado GENILS, que utiliza a melhor solução gerada pelos métodos de Inserção Mais Barata, Inserção Mais Barata com Múltiplas Rotas e uma adaptação da heurística GENIUS (Gendreau *et al.*, 1992), como solução inicial. Como refinamento é utilizado o método *Iterated Local Search* (ILS), tendo o *Variable Neighborhood Descent* (VND) como método de busca local.

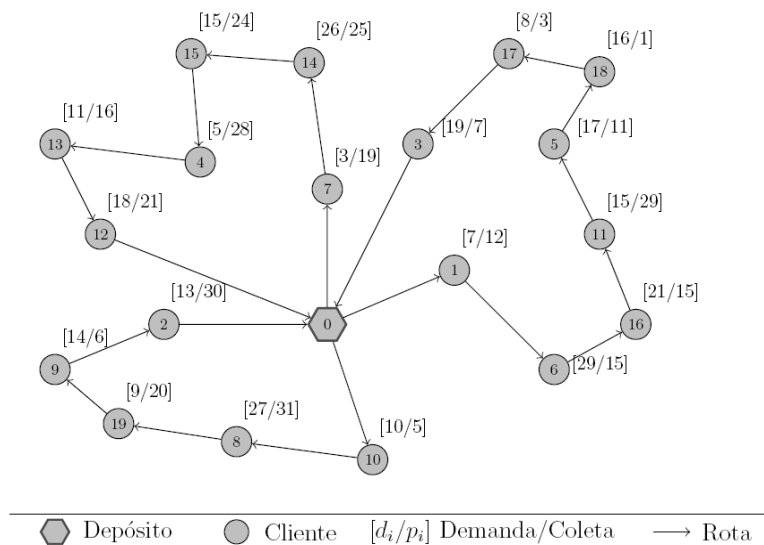
Subramanian *et al.* (2010) apresentaram um algoritmo paralelo para resolver o PRVCES. O algoritmo utiliza uma heurística *multi-start*, onde, a cada iteração, uma solução inicial é gerada por meio do procedimento Inserção mais Barata com Múltiplas Rotas. Essa solução é refinada pelo ILS, o qual utiliza como busca local o VND com ordem de vizinhança aleatória (RVND). Os experimentos realizados pelos autores foram feitos em dois *clusters* de computadores, sendo um com uma arquitetura composta por 128 e outro com 256 núcleos. Os resultados obtidos em 72 problemas-teste consagrados da literatura provaram que ele é o algoritmo de melhor desempenho até agora conhecido.

Este artigo tem seu foco na solução do PRVCES por meio de algoritmos sequenciais. Dada a simplicidade e competitividade do algoritmo GENILS frente aos demais algoritmos sequenciais da literatura, propõe-se aperfeiçoá-lo com a inserção de um módulo de Busca Tabu, substituindo a fase de refinamento do ILS a partir de um determinado número de iterações sem melhora no processo de busca. Além disso, o algoritmo proposto, nomeado GENILS-TS, utiliza recursos computacionais mais acessíveis do que aqueles de Subramanian *et al.* (2010), o que permite seu uso em empresas de menor porte.

## 2. DESCRIÇÃO DO PROBLEMA

O Problema de Roteamento de Veículos com Coleta e Entrega Simultânea (PRVCES), do inglês *Vehicle Routing Problem with Simultaneous Pickup and Delivery* (VRPSPD), envolve um conjunto  $N$  de clientes, cada qual com uma demanda  $d_i$  e coleta  $p_i$  a serem completamente atendidas, e um depósito, que é local de partida de um conjunto de

veículos de capacidade  $Q$  de carga, que são utilizados para satisfazer as necessidades desses clientes. O objetivo é construir um conjunto de rotas, a custo mínimo, que iniciam e terminam no depósito e atendam a todos os clientes sem ultrapassar a capacidade dos veículos. A Figura 1 ilustra um exemplo de uma solução para um PRVCES envolvendo 19 clientes e veículos de capacidade  $Q = 150$  unidades.



**Figura 1 - Exemplo do PRVCES**

Na rota do veículo do canto superior esquerdo da Figura 1, o veículo parte do depósito, passa pelo cliente 7 para entregar 3 unidades do produto e coletar 19 unidades. A seguir, se direciona para os clientes 14, 15, 4, 13, e 12, nesta ordem, entregando e coletando produtos, e retorna ao depósito.

### 3. METODOLOGIA

Nesta Seção é apresentada a metodologia utilizada para resolver o PRVCES. Na Subseção 3.1 é apresentado o algoritmo proposto. Nas subseções seguintes é feito seu detalhamento.

#### 3.1. ALGORITMO GENILS-TS

O algoritmo proposto, GENILS-TS, é um aperfeiçoamento do algoritmo GENILS de Mine *et al.* (2010). Este algoritmo combina as heurísticas ILS, VND (Hansen e Mladenovic, 2001), Busca Tabu – *TS*, do inglês *Tabu Search* (Glover e Laguna, 1997), assim como os procedimentos heurísticos Inserção Mais Barata, Inserção Mais Barata com Múltiplas Rotas e GENIUS (Gendreau *et al.*, 1992). Seu pseudocódigo é apresentado pelo Algoritmo 1.

Como pode ser observado no Algoritmo 1, os três métodos construtivos utilizados, os quais são descritos na Subseção 3.3, produzem as soluções  $s^A$ ,  $s^B$  e  $s^C$ , e cada uma delas é refinada por um procedimento VND. A melhor solução gerada se transforma na solução inicial  $s$ , sendo esta refinada pelo ILS. Para não ficar preso em ótimos locais e se dirigir para outras regiões do espaço de busca, foram utilizadas perturbações descritas na Subseção 3.8. Já os métodos VND e Busca Tabu (descritos nas Subseções 3.6 e 3.7, respectivamente) são utilizados como procedimentos de busca local, sendo a Busca Tabu acionada somente após *iterMaxSemMelhora* iterações sem melhora. Os ótimos locais encontrados são perturbados como descrito na Subseção 3.8.

