

ALGORITMOS *SIMULATED ANNEALING* EFICIENTES PARA RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE TEMPO

Aloísio de Castro Gomes Júnior
Universidade Federal de Ouro Preto
algomesjr2004@yahoo.com.br

Marcene Jamilson Freitas Souza
Universidade Federal de Ouro Preto
marcone@iceb.ufop.br

Alexandre Xavier Martins
Universidade Federal de Ouro Preto
xmartins@uai.com.br

RESUMO

Este trabalho apresenta dois algoritmos eficientes, ambos baseados na metaheurística *Simulated Annealing* (SA), para resolver o Problema de Roteamento de Veículos com Janela de Tempo. Esse problema tem como objetivo determinar as rotas de custo mínimo para uma frota de veículos de mesma capacidade, atendendo à demanda de um conjunto de clientes dentro de um intervalo de tempo determinado, chamado janela de tempo. As metodologias propostas, denominadas SA-A e SA-RAI, incorporam ao algoritmo *Simulated Annealing* clássico, mecanismos auto-adaptativos para determinação da temperatura inicial e número de iterações em uma mesma temperatura. No método SA-RAI, quando a temperatura atinge um valor limiar, a mesma é reaquecida um certo número de vezes, possibilitando escapar de ótimos locais. Além disso, ele conta com uma fase de intensificação. Sempre que uma melhor solução é encontrada, ela é submetida a um procedimento de refinamento, visando ao seu melhoramento. As duas metodologias foram aplicadas a 168 instâncias-teste da literatura e 13 novos melhores resultados foram encontrados.

Palavras-Chave: Roteamento de Veículos, *Simulated Annealing*, Heurísticas.

ABSTRACT

This work presents two efficient algorithms based on Simulated Annealing metaheuristic (SA) for solving the Vehicle Routing Problem with Time Windows. The goal of this problem is to find a set of routes with minimum cost (in terms of distance traveled) for a fleet of identical vehicles based at the depot, obeying the capacity of the vehicles and the demand of a set of costumers and its predefined time windows. The proposed methodologies, named SA-A e SA-RAI, use auto-adaptive mechanisms to determine the initial temperature and number of iterations in a same temperature. In the SA-RAI method, when the temperature is close to zero, this one is reheating in order to escape local optimuns. Besides SA-RAI uses a refinement phase when a new best solution is found. The proposed methodologies were applied to a 168 benchmarks and 13 new best solutions were found.

Keywords: Vehicle routing, Simulated Annealing, Heuristics.

1. Introdução

O Problema de Roteamento de Veículos com Janela de Tempo (PRVJT) é um dos vários problemas derivados do Problema de Roteamento de Veículos (PRV), um dos problemas mais estudados da área de Otimização Combinatória. O PRVJT é um problema da classe NP-difícil, ou seja, não existem algoritmos que o resolva em tempo polinomial e o uso exclusivo de métodos exatos se torna bastante restrito.

O PRVJT pode ser definido como se segue. Seja $G=\{V, A\}$ um grafo onde $V=\{v_0, v_1, \dots, v_n\}$ é o conjunto de vértices, e $A=\{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ é o conjunto de arcos. O vértice v_0 representa o Centro de Distribuição (CD), enquanto os demais n vértices correspondem aos clientes. Com A são associados uma matriz de custos (c_{ij}) e uma matriz de tempo de viagens (t_{ij}) . Se estas matrizes são simétricas, o caso mais comum, então é padrão definir o PRVJT em um grafo não direcionado $G=\{V, E\}$, onde $E=\{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ é um conjunto de arestas. A cada cliente está associada uma demanda não-negativa m_i e um tempo de serviço f_i . Uma frota de veículos idênticos de capacidade Q é atribuída ao CD. O número de veículos pode ser conhecido com antecedência ou tratado como uma variável de decisão. Neste trabalho considera-se que o número de veículos é ilimitado. O PRVJT consiste em designar um conjunto de rotas de entrega ou coleta tal que:

- (1) cada rota inicia e termina no CD;
- (2) cada cliente seja visitado exatamente uma vez por exatamente um veículo;
- (3) a demanda total de cada rota não exceda Q ;
- (4) a janela de tempo $[e_i, l_i]$ de cada cliente deve ser respeitada, isto é, um cliente não pode ser atendido antes de e_i nem após o tempo l_i e
- (5) o custo total da rota seja minimizado.

Encontrar uma solução de boa qualidade para o PRVJT é de fundamental importância, uma vez que segundo Ballou (2001), o custo de transporte corresponde de um terço a dois terços dos custos logísticos de uma empresa.

Devido à importância prática do PRV e o desafio de se resolvê-lo eficientemente, várias técnicas de solução têm sido relatadas na literatura. Algumas delas são apresentadas a seguir.

Clark e Wright (1964) desenvolveram uma técnica heurística que insere, em cada iteração, um cliente a uma rota de acordo com a economia que este apresenta na função de avaliação. O cliente que gera a maior economia é inserido na rota. O processo inicia-se com n rotas. Estas inserções são feitas até o momento que nenhuma economia é mais possível. Esta técnica foi aplicada ao PRV clássico.

Golden *et al.* (1984) adaptaram a heurística de Clark e Wright (1964) ao PRV com frota heterogênea, onde os veículos possuem diferentes capacidades. Os autores desenvolveram várias fórmulas de economia tentando incorporar à fórmula de economia desenvolvida por Clark e Wright (1964) os custos fixos relativos a cada tipo de veículo que poderia ser utilizado.

Liu e Shen (1999) basearam-se no trabalho de Golden *et al.* (1984), inserindo às fórmulas desenvolvidas por estes as restrições de janela de tempo pertinentes a cada cliente. Esses autores abordaram o PRV com frota heterogênea e janelas de tempo.