Algoritmo *GENILS*

**Entrada:** Conjunto de buscas locais, Número máximo de iterações  $iter_{max}$

**Saída:** solução  $s$

```
1  $\gamma \leftarrow$  número aleatório no intervalo [0; 0,7]
2  $s^A \leftarrow IMB-1R()$ 
3  $nrotas \leftarrow$  número de rotas da solução  $s^A$ 
4  $s^B \leftarrow IMB-MR(nrotas, \gamma)$ 
5  $s^C \leftarrow VRGENIUS(nrotas)$ 
6  $s^A \leftarrow VND(s^A)$ 
7  $s^B \leftarrow VND(s^B)$ 
8  $s^C \leftarrow VND(s^C)$ 
9  $s \leftarrow \operatorname{argmin}\{f(s^A), f(s^B), f(s^C)\}$ 
10  $iter \leftarrow 0$ 
11 enquanto ( $iter < iter_{max}$ ) faça
12      $iter \leftarrow iter + 1$ 
13      $s' \leftarrow perturbação(s)$ 
14     se ( $iter < iterMaxSemMelhora$ ) faça
15          $s'' \leftarrow VND(s')$ 
16     senão
17          $s'' \leftarrow TS(s' ; TamListaTabu; iterMaxTS)$ 
18     fim-se
19     se ( $f(s'') < f(s)$ ) faça
20          $s \leftarrow s''; iter \leftarrow 0$ 
21     fim-se
22 fim-enquanto
23 retorne  $s$ 
```

Algoritmo 1: GENILS

### 3.2. FUNÇÃO DE AVALIAÇÃO

Uma solução  $s$  é avaliada pela função  $f$  definida pela Eq. (1). A primeira parcela de  $f$  corresponde à distância total percorrida. Já a segunda parcela é uma penalidade aplicada ao excesso de carga no veículo, ou seja, toda vez que o limite de carga é ultrapassado, o valor excedido é multiplicado por  $\beta$ , que corresponde a um valor suficientemente grande. Vale ressaltar que a geração de soluções inviáveis, onde a carga do veículo não é respeitada, é permitida, porém não é incentivada, já que a função de avaliação deve ser minimizada.

$$f(s) = \sum_{(i,j) \in A} c_{ij} x_{ij} + \beta \times \sum_{l \in R} \max \left\{ 0, \sum_{j \in N_l} (d_j + p_j) - Q \right\} \quad (1)$$

em que:

$N$ : conjunto dos clientes, incluindo o depósito;

$A$ : conjunto dos arcos  $(i, j) \in A$ , com  $i, j \in N$ ;

$R$ : conjunto de rotas existentes na solução;

$N_l$ : conjunto de clientes  $j$  pertencentes à rota  $l$ , com  $j \in N$  e  $l \in R$ ;

$c_{ij}$ : custo de deslocamento ou distância de  $i$  a  $j$ , com  $i, j \in N$ ;  
 $x_{ij}$ : variável binária que assume valor 1 se na solução  $s$  o arco  $(i, j) \in A$  for utilizado ( $x_{ij} = 1$ ) e valor zero ( $x_{ij} = 0$ ), caso contrário;  
 $d_j$ : valor da demanda do cliente  $j \in N$ ;  
 $p_j$ : quantidade a ser coletada no cliente  $j \in N$ ;  
 $Q$ : capacidade máxima do veículo.

### 3.3. GERAÇÃO DA SOLUÇÃO INICIAL

Para gerar a solução inicial, foram utilizadas as heurísticas construtivas de Inserção Mais Barata Rota a Rota, Inserção Mais Barata com Múltiplas Rotas e por último, uma adaptação da heurística GENIUS.

A Inserção Mais Barata Rota a Rota consiste basicamente na construção de uma subrota inicial contendo um cliente aleatório, e os demais clientes são inseridos a cada iteração respeitando as restrições do PRVCES. Para determinar qual cliente será incluído na rota, é utilizada a Equação (2), de forma que  $i$  e  $j$  correspondem a clientes já alocados a alguma rota e  $k$  representa um potencial cliente a ser inserido na rota.

$$e_{ij}^k = (c_{ik} + c_{kj} - c_{ij}) - \gamma \times (c_{0k} + c_{k0}) \quad (2)$$

O cliente  $k$  com o menor custo de inserção é adicionado à rota desde que a restrição de carga do veículo seja respeitada. O valor de  $\gamma \in [0, 1]$  é utilizado para favorecer clientes mais distantes do depósito, que normalmente entrariam tardiamente nas soluções, pelo seu alto custo de inserção.

A heurística Inserção Mais Barata com Múltiplas Rotas foi proposta por Subramanian (2008). Como parâmetro do método é necessário informar *a priori* o número de rotas. No algoritmo implementado, este número é obtido pela aplicação do algoritmo de Inserção Mais Barata Rota a Rota. Cada uma das rotas é iniciada por um único cliente escolhido aleatoriamente e os demais são inseridos a partir da Equação (2), respeitando-se o limite de carga do veículo.

O último método construtivo, denominado VRGENIUS, foi proposto por Mine *et al.* (2010). Ele é composto pelo método construtivo VRGENI e pelo método de refinamento VRUS. Ambos utilizam como busca local os procedimentos *3-opt* e *4-opt*. O funcionamento desses métodos é descrito, em detalhes, no trabalho indicado.

### 3.4. ESTRUTURAS DE VIZINHANÇA

Foram aplicados sete movimentos diferentes para explorar o espaço de soluções, todos bem conhecidos na literatura. O *Shift* é o movimento de realocação, e consiste em transferir um cliente de uma rota para outra. Já o *Shift(2,0)* consiste na realocação de dois clientes. O movimento *Swap* consiste na troca de um cliente de uma rota por um cliente pertencente a outra rota. O movimento *Swap(2,1)* consiste em trocar dois clientes consecutivos de uma rota com um cliente de outra rota, e o *Swap(2,2)* em trocar dois clientes consecutivos de uma rota com dois clientes consecutivos de outra. O movimento *M2-Opt* consiste em remover dois arcos e inserir dois novos arcos de forma a gerar uma nova rota. Finalmente, o movimento *kOr-Opt* realoca uma sequência de  $k$  clientes consecutivos para outro lugar na mesma rota, sendo  $k$  um parâmetro do método; no nosso caso,  $k$  varia de 3 a 5.

### 3.5. G3-OPT, G4-OPT E REVERSE

Os métodos de busca local *G3-opt* e *G4-opt* utilizam técnicas de inserção e remoção de arcos combinados com a ideia do *3-opt* e *4-opt* para melhorar a solução. Detalhes desses métodos são apresentados em Mine *et al.* (2010).

O procedimento *reverse* consiste em inverter o sentido de uma rota, sendo aplicado somente se ocorrer um aumento na carga residual da rota. A carga residual de uma rota é o valor da capacidade do veículo subtraído da maior carga do veículo nessa rota.

### 3.6. VND

O VND explora o espaço de soluções por meio de uma mudança sistemática de vizinhanças. No VND clássico considera-se um conjunto de vizinhanças em uma ordem pré-estabelecida, sendo que quando uma solução melhor é encontrada, retorna-se à primeira vizinhança. No VND implementado, no entanto, tal como em Subramanian *et al.* (2010) e Souza *et al.* (2010), a cada iteração a ordem das vizinhanças a serem exploradas pode mudar, pois há uma ordenação aleatória das sete vizinhanças descritas na Subseção 3.4. Logo depois é realizada uma intensificação por buscas locais utilizando as sete vizinhanças, seguido dos procedimentos *G3-opt*, *G4-opt* e *reverse*, apresentados na Subseção 3.5. O pseudocódigo do VND é apresentado no Algoritmo 2.

```
Procedimento VND
Entrada: solução  $s$ , conjunto de  $t$  vizinhanças distintas
Saída: solução refinada  $s$ 
1    $RN \leftarrow$  conjunto de  $t$  vizinhanças, em uma ordem aleatória
2    $i \leftarrow 1$ 
3   enquanto  $i \leq t$  faça
4     Seja  $s_0$  o melhor vizinho na vizinhança  $RN^{(i)}(s)$ 
5     se  $f(s_0) < f(s)$  então
6        $s \leftarrow s_0$ ;  $i \leftarrow 1$ 
7       { Intensificação nas rotas modificadas de  $s$  }
8        $s \leftarrow$  Busca Local com Shift
9        $s \leftarrow$  Busca Local com Shift(2,0)
10       $s \leftarrow$  Busca Local com Swap
11       $s \leftarrow$  Busca Local com M2-opt
12       $s \leftarrow$  Busca Local com Swap(2,1)
13       $s \leftarrow$  Busca Local com Swap(2,2)
14       $s \leftarrow$  Busca Local G3-opt
15       $s \leftarrow$  Busca Local G4-opt
16       $s \leftarrow$  Busca Local com kOr-opt,  $k = 3, 4, 5$ 
17       $s \leftarrow$  reverse
18    senão
19       $i \leftarrow i + 1$ 
20    fim-se
21  fim-enquanto
22  retorne  $s$ 
```

Algoritmo 2: Procedimento VND

### 3.7. BUSCA TABU

Busca Tabu (*TS – Tabu Search*) é uma metaheurística que utiliza uma estrutura de memória para explorar o espaço de soluções. A TS caminha no espaço de soluções movendo-se de uma solução para outra que seja seu melhor vizinho, mesmo que este vizinho seja um movimento de piora. Como estratégia para não retornar a uma solução já gerada anteriormente e, portanto, ciclar, a *TS* guarda em uma lista, a chamada Lista Tabu, informações sobre as soluções já geradas. O tamanho dessa lista é um parâmetro do método, que determina quanto tempo o movimento permanecerá proibido (tabu). O Algoritmo 3 apresenta o pseudocódigo da Busca Tabu implementada.

```
Procedimento TS
Entrada: TamListaTabu, iterMaxTS, Solução s
Saída: solução s
1   $s^* \leftarrow s$  {Melhor solução obtida até então}
2   $TamOriginal \leftarrow TamListaTabu$  {Tamanho original da Lista Tabu}
3   $iter \leftarrow 0$ ;  $MelhorIter \leftarrow 0$ 
4  enquanto ( $iter - MelhorIter < iterMaxTS$ ) faça
5      se ( $iter - MelhorIter < iterAux$ ) faça
6           $TamListaTabu \leftarrow TamListaTabu + 1$ 
7      fim-se
8       $s_0 \leftarrow Shift(s)$ 
9       $s_1 \leftarrow Swap(s)$ 
10      $s_2 \leftarrow Shift(2,0)(s)$ 
11      $s_3 \leftarrow Swap(2,1)(s)$ 
12      $s_4 \leftarrow Swap(2,2)(s)$ 
13      $s \leftarrow \text{argmin}\{f(s_0), f(s_1), f(s_2), f(s_3), f(s_4)\}$ 
14      $AtualizaListaTabu(TamListaTabu, Lista, Ant, Prox, iter, \Delta)$ 
15     se ( $f(s) < f(s^*)$ ) faça
16          $s^* \leftarrow s$ ;  $MelhorIter \leftarrow iter$ 
17          $TamListaTabu \leftarrow TamOriginal$ 
18     fim-se
19      $iter \leftarrow iter + 1$ 
20 fim-enquanto
21 retorne  $s^*$ 
```

Algoritmo 3: Procedimento *TS*

No Algoritmo 3, cinco soluções  $s_0, s_1, s_2, s_3$  e  $s_4$  são geradas a cada iteração utilizando os movimentos *Shift*, *Swap*, *Shift(2,0)*, *Swap(2,1)* e *Swap(2,2)* descritos na Subseção 3.4, sendo que a melhor delas passa a ser a solução corrente  $s$ . Em relação a Silva *et al.* (2011), a Busca Tabu contém três movimentos a mais, a saber: *Shift(2,0)*, *Swap(2,1)* e *Swap(2,2)*. Cada vez que se move para uma nova solução, a Lista Tabu é atualizada (vide Subseção 3.7.1). Após *iterAux* iterações sem melhora na solução global, o tamanho da Lista Tabu (*TamListaTabu*) é incrementado em uma unidade; entretanto, havendo melhora, ela retorna ao seu tamanho original. O procedimento é interrompido quando o número máximo de iterações sem melhora (*iterMaxTS*) na solução é atingido.