Renaud e Boctor (2002) apresentaram uma outra técnica, denominada Algoritmo de Varreduras. Este algoritmo se baseia na geração de rotas factíveis pela rotação de um raio centrado no CD e gradualmente inclui clientes em uma rota de um veículo até a capacidade do veículo ou comprimento da rota ser atingido. Uma nova rota é então iniciada e o processo é repetido até que o plano inteiro tenha sido varrido. Cada rota gerada é denominada pétala.

Gendreau *et al.* (1999) aplicaram a metaheurística Busca Tabu (BT) ao PRV com frota heterogênea. O procedimento desenvolvido por esses autores se baseia em um procedimento de memória adaptativa para gerar soluções iniciais para o procedimento BT. Nesse procedimento, as inserções dos clientes às rotas são baseadas na heurística GENIUS, desenvolvida por Gendreau *et al.* (1992) para resolver o Problema do Caixeiro Viajante.

Tailard (1993) aplica o mesmo algoritmo de Gendreau *et al.* (1999) ao PRV com frota heterogênea. O procedimento difere na estrutura de vizinhança. Ao invés de usar a heurística GENIUS, o mesmo utiliza trocas e inserções convencionais. Outro ponto diferente é que o método proposto pelo autor faz uso de um procedimento de re-otimização das rotas através da aplicação de um algoritmo exato para o Problema do Caixeiro Viajante.

Tan *et al.* (2001) aplicaram as metaheurísticas *Simulated Annealing*, Busca Tabu e Algoritmos Genéticos ao PRV com janela de tempo. Nessas técnicas, procura-se gerar soluções iniciais factíveis baseadas na inserção, em cada iteração, de um cliente a uma rota, respeitando-se as restrições de capacidade do veículo e da janela de tempo de cada cliente. Há, também, um procedimento de dupla troca baseado em determinados tipos de operações, identificadas por dois números, os quais indicam quantos clientes devem ser passados de uma rota para a outra. Por exemplo, a operação (2, 1) indica que devem ser passados dois clientes da rota 1 para a rota 2 e um cliente da rota 2 para a rota 1. O movimento a ser realizado é aquele que entre toda a vizinhança gera a maior economia. No procedimento de dupla troca somente as seguintes operações são permitidas: (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2). Com as metodologias propostas, os autores encontraram 18 novas melhores soluções.

Cordeau *et al.* (2002) apresentaram quatro características cruciais para um bom algoritmo para o PRV e suas variantes. As quatro características apontadas são: Precisão (que mede o quão distante a solução heurística ficou da solução ótima ou da melhor solução conhecida); Velocidade (que avalia o tempo para a tomada de decisões); Simplicidade (que avalia a facilidade de se implementar e entender o código e também o número de parâmetros que são utilizados, que podem facilitar ou dificultar a compreensão do algoritmo, além de dificultar a implementação do mesmo) e a Flexibilidade (que avalia a capacidade para incluir novas restrições comumente encontradas na maioria das aplicações da vida real). Esses autores apresentam, ainda, uma comparação entre várias técnicas aplicadas à resolução do PRV baseadas nestas quatro características.

Neste trabalho são apresentadas duas metodologias, ambas baseadas na metaheurística *Simulated Annealing*, para resolver eficientemente o PRVJT. Essas metodologias fazem uso de mecanismos auto-adaptativos para determinar a temperatura inicial e o número máximo de iterações em uma dada temperatura. A segunda metodologia consiste em uma variação da primeira, com a temperatura sendo reinicializada um determinado número de vezes. A temperatura é reinicializada sempre que a temperatura atinge um valor limiar. Além disso, sempre que o método encontra uma melhor solução, é aplicado um mecanismo de busca local para refinar esta solução.

Este trabalho está organizado como segue. A seção 2 apresenta um modelo de programação matemática para o PRVJT. A seção 3 apresenta, em detalhe, as metodologias desenvolvidas. Os resultados são apresentados e discutidos na seção 4. A seção 5 conclui o trabalho.

2. Formulação de Programação Matemática

Apresenta-se, a seguir, uma formulação de programação matemática, descrita em Tan *et al.* (2001), para o Problema de Roteamento de Veículos com Janela de Tempo, considerando que a frota é heterogênea.

- Variáveis de decisão:

t_i = tempo de chegada no cliente i

w_i = tempo de espera no cliente i

$$x_{ijk} = \begin{cases} 1 & \text{se há um arco ligando o cliente } i \text{ ao cliente } j \text{ usando o veículo } k \\ 0 & \text{caso contrário} \end{cases}$$

- Dados de entrada:

K = número total de veículos

n = número total de clientes

d_{ij} = distância euclidiana entre o cliente i e o cliente j

c_{ij} = custo incorrido no arco do cliente i ao cliente j

t_{ij} = tempo de viagem entre o cliente i e o cliente j

m_i = demanda do cliente i

q_k = capacidade do veículo k

e_i = início da janela de tempo do cliente i

l_i = final da janela de tempo do cliente i

f_i = tempo de serviço no cliente i

r_k = tempo de rota máximo permitido para o veículo k .

As equações (1) a (11) modelam o PRVJT:

$$\text{Min} \sum_{i=0}^n \sum_{j=0, j \neq i}^n \sum_{k=1}^K c_{ij} x_{ijk} \quad (1)$$