### 3.7.1. Lista Tabu

O movimento tabu utilizado pelo procedimento *TS* para evitar o retorno a uma solução gerada anteriormente consiste em proibir que o cliente afetado pelo movimento gerado seja sucessor do cliente adjacente a ele antes do movimento. A atualização da Lista Tabu é feita pelo procedimento descrito no Algoritmo 4.

|  |  |
|--|--|
| Procedimento AtualizaListaTabu   |  |
| <b>Entrada:</b> <i>TamListaTabu</i> , <i>Lista</i> , <i>Ant</i> , <i>Prox</i> , <i>iter</i> , $\Delta$ |  |
| <b>Saída:</b> <i>Lista</i>   |  |
| 1  | $min \leftarrow TamListaTabu + iter - \Delta$                    |
| 2  | $max \leftarrow TamListaTabu + iter + \Delta$                    |
| 3  | $DuracaoTS \leftarrow$ Valor aleatório no intervalo $[min, max]$ |
| 4  | $Lista(Ant, Prox) \leftarrow DuracaoTS$                          |
| 5  | retorne <i>Lista</i>   |

Algoritmo 4: Atualização da Lista Tabu

Conforme pode ser observado no Algoritmo 4, a lista assume tamanhos diferentes, devido à aleatoriedade presente no procedimento. Quando o tamanho da lista é menor, ou seja, menos proibitiva, se incentiva a intensificação da busca, contrastando com a diversificação que ocorre quando a lista é maior. Sendo assim, além de fazer um balanço entre intensificação e diversificação, essa estratégia procura evitar a ocorrência de ciclagem.

Para representar computacionalmente a Lista Tabu, utilizou-se uma matriz quadrada, em que cada célula  $(i, j)$  representa até que iteração está proibido o cliente  $j$  ficar como sucessor do cliente  $i$  na rota. As Figuras 2 e 3 ilustram a operação de atualização dessa matriz. Esta representação tem a vantagem de requerer complexidade  $O(1)$  para consultar se um movimento é tabu ou não. Caso fosse utilizada uma lista encadeada, a consulta seria  $O(m)$  no pior caso, sendo  $m$  o tamanho da lista.

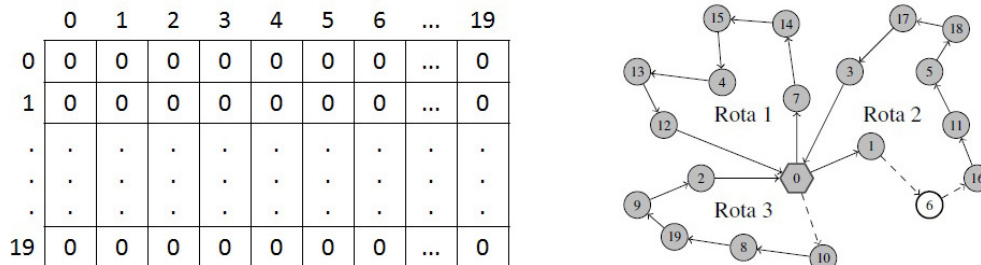


Figura 2 - Configuração da Matriz antes do Movimento

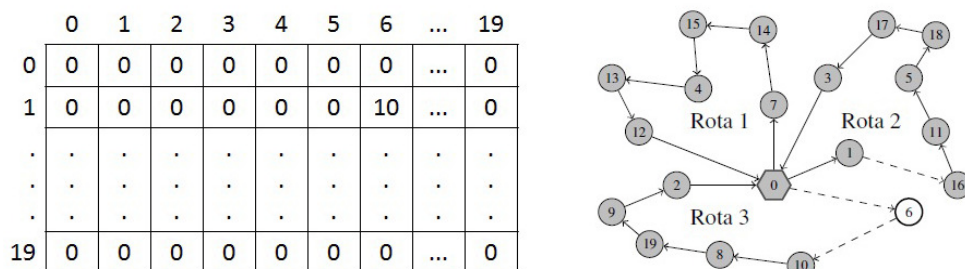


Figura 3 – Atualização da Matriz após o Movimento



Na Figura 2, o lado esquerdo representa a Lista Tabu em forma de matriz. Considerando que é a primeira iteração, ainda não existe nenhum movimento tabu. Já do lado direito temos a configuração da solução, onde pode ser observado que na rota 2 o veículo sai do depósito e atende aos clientes 1, 6, 16, 11, 5, 18, 17 e 3, nesta ordem, e retorna ao depósito. Já na Figura 3, o cliente 6 foi realocado da rota 2 para rota 3 pelo movimento *Shift*. Dessa forma, a atualização da matriz é realizada na linha 1 e coluna 6 da matriz, já que na configuração anterior o cliente 6 é o sucessor do cliente 1. Neste exemplo, qualquer movimento em que o cliente 1 seja sucedido pelo cliente 6 estará proibido até a iteração 10.

### 3.8. PERTURBAÇÕES

As perturbações são realizadas por um dos três procedimentos: Múltiplos *Shifts*, Múltiplos *Swaps* e *Ejection chain*, escolhidos aleatoriamente. Os procedimentos Múltiplos *Shifts* e Múltiplos *Swaps* consistem em  $k$  aplicações sucessivas dos movimentos *shift* e *swap*, sendo  $k$  um valor inteiro aleatório entre 1 e 3. O *Ejection chain* foi proposto por Rego and Roucairol (1996). Inicialmente, seleciona-se um subconjunto de  $m$  rotas  $R = r_1, r_2, \dots, r_m$  de forma arbitrária. Em seguida, transfere-se um cliente da rota  $r_1$  para a rota  $r_2$ , um cliente de  $r_2$  para  $r_3$  e assim sucessivamente até que um cliente seja transferido da rota  $r_m$  para a primeira rota  $r_1$ . Nesse movimento os clientes são escolhidos de forma aleatória.

## 4. RESULTADOS PARCIAIS

O algoritmo GENILS-TS foi codificado em C++ usando o compilador Visual C++ 2005 e executado em um microcomputador *Quad core*, 1,66 GHz e 4 GB de memória RAM e sistema operacional Windows Vista, sem explorar multiprocessamento.

Para validá-lo, os resultados alcançados pelo GENILS-TS foram comparados com os de outros quatro trabalhos da literatura: o algoritmo evolutivo de Zachariadis *et al.* (2010), o algoritmo paralelo de ILS-RVND de Subramanian *et al.* (2010) (256 núcleos), o GENILS de Mine *et al.* (2010) e a versão anterior do GENILS-TS de Silva *et al.* (2011). Destaca-se que os algoritmos de Zachariadis *et al.* (2010) e Mine *et al.* (2010) são, de nosso conhecimento, os melhores algoritmos seqüenciais conhecidos, enquanto o algoritmo de Subramanian *et al.* (2010) é o melhor algoritmo encontrado para o PRVCES. A versão anterior do GENILS-TS de Silva *et al.* (2011) difere da atual no número de vizinhanças exploradas pela Busca Tabu: enquanto a atual usa cinco vizinhanças, a anterior usava somente duas.

O algoritmo GENILS-TS foi testado nos 40 problemas-teste de Dethloff (2001); 14 de Salhi e Nagy (1999) e 18 de Montané e Galvão (2006). Os parâmetros adotados foram os seguintes:  $TamListaTabu = 10$ ,  $iterMaxTS = 300$ ,  $maxIter = 10000$  e  $\Delta = 3$ . Dado seu caráter estocástico, ele foi executado 25 vezes em cada problema-teste.

As Tabelas 1, 2 e 3 apresentam os melhores resultados encontrados pelo GENILS-TS e os comparam com os melhores resultados encontrados na literatura, que se referem aos trabalhos de Zachariadis *et al.* (2010), Subramanian *et al.* (2010) (256 núcleos) e Mine *et al.* (2010), assim como com os resultados da versão anterior, relatada em Silva *et al.* (2011). Nesta tabela, a primeira coluna representa o problema-teste e a segunda o melhor resultado da literatura. As colunas *Melhor* e  $Desv^{Best}$  apresentam, respectivamente, o melhor resultado de cada algoritmo e o desvio percentual em relação ao melhor resultado conhecido na literatura. A coluna *Tempo* apresenta o tempo médio das 25 execuções do GENILS-TS, em segundos.

Como pode ser observado na Tabela 1, o algoritmo proposto foi capaz de alcançar as melhores soluções conhecidas em todos os problemas-teste de Dethloff (2001).

Analisando a Tabela 2, que contém os resultados nos problemas-teste de Salhi e Nagy (1999), verifica-se que o algoritmo de Subramanian *et al.* (2010) é o de melhor

desempenho. De fato, ele alcança 8 das melhores soluções com um desvio médio de 0,65%, enquanto GENILS-TS obtém 6 melhores resultados da literatura com um desvio médio de 0,69%, melhorando em 0,01% o desvio médio apresentado por Silva *et al.* (2011). Por sua vez, o algoritmo de Zachariadis *et al.* (2010) encontrou 4 melhores soluções e um desvio de 0,76%, enquanto o de Mine *et al.* (2010) alcançou 4 das melhores soluções da literatura, mas com um desvio mais elevado, de 0,94%.

Na Tabela 3, em que são apresentados os resultados nos problemas-teste de Montané e Galvão (2006), o algoritmo de Subramanian *et al.* (2010) é o de melhor desempenho, pois encontra a maior quantidade de melhores soluções (15 no total) e com o menor desvio (0,01%), seguido do GENILS-TS com 12 melhores soluções e um desvio de 0,18%. Observa-se que o GENILS-TS, além de não perder para os demais algoritmos com relação à qualidade das melhores soluções e variabilidade das soluções, supera 9 resultados de Silva *et al.* (2011), 6 de Mine *et al.* (2010) e 10 de Zachariadis *et al.* (2010).