Sujeito a:

$$\sum_{k=1}^K \sum_{j=1}^n x_{ijk} \leq K \quad \text{para } i = 0 \quad (2)$$

$$\sum_{j=1}^n x_{ijk} = \sum_{j=1}^n x_{jik} \leq 1 \quad \text{para } i = 0 \text{ e } \forall k \in \{1, 2, \dots, K\} \quad (3)$$

$$\sum_{k=1}^K \sum_{j=0, j \neq i}^n x_{ijk} = 1 \quad \forall i \in \{1, \dots, n\} \quad (4)$$

$$\sum_{k=1}^K \sum_{i=0, i \neq j}^n x_{ijk} = 1 \quad \forall j \in \{1, \dots, n\} \quad (5)$$

$$\sum_{i=1}^n m_i \sum_{j=0, j \neq i}^n x_{ijk} \leq q_k \quad \forall k \in \{1, \dots, K\} \quad (6)$$

$$\sum_{i=0}^n \sum_{j=0, j \neq i}^n x_{ijk} (t_{ij} + f_i + w_i) \leq r_k \quad \forall k \in \{1, \dots, K\} \quad (7)$$

$$t_0 = w_0 = f_0 = 0 \quad (8)$$

$$\sum_{k=1}^K \sum_{i=0, i \neq j}^n x_{ijk} (t_i + t_{ij} + f_i + w_i) \leq t_j \quad \forall j \in \{1, \dots, n\} \quad (9)$$

$$e_i \leq (t_i + w_i) \leq l_i \quad \forall i \in \{1, \dots, n\} \quad (10)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j \in \{1, \dots, n\}, \quad \forall k \in \{1, \dots, K\} \quad (11)$$

A expressão (1) é a função objetivo do problema. A restrição (2) especifica que há no máximo K rotas saindo do CD. As equações (3) fazem com que todas as rotas iniciem e terminem no CD. Os conjuntos de equações (4) e (5) definem que todo nó cliente pode ser visitado somente uma vez por cada veículo. As equações (6) limitam o carregamento dos veículos às suas capacidades. As equações (7) limitam o tempo máximo de viagem. As restrições (8) a (10) definem as janelas de tempo, enquanto as restrições (11) indicam a bivalência das variáveis de decisão.

3. Metodologias Desenvolvidas

Nesta seção mostra-se como uma solução para o problema é representada, como são geradas as soluções iniciais, os tipos de movimentos usados para explorar o espaço de soluções, a função de avaliação para guiar a busca e a descrição detalhada dos dois algoritmos propostos.

3.1. Representação de uma Solução

Uma solução é representada por uma lista de números inteiros, sendo que os números positivos representam os clientes a serem visitados e os negativos, os veículos e suas respectivas capacidades. Essa lista é ordenada e a ordem de apresentação dos números indica o veículo usado e a ordem de visita dos clientes. Assim, a lista $\{-200, 1, 3, 2, 5, -200, 6, 4, 7\}$ indica que um veículo de capacidade 200 sai do centro de distribuição (CD) e visita os clientes 1, 3, 2 e 5, nesta seqüência, e retorna ao CD. Outro veículo de mesma capacidade sai do CD e visita os clientes 6, 4 e 7 e retorna ao CD.

3.2. Determinação da Solução Inicial

Com a finalidade de se gerar uma solução inicial, criou-se um procedimento capaz de gerar soluções iniciais factíveis. Este procedimento insere, em cada iteração, um cliente em uma rota. A inserção é feita observando o limite de capacidade do veículo e a janela de tempo de cada cliente.

Os clientes a serem inseridos nas rotas são escolhidos aleatoriamente. Estes vão sendo inseridos enquanto o limite de capacidade do veículo alocado à rota não for superado. Quando este limite é ultrapassado, inicia-se uma nova rota e um novo cliente é selecionado.

Para não ultrapassar a janela de tempo pré-definida para cada cliente, toda vez que um cliente é inserido, define-se um grupo de possíveis clientes candidatos a serem inseridos e o sorteio do próximo cliente é feito entre eles. Se esse grupo é vazio então é iniciada uma nova rota e seleciona-se um novo cliente dentro do grupo, que agora conta com todos os clientes que ainda não foram inseridos.

Este procedimento termina quando todos os clientes forem inseridos. Observa-se que, com este procedimento, as soluções iniciais são sempre factíveis. Além disso, tem-se um limite superior para o número de veículos a serem utilizados.

3.3. Estrutura de Vizinhaça

Dada uma solução s , um vizinho s' desta solução é obtido ou por um movimento de realocação de clientes ou por um movimento de troca entre clientes.

O movimento de realocação pode inter-rotas, quando um cliente é passado de uma rota para outra; ou intra-rotas, quando a ordem de visita de um cliente é alterada dentro de uma mesma rota. A Figura 1 ilustra a aplicação dos dois possíveis movimentos de realocação.

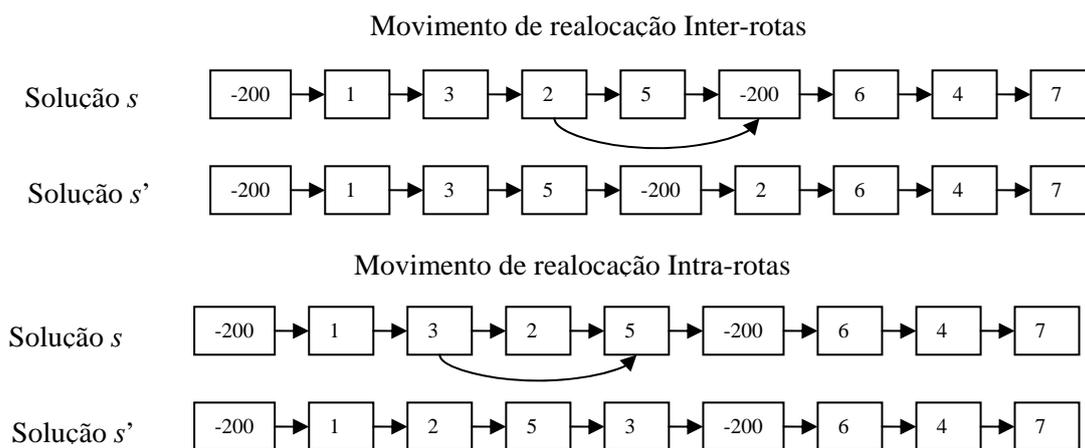


Figura 1: Estrutura de Vizinhaça

Na Figura 1, o movimento inter-rotas mostra que o cliente 2 é realocado para ser o primeiro cliente a ser visitado pelo segundo veículo de capacidade 200. Já o movimento intra-rotas desta figura ilustra o cliente 3 sendo realocado para ser visitado após o cliente 5 da primeira rota.