**Tabela 1. Resultados para os problemas-teste de Dethloff (2001)**

| Problema | MelhorLit | Zachariadis et al. (2010) |          | Subramanian et al. (2010) |          | Mine et al. (2010) |          | Silva et al. (2011) |          | GENILS-TS |          |          |
|----------|-----------|---------------------------|----------|---------------------------|----------|--------------------|----------|---------------------|----------|-----------|----------|----------|
|          |           | Melhor                    | DesvBest | Melhor                    | DesvBest | Melhor             | DesvBest | Melhor              | DesvBest | Melhor    | DesvBest | Tempo(s) |
| -        | -         | -                         | -        | -                         | -        | -                  | -        | -                   | -        | -         | -        | -        |
| SCA3-0   | 635,62    | 635,62                    | 0        | 635,62                    | 0        | 635,62             | 0        | 635,62              | 0        | 635,62    | 0        | 75,02    |
| SCA3-1   | 697,84    | 697,84                    | 0        | 697,84                    | 0        | 697,84             | 0        | 697,84              | 0        | 697,84    | 0        | 0,55     |
| SCA3-2   | 659,34    | 659,34                    | 0        | 659,34                    | 0        | 659,34             | 0        | 659,34              | 0        | 659,34    | 0        | 0,27     |
| SCA3-3   | 680,04    | 680,04                    | 0        | 680,04                    | 0        | 680,04             | 0        | 680,04              | 0        | 680,04    | 0        | 79,92    |
| SCA3-4   | 690,50    | 690,50                    | 0        | 690,50                    | 0        | 690,50             | 0        | 690,50              | 0        | 690,50    | 0        | 0,08     |
| SCA3-5   | 659,90    | 659,90                    | 0        | 659,90                    | 0        | 659,90             | 0        | 659,90              | 0        | 659,90    | 0        | 0,01     |
| SCA3-6   | 651,09    | 651,09                    | 0        | 651,09                    | 0        | 651,09             | 0        | 651,09              | 0        | 651,09    | 0        | 0,95     |
| SCA3-7   | 659,17    | 659,17                    | 0        | 659,17                    | 0        | 659,17             | 0        | 659,17              | 0        | 659,17    | 0        | 83,59    |
| SCA3-8   | 719,48    | 719,48                    | 0        | 719,48                    | 0        | 719,48             | 0        | 719,48              | 0        | 719,48    | 0        | 1,75     |
| SCA3-9   | 681,00    | 681,00                    | 0        | 681,00                    | 0        | 681,00             | 0        | 681,00              | 0        | 681,00    | 0        | 0,08     |
| SCA8-0   | 961,50    | 961,50                    | 0        | 961,50                    | 0        | 961,50             | 0        | 961,50              | 0        | 961,50    | 0        | 2,97     |
| SCA8-1   | 1049,65   | 1049,65                   | 0        | 1049,65                   | 0        | 1049,65            | 0        | 1049,65             | 0        | 1049,65   | 0        | 0,72     |
| SCA8-2   | 1039,64   | 1039,64                   | 0        | 1039,64                   | 0        | 1039,64            | 0        | 1039,64             | 0        | 1039,64   | 0        | 52,78    |
| SCA8-3   | 983,34    | 983,34                    | 0        | 983,34                    | 0        | 983,34             | 0        | 983,34              | 0        | 983,34    | 0        | 0,50     |
| SCA8-4   | 1065,49   | 1065,49                   | 0        | 1065,49                   | 0        | 1065,49            | 0        | 1065,49             | 0        | 1065,49   | 0        | 0,39     |
| SCA8-5   | 1027,08   | 1027,08                   | 0        | 1027,08                   | 0        | 1027,08            | 0        | 1027,08             | 0        | 1027,08   | 0        | 3,17     |
| SCA8-6   | 971,82    | 971,82                    | 0        | 971,82                    | 0        | 971,82             | 0        | 971,82              | 0        | 971,82    | 0        | 64,64    |
| SCA8-7   | 1051,28   | 1051,28                   | 0        | 1051,28                   | 0        | 1051,28            | 0        | 1051,28             | 0        | 1051,28   | 0        | 70,09    |
| SCA8-8   | 1071,18   | 1071,18                   | 0        | 1071,18                   | 0        | 1071,18            | 0        | 1071,18             | 0        | 1071,18   | 0        | 66,02    |
| SCA8-9   | 1060,50   | 1060,50                   | 0        | 1060,50                   | 0        | 1060,50            | 0        | 1060,50             | 0        | 1060,50   | 0        | 55,52    |
| CON3-0   | 616,52    | 616,52                    | 0        | 616,52                    | 0        | 616,52             | 0        | 616,52              | 0        | 616,52    | 0        | 6,33     |
| CON3-1   | 554,47    | 554,47                    | 0        | 554,47                    | 0        | 554,47             | 0        | 554,47              | 0        | 554,47    | 0        | 33,15    |
| CON3-2   | 518,00    | 518,00                    | 0        | 518,00                    | 0        | 518,00             | 0        | 518,00              | 0        | 518,00    | 0        | 92,30    |
| CON3-3   | 591,19    | 591,19                    | 0        | 591,19                    | 0        | 591,19             | 0        | 591,19              | 0        | 591,19    | 0        | 0,32     |
| CON3-4   | 588,79    | 588,79                    | 0        | 588,79                    | 0        | 588,79             | 0        | 588,79              | 0        | 588,79    | 0        | 14,86    |
| CON3-5   | 563,70    | 563,70                    | 0        | 563,70                    | 0        | 563,70             | 0        | 563,70              | 0        | 563,70    | 0        | 22,92    |
| CON3-6   | 499,05    | 499,05                    | 0        | 499,05                    | 0        | 499,05             | 0        | 499,05              | 0        | 499,05    | 0        | 63,03    |
| CON3-7   | 576,48    | 576,48                    | 0        | 576,48                    | 0        | 576,48             | 0        | 576,48              | 0        | 576,48    | 0        | 4,34     |
| CON3-8   | 523,05    | 523,05                    | 0        | 523,05                    | 0        | 523,05             | 0        | 523,05              | 0        | 523,05    | 0        | 34,13    |
| CON3-9   | 578,25    | 578,25                    | 0        | 578,25                    | 0        | 578,25             | 0        | 578,25              | 0        | 578,25    | 0        | 32,13    |
| CON8-0   | 857,17    | 857,17                    | 0        | 857,17                    | 0        | 857,17             | 0        | 857,17              | 0        | 857,17    | 0        | 27,87    |
| CON8-1   | 740,85    | 740,85                    | 0        | 740,85                    | 0        | 740,85             | 0        | 740,85              | 0        | 740,85    | 0        | 9,88     |
| CON8-2   | 712,89    | 712,89                    | 0        | 712,89                    | 0        | 712,89             | 0        | 712,89              | 0        | 712,89    | 0        | 6,74     |
| CON8-3   | 811,07    | 811,07                    | 0        | 811,07                    | 0        | 811,07             | 0        | 811,07              | 0        | 811,07    | 0        | 7,08     |
| CON8-4   | 772,25    | 772,25                    | 0        | 772,25                    | 0        | 772,25             | 0        | 772,25              | 0        | 772,25    | 0        | 1,74     |
| CON8-5   | 754,88    | 754,88                    | 0        | 754,88                    | 0        | 754,88             | 0        | 754,88              | 0        | 754,88    | 0        | 32,50    |
| CON8-6   | 678,92    | 678,92                    | 0        | 678,92                    | 0        | 678,92             | 0        | 678,92              | 0        | 678,92    | 0        | 39,37    |
| CON8-7   | 811,96    | 811,96                    | 0        | 811,96                    | 0        | 811,96             | 0        | 811,96              | 0        | 811,96    | 0        | 3,17     |
| CON8-8   | 767,53    | 767,53                    | 0        | 767,53                    | 0        | 767,53             | 0        | 767,53              | 0        | 767,53    | 0        | 4,49     |
| CON8-9   | 809,00    | 809,00                    | 0        | 809,00                    | 0        | 809,00             | 0        | 809,00              | 0        | 809,00    | 0        | 34,80    |
| Média    | -         | -                         | 0        | -                         | 0        | -                  | 0        | -                   | 0        | -         | 0        | -        |