O movimento de troca também pode ser intra-rotas, situação na qual dois clientes de uma mesma rota tem suas ordens de visita trocadas entre si; ou inter-rotas, quando há a permutação entre clientes de rotas distintas.

3.4. Determinação da Temperatura Inicial

Utilizou-se um mecanismo auto-adaptativo para determinar a temperatura inicial dos algoritmos baseados em *Simulated Annealing* (SA). Tal mecanismo, descrito em Souza (2005), consiste em partir de uma solução s qualquer e de uma temperatura de partida baixa. A seguir, conta-se quantos vizinhos dessa solução s são aceitos em um determinado número de iterações (S_{Amax}) nessa temperatura. Caso esse número de vizinhos aceitos seja elevado, no caso 90% dos vizinhos, então retorna-se a temperatura corrente como a temperatura inicial para o processo de refinamento pelos métodos baseados em SA. Caso o número de vizinhos aceitos não atinja o valor mínimo requerido, aumenta-se

a temperatura segundo uma certa taxa β , e repete-se a contagem do número de vizinhos aceitos naquela temperatura. O procedimento prossegue até que se obtenha o número mínimo de vizinhos aceitos. A temperatura na qual essa condição ocorre representa a temperatura inicial para o método *Simulated Annealing*. A Figura 2 apresenta o pseudocódigo para este procedimento. Nesta figura, β é a taxa de aumento da temperatura (no caso, $\beta = 1,1$), γ é a taxa mínima de aceitação de soluções vizinhas (no caso, $\gamma = 0,90$) e T_0 é uma temperatura de partida para o método ($T_0 = 2$).

```

Procedimento TemperaturaInicial ( $f(\cdot)$ ,  $N(\cdot)$ ,  $\beta$ ,  $\gamma$ ,  $S_{Amax}$ ,  $T_0$ ,  $s$ )
1.  $T \leftarrow T_0$ ; { Temperatura Corrente }
2. Continua  $\leftarrow$  TRUE;
3. enquanto (Continua) faça
4.   Aceitos  $\leftarrow$  0; { Número de vizinhos aceitos na temperatura  $T$  }
5.   Para IterT = 1 até  $S_{Amax}$  faça
6.     Gere um vizinho qualquer  $s' \in N(s)$ ;
7.      $\Delta = f(s') - f(s)$ ;
8.     se ( $\Delta < 0$ )
9.       então
10.        Aceitos  $\leftarrow$  Aceitos + 1;
11.      senão
12.        Tome  $x \in [0, 1]$ ;
13.        se ( $x < e^{-\Delta/T}$ ) então Aceitos  $\leftarrow$  Aceitos + 1;
14.      fim-se;
15.    fim-para;
16.    se (Aceitos  $\geq \gamma \times S_{Amax}$ )
17.      então Continua  $\leftarrow$  FALSE;
18.      senão  $T \leftarrow \beta \times T$ ;
19.    fim-se;
20.  fim-enquanto;
21.   $T_0 \leftarrow T$ ;
22.  Retorne  $T_0$ ;
fim TemperaturaInicial;

```

Figura 2: Pseudocódigo do procedimento de determinação da temperatura inicial

3.5. Função de Avaliação

Uma solução s é avaliada por uma função que leva em conta o custo de transporte do cliente i para o cliente j , a capacidade do veículo que irá atender aquela rota e a janela de tempo em que o cliente deve ser atendido. Esta função é dada pela expressão (12):

$$f(s) = \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} + \mu f_1(s) + \lambda f_2(s) \quad (12)$$

em que c_{ij} representa o custo de transporte do cliente i para o cliente j , $f_1(s)$ representa a sobrecarga dos veículos, ou seja, o excesso de peso nos veículos e $f_2(s)$ representa o tempo em que um veículo excede o horário máximo permitido para atendimento a determinado cliente. O cliente 0 é o centro de distribuição (CD), μ é um fator de penalidade igual ao valor da maior distância entre dois clientes quaisquer da instância considerada e λ outro fator de penalidade, fixado em $\lambda=1000$.

3.6. Algoritmo SA-A

Este método consiste no algoritmo *Simulated Annealing* clássico, ao qual é incorporado um mecanismo auto-adaptativo para determinar a temperatura inicial e o número máximo de iterações em uma dada temperatura. Esse método parte de uma solução inicial gerada conforme seção 3.2 e consiste de um procedimento iterativo que gera, a cada iteração, um único vizinho s' da solução corrente s . Seja Δ a variação de valor da função objetivo ao mover-se para uma solução vizinha candidata, isto é,

$\Delta = f(s') - f(s)$. Para o PRVJT, o método aceita o movimento e a solução vizinha passa a ser a nova solução corrente se $\Delta < 0$. Caso $\Delta \geq 0$ a solução vizinha candidata também poderá ser aceita, mas neste caso, com uma probabilidade $e^{-\Delta/T}$, onde T é um parâmetro do método, chamado de temperatura, que regula a probabilidade de se aceitar soluções de pior custo. A temperatura T assume, inicialmente, um valor elevado T_0 (obtida pelo procedimento auto-adaptativo descrito na seção 3.4). Após um número fixo de iterações, que representa o número de iterações necessárias para o sistema atingir o equilíbrio térmico em uma dada temperatura, esta é gradativamente diminuída por uma razão de resfriamento α , tal que $T_k \leftarrow \alpha \times T_{k-1}$, sendo $0 < \alpha < 1$. Com esse procedimento, dá-se, no início uma chance maior para escapar de mínimos locais e, à medida que T se aproxima de zero, o algoritmo se comporta como um método de descida, uma vez que diminui a probabilidade de se aceitar movimentos de piora.

O procedimento pára quando a temperatura chega a um valor próximo de zero e nenhuma solução de piora da solução corrente é mais aceita, isto é, quando o sistema está estável. A solução obtida quando o sistema encontra-se nesta situação evidencia o encontro de um mínimo local.