**Tabela 2. Resultados para os problemas-teste de Salhi e Nagy (1999)**

| Problema | MelhorLit | Zachariadis et al. (2010) |          | Subramanian et al. (2010) |          | Mine et al. (2010) |          | Silva et al. (2011) |          | GENILS-TS |          |          |
|----------|-----------|---------------------------|----------|---------------------------|----------|--------------------|----------|---------------------|----------|-----------|----------|----------|
|          |           | Melhor                    | DesvBest | Melhor                    | DesvBest | Melhor             | DesvBest | Melhor              | DesvBest | Melhor    | DesvBest | Tempo(s) |
| -        | -         | -                         | -        | -                         | -        | -                  | -        | -                   | -        | -         | -        | -        |
| CMT1X    | 466,77    | 469,80                    | 0,65     | 466,77                    | 0,00     | 466,77             | 0,00     | 466,77              | 0,00     | 466,77    | 0,00     | 2,25     |
| CMT1Y    | 466,77    | 469,80                    | 0,65     | 466,77                    | 0,00     | 466,77             | 0,00     | 466,77              | 0,00     | 466,77    | 0,00     | 62,20    |
| CMT2X    | 668,77    | 684,21                    | 2,31     | 684,21                    | 2,31     | 684,11             | 2,29     | 684,11              | 2,29     | 684,11    | 2,29     | 127,97   |
| CMT2Y    | 663,25    | 684,21                    | 3,16     | 684,21                    | 3,16     | 684,11             | 3,15     | 684,11              | 3,15     | 684,11    | 3,15     | 132,64   |
| CMT3X    | 721,27    | 721,27                    | 0,00     | 721,27                    | 0,00     | 721,40             | 0,02     | 721,27              | 0,00     | 721,27    | 0,00     | 265,77   |
| CMT3Y    | 721,27    | 721,27                    | 0,00     | 721,27                    | 0,00     | 721,27             | 0,00     | 721,27              | 0,00     | 721,27    | 0,00     | 256,88   |
| CMT12X   | 644,70    | 662,22                    | 2,72     | 662,22                    | 2,72     | 662,22             | 2,72     | 662,22              | 2,72     | 662,22    | 2,72     | 269,68   |
| CMT12Y   | 659,52    | 662,22                    | 0,41     | 662,22                    | 0,41     | 663,50             | 0,60     | 663,50              | 0,60     | 662,22    | 0,41     | 208,54   |
| CMT11X   | 833,92    | 833,92                    | 0,00     | 833,92                    | 0,00     | 846,23             | 1,48     | 833,92              | 0,00     | 833,92    | 0,00     | 416,26   |
| CMT11Y   | 830,39    | 833,92                    | 0,43     | 833,92                    | 0,43     | 836,04             | 0,68     | 833,92              | 0,43     | 833,92    | 0,43     | 488,47   |
| CMT4X    | 852,46    | 852,46                    | 0,00     | 852,46                    | 0,00     | 852,46             | 0,00     | 852,46              | 0,00     | 852,46    | 0,00     | 858,95   |
| CMT4Y    | 852,35    | 852,46                    | 0,01     | 852,46                    | 0,01     | 862,28             | 1,17     | 855,52              | 0,37     | 855,52    | 0,37     | 406,01   |
| CMT5X    | 1029,25   | 1030,55                   | 0,13     | 1029,25                   | 0,00     | 1033,51            | 0,41     | 1030,50             | 0,13     | 1030,50   | 0,12     | 1655,14  |
| CMT5Y    | 1029,25   | 1030,55                   | 0,13     | 1029,25                   | 0,00     | 1036,14            | 0,67     | 1030,50             | 0,13     | 1030,50   | 0,12     | 790,88   |
| Média    | -         | -                         | 0,76     | -                         | 0,65     | -                  | 0,94     | -                   | 0,70     | -         | 0,69     | -        |

**Tabela 3. Resultados para os problemas-teste de Montané e Galvão (2006)**

| Problema | MelhorLit | Zachariadis et al. (2010) |          | Subramanian et al. (2010) |          | Mine et al.(2010) |          | Silva et al.(2011) |          | GENILS-TS |          |          |
|----------|-----------|---------------------------|----------|---------------------------|----------|-------------------|----------|--------------------|----------|-----------|----------|----------|
|          |           | Melhor                    | DesvBest | Melhor                    | DesvBest | Melhor            | DesvBest | Melhor             | DesvBest | Melhor    | DesvBest | Tempo(s) |
| -        | -         | -                         | -        | -                         | -        | -                 | -        | -                  | -        | -         | -        | -        |
| r101     | 1009,95   | 1009,95                   | 0,00     | 1009,95                   | 0,00     | 1009,95           | 0,00     | 1009,95            | 0,00     | 1009,95   | 0,00     | 208,56   |
| r201     | 666,20    | 666,20                    | 0,00     | 666,20                    | 0,00     | 666,20            | 0,00     | 666,20             | 0,00     | 666,20    | 0,00     | 165,84   |
| c101     | 1220,18   | 1220,99                   | 0,07     | 1220,18                   | 0,00     | 1220,18           | 0,00     | 1220,18            | 0,00     | 1220,18   | 0,00     | 203,19   |
| c201     | 662,07    | 662,07                    | 0,00     | 662,07                    | 0,00     | 662,07            | 0,00     | 662,07             | 0,00     | 662,07    | 0,00     | 91,87    |
| rc101    | 1059,32   | 1059,32                   | 0,00     | 1059,32                   | 0,00     | 1059,32           | 0,00     | 1059,32            | 0,00     | 1059,32   | 0,00     | 93,86    |
| rc201    | 672,92    | 672,92                    | 0,00     | 672,92                    | 0,00     | 672,92            | 0,00     | 672,92             | 0,00     | 672,92    | 0,00     | 189,50   |
| r1 2 1   | 3357,64   | 3376,30                   | 0,55     | 3360,02                   | 0,07     | 3357,64           | 0,00     | 3357,64            | 0,00     | 3357,64   | 0,00     | 890,19   |
| r2 2 1   | 1665,58   | 1665,58                   | 0,00     | 1665,58                   | 0,00     | 1665,58           | 0,00     | 1672,50            | 0,42     | 1665,58   | 0,00     | 1038,79  |
| c1 2 1   | 3629,89   | 3643,82                   | 0,38     | 3629,89                   | 0,00     | 3636,74           | 0,19     | 3645,45            | 0,43     | 3634,65   | 0,13     | 800,47   |
| c2 2 1   | 1726,59   | 1726,59                   | 0,00     | 1726,59                   | 0,00     | 1726,59           | 0,00     | 1731,74            | 0,30     | 1726,58   | 0,00     | 773,48   |
| rc1 2 1  | 3306,00   | 3323,56                   | 0,53     | 3306,00                   | 0,00     | 3312,92           | 0,21     | 3344,79            | 1,17     | 3312,92   | 0,21     | 977,57   |
| rc2 2 1  | 1560,00   | 1560,00                   | 0,00     | 1560,00                   | 0,00     | 1560,00           | 0,00     | 1560,00            | 0,00     | 1560,00   | 0,00     | 914,61   |
| r1 4 1   | 9605,75   | 9691,60                   | 0,89     | 9605,75                   | 0,00     | 9627,43           | 0,23     | 9627,88            | 0,23     | 9627,88   | 0,23     | 5415,90  |
| r2 4 1   | 3551,38   | 3572,38                   | 0,59     | 3551,38                   | 0,00     | 3582,08           | 0,86     | 3595,88            | 1,25     | 3582,08   | 0,86     | 4027,96  |
| c1 4 1   | 11098,21  | 11179,36                  | 0,73     | 11099,54                  | 0,01     | 11098,21          | 0,00     | 11125,55           | 0,25     | 11098,21  | 0,00     | 3757,66  |
| c2 4 1   | 3546,10   | 3549,27                   | 0,09     | 3546,10                   | 0,00     | 3596,37           | 1,42     | 3605,22            | 1,67     | 3592,71   | 1,31     | 3293,82  |
| rc1 4 1  | 9535,46   | 9645,27                   | 1,14     | 9536,77                   | 0,01     | 9535,46           | 0,00     | 9535,46            | 0,00     | 9535,46   | 0,00     | 6004,00  |
| rc2 4 1  | 3403,70   | 3423,62                   | 0,58     | 3403,70                   | 0,00     | 3422,11           | 0,54     | 3432,41            | 0,84     | 3419,76   | 0,47     | 3600,39  |
| Média    | -         | -                         | 0,31     | -                         | 0,01     | -                 | 0,19     | -                  | 0,36     | -         | 0,18     | -        |