Os parâmetros de controle do procedimento são a razão de resfriamento α (fixada em 0,998), o número de iterações para cada temperatura ($S_{Amax} = k \times n$, onde $k = 1$ e $n =$ número total de clientes) e a temperatura inicial T_0 . A Figura 3 apresenta o pseudocódigo para o algoritmo SA-A.

```

Procedimento SA-A ( $f(\cdot)$ ,  $N(\cdot)$ ,  $\alpha$ ,  $S_{Amax}$ ,  $T_0$ ,  $s$ )
1.  $s^* \leftarrow s$ ; {melhor solução obtida até então}
2. IterT  $\leftarrow 0$ ; {Número de iterações na temperatura  $T$ }
3.  $T \leftarrow T_0$ ; {Temperatura Corrente}
4. enquanto ( $T > 0$ ) faça
5.     enquanto (IterT <  $S_{Amax}$ ) faça
6.         IterT  $\leftarrow$  IterT + 1;
7.         Gere um vizinho qualquer  $s' \in N(s)$ ;
8.          $\Delta = f(s') - f(s)$ ;
9.         se ( $\Delta < 0$ )
10.            então
11.                 $s \leftarrow s'$ ;
12.                se ( $f(s') < f(s^*)$ ) então  $s^* \leftarrow s'$ ;
13.            senão
14.                Escolha  $x \in [0, 1]$ ;
15.                se ( $x < e^{-\Delta/T}$ ) então  $s \leftarrow s'$ ;
16.         fim-se;
17.     fim-enquanto;
18.      $T \leftarrow \alpha \times T$ ;
19.     IterT  $\leftarrow 0$ ;
20. fim-enquanto;
21.  $s \leftarrow s^*$ ;
22. Retorne  $s$ ;
fim SA-A;

```

Figura 3: Pseudocódigo do algoritmo SA-A

3.7. O algoritmo SA-RAI

O algoritmo SA-RAI é um aprimoramento do algoritmo SA-A, consistindo de duas modificações neste. A primeira incorpora ao algoritmo SA-A um mecanismo de busca local (MBL), que utiliza movimentos de troca (intra e inter-rotas) para melhorar a melhor solução obtida até então. Este mecanismo é bastante simples, consistindo em fixar um cliente e procurar uma melhor posição para ele através de um movimento de troca. Se esta nova posição gerar um custo menor, o movimento é aceito. Este procedimento é repetido para todos os clientes. A segunda mudança consistiu na incorporação ao algoritmo SA-A de um dispositivo de reaquecimento. Quando a temperatura atinge seu valor mínimo, ela é reaquecida de acordo com a equação (13):

$$T_r = \frac{T_{r-1}}{e^{\ln(T_{r-1}/100)/(NR-r+1)}} \quad (13)$$

em que T_r é a r -ésima temperatura de reinício ($r > 1$) e NR é o número de reinícios. Esta fórmula obriga o algoritmo em seu último reinício a trabalhar com uma temperatura inicial igual a 100, que foi o parâmetro utilizado por Tan *et al.* (2001). Nesta metodologia fixou-se $NR = 2$.

O algoritmo SA-RAI parte de uma solução s gerada conforme seção 3.2 e temperatura inicial gerada de forma auto-adaptativa conforme seção 3.4. O pseudocódigo é apresentado na Figura 4.

```

Procedimento SA-RAI ( $f(\cdot)$ ,  $N(\cdot)$ ,  $\alpha$ ,  $SAmax$ ,  $T_0$ ,  $NR$ ,  $s$ )
1    $s^* \leftarrow s$ ; {melhor solução obtida até então}
2   IterT  $\leftarrow$  0; {Número de iterações na temperatura  $T$ }
3    $T \leftarrow T_0$ ; {Temperatura Corrente}
4    $r \leftarrow 1$ ; {Número corrente de reinícios};
5    $T_r \leftarrow T_0$ ;
6   enquanto ( $r \leq NR$ )
7       enquanto ( $T > 0$ ) faça
8           enquanto (IterT <  $SAmax$ ) faça
9               IterT  $\leftarrow$  IterT + 1;
10              Gere um vizinho qualquer  $s' \in N(s)$ ;
11               $\Delta = f(s') - f(s)$ ;
12              se ( $\Delta < 0$ )
13                  então
14                       $s \leftarrow s'$ ;
15                      se ( $f(s') < f(s^*)$ )
16                          então
17                              Melhore  $s'$  usando MBL;
18                               $s^* \leftarrow s'$ ;
19                  senão
20                      Escolha  $x \in [0, 1]$ ;
21                      se ( $x < e^{-\Delta/T}$ ) então  $s \leftarrow s'$ ;
22              fim-se;
23          fim-enquanto;
24           $T \leftarrow \alpha \times T$ ;
25          IterT  $\leftarrow$  0;
26      fim-enquanto;
27       $T \leftarrow T_r / e^{\ln(T_r/100)/(NR-r+1)}$ ;
28       $T_r \leftarrow T$ ;
29       $r \leftarrow r + 1$ ;
30  fim-enquanto;
31   $s \leftarrow s^*$ ;
32  Retorne  $s$ ;
fim SA-RAI;

```

Figura 4: Pseudocódigo do algoritmo SA-RAI

4. Apresentação e Análise dos Resultados Computacionais

Os algoritmos propostos foram desenvolvidos na linguagem C, utilizando o compilador C++ Builder 6.0 da Borland, e testados em um computador PENTIUM® IV, 1.8 GHz com 512 MB de memória RAM, sob plataforma Windows XP Pro.

Os problemas-teste adotados para a validação dos algoritmos foram as 168 instâncias apresentadas por Solomon (1987), as quais envolvem 25, 50 e 100 clientes. Essas instâncias são divididas em 3 grupos: C – onde os clientes se encontram clusterizados, ou seja, estão distribuídos geograficamente em grupos de clientes próximos uns dos outros; R – onde os clientes são distribuídos

aleatoriamente sem formar grupos e distantes uns dos outros e RC – onde se tem uma mistura dos dois grupos anteriores. Cada um dos dois algoritmos desenvolvidos foi executado trinta vezes, cada qual partindo de uma semente diferente de números aleatórios.

Os resultados para as duas metodologias são apresentados nas tabelas a seguir. Nessas tabelas, os melhores resultados da literatura advêm de duas fontes, cujo acesso se deu em 16/05/2005: (1) <http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html> e (2) <http://w.cba.neu.edu/~msolomon>.

A coluna “Desvio Médio” é calculada com base na fórmula (14):

$$Desvio\ Médio = \frac{\frac{1}{30} \left(\sum_{i=1}^{30} fo_i \right) - fo^*}{fo^*} \quad (14)$$

em que fo^* é o melhor resultado encontrado na literatura para a distância total percorrida por todos os veículos e fo_i é o resultado encontrado na i -ésima execução do algoritmo.

Já a coluna “Desvio da Melhor Solução” é calculada com base na fórmula (15):

$$Desvio\ da\ Melhor\ Solução = \frac{\min_{i=1, \dots, 30} \{fo_i\} - fo^*}{fo^*} \quad (15)$$

A Tabela 1 apresenta os resultados relativos ao desempenho dos algoritmos aplicados aos grupos de instâncias envolvendo 25, 50 e 100 clientes. Nesta tabela são apresentadas, para cada algoritmo, a média dos desvios médios encontrados para cada grupo de instância, bem como a média dos desvios das melhores soluções encontradas nesse grupo.

Tabela 1: Desempenho médio dos algoritmos nos grupos de instâncias

Grupo de Instâncias	SA-A		SA-RAI	
	Média do “Desvio Médio”	Média do “Desvio da Melhor Solução”	Média do “Desvio Médio”	Média do “Desvio da Melhor Solução”
C25	2,92%	0,00%	0,13%	0,00%
R25	2,78%	0,00%	0,88%	0,00%
RC25	3,06%	0,02%	0,98%	0,02%
C50	7,04%	0,00%	1,56%	0,15%
R50	5,27%	1,06%	2,94%	0,56%
RC50	7,95%	1,73%	4,04%	0,89%
C100	14,78%	3,50%	8,23%	2,72%
R100	6,01%	1,62%	3,62%	0,07%
RC100	8,62%	2,69%	5,78%	1,67%
Média	6,49%	1,18%	3,13%	0,68%

Como pode ser observado na Tabela 1, as melhores soluções geradas pelo procedimento SA-A desviam dos melhores resultados da literatura em apenas 1,18%, na média, enquanto que para o procedimento SA-RAI esse desvio desce para 0,68%. Considerando os valores médios gerados nas diversas execuções dos algoritmos, o procedimento SA-A tem um desvio médio de 6,49%, enquanto que no procedimento SA-RAI esse desvio é reduzido para 3,13%. Assim, tendo em vista que as melhores soluções da literatura foram obtidas por diferentes algoritmos, cada qual explorando as especificidades de uma dada instância ou conjunto de instâncias, os algoritmos desenvolvidos mostraram-se capazes de produzir soluções finais bastante satisfatórias.

A Tabela 2 mostra as características de três novas melhores soluções geradas a partir dos procedimentos propostos, relativas às instâncias R204-50, R207-50 e R211-50.

As tabelas 3(a) e 3(b) mostram as características das soluções geradas pelos algoritmos propostos, bem como o desempenho desses na resolução de todas as instâncias relativas a 100 clientes. Observa-se que dez novas melhores soluções são geradas.

Como pode ser verificado pelas tabelas 2, 3(a) e 3(b), o procedimento SA-RAI demanda cerca de 2,5 vezes mais tempo que o procedimento SA-A. Em contrapartida, como se deduz da Tabela 1, ele consegue soluções finais com desvios aproximadamente 50% menores.

5. Conclusões

Neste trabalho foram propostos dois algoritmos eficientes, baseados na metaheurística *Simulated Annealing*, para resolver o problema de roteamento de veículos com janela de tempo. As metodologias propostas, denominadas SA-A e SA-RAI, incorporam ao algoritmo *Simulated Annealing* clássico, mecanismos auto-adaptativos para determinação da temperatura inicial e número de iterações em uma mesma temperatura. No método SA-RAI, quando a temperatura atinge um valor limiar, a mesma é reaquecida um certo número de vezes, possibilitando escapar de ótimos locais. Além disso, ele conta com uma fase de intensificação. Sempre que uma melhor solução é encontrada, ela é submetida a um procedimento de refinamento, visando ao seu melhoramento.

As duas metodologias foram aplicadas a 168 instâncias-teste da literatura e 13 novos melhores resultados foram encontrados. Além desses resultados obtidos, as duas metodologias produziram soluções finais que, em média, diferem pouco das melhores soluções da literatura. O algoritmo SA-RAI mostrou-se superior ao algoritmo SA-A, pois suas soluções finais têm um desvio em relação às melhores soluções da literatura, em média, 50% menores em um tempo computacional cerca de 2,5 vezes maior.

Referências Bibliográficas

BALLOU, R. **Gerenciamento da Cadeia de Suprimentos: Planejamento, Organização e Logística Empresarial**. São Paulo: Bookman, 2001.

CLARK G.; WRIGHT, J. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. **Operations Research**, v. 12, p. 568-581, 1964.

CORDEAU, J.F.; GENDREAU, M.; LAPORTE, G.; POTVIN, J.Y.; SEMET, F. A guide to Vehicle Routing Problem. **Journal of the Operational Research Society**, v. 53, p. 512-522, 2002.

GENDREAU, M.; LAPORTE, G.; MUSARAGANYI, C.; TAILLARD, E. D. A Tabu Search Heuristic for the Heterogeneous Fleet Vehicle Routing Problem. **Computers & Operations Research**, v. 26, p. 1153-1173, 1999.