## 5. CONCLUSÕES

Este trabalho teve seu foco no Problema de Roteamento de Veículos com Coleta e Entrega Simultânea (PRVCES). Dada a dificuldade de resolução desse problema em tempos computacionais aceitáveis no caso geral, foi proposto um algoritmo heurístico que combina os procedimentos *Iterated Local Search*, *Variable Neighborhood Descent*, Busca Tabu, Inserção Mais Barata Rota a Rota, Inserção Mais Barata com Múltiplas Rotas e uma adaptação da heurística GENIUS.

O algoritmo, nomeado GENILS-TS, foi testado em um conjunto de problemas-teste da literatura e comparado com quatro algoritmos eficientes da literatura. O algoritmo de Subramanian *et al.* (2010) foi, claramente, o de melhor desempenho. Entretanto, esse algoritmo usa 256 núcleos de processamento para explorar o espaço de soluções, enquanto os demais usam apenas um. O GENILS-TS, por sua vez, teve o segundo melhor desempenho, em termos de desvio médio e quantidade de melhores soluções, destacando-se, portanto, dentre os algoritmos sequenciais. Tendo em vista a estrutura computacional sofisticada requerida para aplicação do algoritmo paralelo de Subramanian *et al.* (2010), e que essa não é provavelmente a realidade da maioria das empresas de transporte, o GENILS-TS pode ser considerado, assim, uma alternativa mais factível.

Muitos aperfeiçoamentos ainda podem ser feitos no algoritmo proposto com o intuito de melhorar a qualidade de seus resultados. Como trabalho futuro pretende-se sofisticar a Busca Tabu com a inserção de mais estruturas de vizinhança, memória de longo prazo e Reconexão por Caminhos.

## AGRADECIMENTOS

Os autores agradem à FAPEMIG e ao CNPq pelo apoio dado ao presente trabalho.

## REFERÊNCIAS

- Dantzig, G. B.; Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6:80–91.
- Dethloff, J. (2001). Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR Spektrum*, 23:79–96.

- Gendreau, M.; Hertz, A.; Laporte, G. (1992). New insertion and post optimization procedures or the traveling salesman problem. *Operations Research*, 40:1086–1094.
- Glover, F.; Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publisher, Boston.
- Hansen, P.; Mladenovic, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467.
- Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research A*, 23(5):377–386.
- Mine, M. T.; Silva, M. S. A.; Ochi, L. S.; Souza, M. J. F. (2010). O problema de roteamento de veículos com coleta e entrega simultânea: uma abordagem via Iterated Local Search e GENIUS. In *Transporte em transformação XIV: trabalhos vencedores do prêmio CNT de Produção Acadêmica 2009*, p. 59–78. Editora Positiva, Brasília.
- 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*, 33(3):595–619.
- Rego, C.; Roucairol, C. (1996). *Meta-Heuristics Theory and Applications*, capítulo A Parallel Tabu Search Algorithm Using Ejection Chains for the Vehicle Routing Problem, p. 661–675. Kluwer Academic Publisher, Boston.
- 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*, 50:1034–1042.
- Silva, T. C. B.; Cruz, R. C.; Mine, M. T.; Souza, M. J. F.; Santibanez, E. R. (2011). GENILS-TS: um algoritmo heurístico híbrido para resolução do Problema de Roteamento de Veículos com Coleta e Entrega Simultânea. Submetido ao XLIII Simpósio Brasileiro de Pesquisa Operacional.
- Souza, M. J. F.; Coelho, I. M.; Ribas, S.; Santos, H. G.; Merschmann, L. H. C. (2010). A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research*, 207: 1041-1051.
- Subramanian, A. (2008). Metaheurística Iterated Local Search aplicada ao problema de roteamento de veículos com coleta e entrega simultânea. Dissertação de mestrado, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal da Paraíba, João Pessoa.
- Subramanian, A.; Drummond, L. M. A.; Bentes, C.; Ochi, L. S.; Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers and Operations Research*, 37:1899–1911.
- Voudouris, C.; Tsang, E. (1996). Partial constraint satisfaction problems and guided local search. In *The Second International Conference on the Practical Application of Constraint Technology (PACT'96)*, p. 337–356.
- Wassan, N. A.; Wassan, A. H.; Nagy, G. (2007). A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. *Journal of Combinatorial Optimization*, 15(4):368–386.
- Zachariadis, E. E.; Tarantilis, C. D.; Kiranoudis, C. T. (2009). A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service. *Expert Systems with Applications*, 36(2):1070–1081.
- Zachariadis, E. E.; Tarantilis, C. D.; Kiranoudis, C. T. (2010). An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries. *European Journal of Operational Research*, 202:401–411.