GENDREAU, M.; HERTZ, A.; LAPORTE, G. New Insertion and Postoptimization Procedures for the Traveling Salesman Problem. **Operations Research**, v. 40, p. 1086-1093, 1992.

GOLDEN, B.; ASSAD, A.; LEVY, L.; GHEYSENS, F. The Fleet Size and Mix Vehicle Routing. **Computers & Operations Research**. Grã-Bretanha, v. 11, n. 1, p. 49-66, 1984.

LIU, F.; SHEN, S. Y.. A Method for Vehicle Routing Problem with Multiple Vehicle Types and Time Windows. **Proc. Natl. Sci. Council**. Hsinchu, Taiwan, v. 23, p. 526-536, 1999.

RENAUD, J.; BOCTOR, F. F. A Sweep-based Algorithm for the Fleet Size and Mix Vehicle Routing Problem. **European Journal of Operational Research**, v. 240, p. 618-628, 2002.

SOUZA, M. J. F. **Inteligência Computacional para Otimização**. Departamento de Computação, Universidade Federal de Ouro Preto, Ouro Preto, Disponível em <http://www.decom.ufop.br/prof/marcone/InteligenciaComputacional/InteligenciaComputacional.ps>, 2005.

TAILLARD, É. Parallel iterative search methods for vehicle routing problems. **Networks**, v. 23, p. 661-673, 1993.

TAN, K.C.; LEE, L. H.; ZHU, Q. L.; OU, K. Heuristic methods for vehicle routing problem with time windows. **Artificial Intelligence in Engineering**, v 15, p. 281-295, 2001.

Tabela 2 (a): Desempenho dos algoritmos nas instâncias R204, R207 e R211 com 50 clientes

Instância	Melhor Publicado		SA-A					SA-RAI				
	DT	NV	DT	NV	Desvio Médio	Desvio da Melhor Solução	Tempo médio (s)	DT	NV	Desvio Médio	Desvio da Melhor Solução	Tempo médio (s)
R204-50	509,5 ⁽¹⁾	2	508,1	3	5,32%	-0,27%	11,78	508,1	3	2,28%	-0,27%	28,09
R207-50	584,6 ⁽¹⁾	4	576,2	3	4,43%	-1,44%	12,40	576,8	3	1,55%	-1,33%	29,26
R211-50	551,3 ⁽¹⁾	1	542,1	3	4,29%	-1,67%	12,12	541,9	3	1,78%	-1,71%	28,25

Tabela 3(a): Desempenho dos algoritmos nas instâncias com 100 clientes

Instância	Melhor Publicado		SA-A					SA-RAI				
	DT	NV	DT	NV	Desvio Médio	Desvio da Melhor Solução	Tempo médio (s)	DT	NV	Desvio Médio	Desvio da Melhor Solução	Tempo médio (s)
C101-100	827,3 ⁽²⁾	10	827,3	10	10,02%	0,00%	84,83	827,3	10	2,65%	0,00%	216,93
C102-100	827,3 ⁽²⁾	10	827,3	10	11,09%	0,00%	78,35	827,3	10	3,43%	0,00%	195,78
C103-100	826,3 ⁽²⁾	10	834,8	10	13,89%	1,03%	72,59	826,3	10	5,93%	0,00%	176,10
C104-100	822,9 ⁽²⁾	10	824,1	10	10,78%	0,15%	65,00	834,0	10	6,68%	1,35%	153,89
C105-100	827,3 ⁽²⁾	10	827,3	10	10,88%	0,00%	78,74	827,3	10	3,64%	0,00%	199,60
C106-100	827,3 ⁽²⁾	10	827,3	10	8,18%	0,00%	79,19	827,3	10	2,20%	0,00%	191,75
C107-100	827,3 ⁽²⁾	10	827,3	10	12,12%	0,00%	76,03	827,3	10	2,67%	0,00%	189,05
C108-100	827,3 ⁽²⁾	10	827,3	10	10,91%	0,00%	72,63	827,3	10	3,23%	0,00%	173,74
C109-100	827,3 ⁽²⁾	10	827,3	10	9,89%	0,00%	68,09	827,3	10	3,76%	0,00%	160,79
C201-100	589,1 ⁽²⁾	3	623,9	4	21,71%	5,91%	77,36	624,0	4	16,79%	5,92%	187,29
C202-100	589,1 ⁽²⁾	3	626,8	4	24,38%	6,40%	73,60	626,9	4	15,25%	6,42%	173,76
C203-100	588,7 ⁽²⁾	3	642,3	4	18,88%	9,10%	68,39	617,8	4	14,38%	4,94%	159,53
C204-100	588,1 ⁽²⁾	3	654,3	5	19,12%	11,26%	62,34	635,1	4	12,96%	7,99%	144,93
C205-100	586,4 ⁽²⁾	3	622,2	4	17,60%	6,11%	72,48	619,8	4	13,26%	5,70%	165,34
C206-100	586,0 ⁽²⁾	3	624,0	4	17,52%	6,48%	69,54	613,1	4	10,43%	4,62%	157,78
C207-100	585,8 ⁽²⁾	3	624,3	4	16,80%	6,57%	69,06	613,3	4	11,75%	4,69%	158,04
C208-100	585,8 ⁽²⁾	3	623,8	4	17,41%	6,49%	67,22	612,9	4	10,95%	4,63%	151,83
R101-100	1637,7 ⁽²⁾	20	1707,0	20	6,95%	4,23%	88,46	1675,9	21	4,00%	2,33%	228,56
R102-100	1466,6 ⁽²⁾	18	1505,0	18	6,05%	2,62%	82,38	1491,1	19	3,88%	1,67%	212,50
R103-100	1208,7 ⁽²⁾	14	1248,6	15	8,02%	3,30%	75,16	1236,0	15	4,85%	2,26%	191,48
R104-100	971,5 ⁽²⁾	11	1046,0	13	11,60%	7,67%	69,59	1021,5	12	8,42%	5,15%	171,01
R105-100	1355,3 ⁽²⁾	15	1412,9	17	9,04%	4,25%	80,46	1399,1	17	6,38%	3,23%	201,05
R106-100	1234,6 ⁽²⁾	13	1287,8	15	8,16%	4,31%	77,23	1267,7	14	6,15%	2,68%	196,39
R107-100	1064,6 ⁽²⁾	11	1120,3	12	8,79%	5,23%	73,02	1107,7	13	7,74%	4,05%	181,88

Tabela 3(b): Desempenho dos algoritmos nas instâncias com 100 clientes

Instância	Melhor Publicado		SA-A					SA-RAI				
	DT	NV	DT	NV	Desvio Médio	Desvio da Melhor Solução	Tempo médio (s)	DT	NV	Desvio Médio	Desvio da Melhor Solução	Tempo médio (s)
R108-100	960,88 ⁽²⁾	9	990,09	10	7,84%	3,04%	72,09	967,85	11	5,18%	0,73%	167,82
R109-100	1146,9 ⁽²⁾	13	1219,2	14	10,39%	6,30%	72,63	1185,3	13	7,26%	3,35%	180,77
R110-100	1068,0 ⁽²⁾	12	1126,4	12	10,14%	5,47%	69,42	1119,8	12	7,93%	4,85%	175,59
R111-100	1048,7 ⁽²⁾	12	1114,3	13	9,86%	6,26%	71,55	1101,2	13	7,75%	5,01%	179,53
R112-100	982,14 ⁽²⁾	9	1009,20	11	8,31%	2,76%	70,85	1000,26	11	5,31%	1,84%	162,85
R201-100	1143,2 ⁽²⁾	8	1206,1	10	8,40%	5,50%	67,37	1188,2	10	6,50%	3,94%	160,34
R202-100	1191,7 ⁽²⁾	3	1106,08	9	-4,16%	-7,18%	70,43	1062,06	9	-6,99%	-10,88%	156,87
R203-100	939,54 ⁽²⁾	3	906,01	7	1,54%	-3,57%	67,65	889,69	6	0,26%	-5,31%	149,18
R204-100	825,52 ⁽²⁾	2	766,71	5	-0,50%	-7,12%	63,60	759,70	5	-3,17%	-7,97%	139,16
R205-100	994,42 ⁽²⁾	3	1020,26	7	6,41%	2,60%	68,79	1005,73	7	3,99%	1,14%	147,07
R206-100	906,14 ⁽²⁾	3	930,41	7	7,97%	2,68%	65,70	917,41	6	5,15%	1,24%	144,24
R207-100	893,33 ⁽²⁾	2	845,41	5	0,71%	-5,36%	64,54	832,38	6	-1,98%	-6,82%	141,05
R208-100	726,75 ⁽²⁾	2	749,36	4	8,54%	3,11%	61,32	732,60	4	6,37%	0,80%	134,48
R209-100	909,16 ⁽²⁾	3	903,57	7	3,95%	-0,61%	65,74	897,53	6	2,51%	-1,28%	143,38
R210-100	939,34 ⁽²⁾	3	950,07	9	5,37%	1,14%	66,91	952,22	8	3,90%	1,37%	147,08
R211-100	892,71 ⁽²⁾	2	809,00	6	-5,10%	-9,38%	61,78	786,80	6	-8,10%	-11,86%	134,52
RC101-100	1619,8 ⁽²⁾	15	1712,5	19	10,40%	5,72%	80,37	1686,0	18	7,14%	4,09%	210,34
RC102-100	1457,4 ⁽²⁾	14	1531,4	16	10,34%	5,08%	77,96	1515,7	16	7,37%	4,00%	205,81
RC103-100	1258,0 ⁽²⁾	11	1342,2	13	12,28%	6,69%	75,16	1335,4	12	10,80%	6,15%	192,22
RC104-100	1135,48 ⁽²⁾	10	1219,61	12	12,69%	7,41%	76,08	1204,05	12	10,19%	6,04%	178,44
RC105-100	1513,7 ⁽²⁾	15	1593,3	17	10,46%	5,26%	78,52	1544,0	16	7,15%	2,00%	202,19
RC106-100	1424,73 ⁽²⁾	11	1450,13	15	6,25%	1,78%	79,86	1454,80	14	4,33%	2,11%	188,23
RC107-100	1207,8 ⁽²⁾	12	1289,0	13	13,91%	6,72%	72,15	1292,2	13	10,73%	6,99%	182,61
RC108-100	1114,2 ⁽²⁾	11	1183,1	12	13,90%	6,18%	69,54	1143,5	12	9,09%	2,63%	176,49
RC201-100	1261,8 ⁽²⁾	9	1294,8	10	11,03%	2,62%	68,30	1311,8	10	7,65%	3,96%	165,48
RC202-100	1092,3 ⁽²⁾	8	1127,3	9	10,52%	3,20%	66,48	1140,3	10	7,33%	4,39%	159,54
RC203-100	1049,62 ⁽²⁾	3	958,04	6	-0,76%	-8,73%	68,99	977,38	7	-3,20%	-6,88%	152,58
RC204-100	798,41 ⁽²⁾	3	832,06	5	9,79%	4,21%	64,69	813,38	5	7,20%	1,87%	142,80
RC205-100	1154,0 ⁽²⁾	7	1217,3	9	10,66%	5,49%	65,93	1183,7	8	6,36%	2,57%	159,85
RC206-100	1146,32 ⁽²⁾	3	1108,55	7	1,65%	-3,29%	69,25	1107,20	7	-0,43%	-3,41%	151,13
RC207-100	1061,14 ⁽²⁾	3	1015,87	7	1,22%	-4,27%	65,58	993,42	7	-2,01%	-6,38%	145,64
RC208-100	828,14 ⁽²⁾	3	819,86	5	3,59%	-1,00%	63,41	800,46	5	2,71%	-3,34%	138,82