Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXIX, págs. 9471-9484 (artículo completo)
Eduardo Dvorkin, Marcela Goldschmit, Mario Storti (Eds.)
Buenos Aires, Argentina, 15-18 Noviembre 2010

UM ALGORITMO HÍBRIDO PARA A RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELAS DE TEMPO

Sabir Ribas[a], Anand Subramanian[a], Igor Machado Coelho[a], Luiz Satoru Ochi[a] e Marcone J. F. Souza[b]

[a]*Instituto de Computação, Universidade Federal Fluminense, Niterói/RJ, Brasil, {sribas,anand,imcoelho,satoru}@ic.uff.br*

[b]*Departamento de Computação, Universidade Federal de Ouro Preto, Ouro Preto/MG, Brasil, marcone@iceb.ufop.br*

Palavras Chave: Otimização, Heurística Híbrida, Roteamento de Veículos com Janelas de Tempo.

Resumo. Este trabalho trata do Problema de Roteamento de Veículos com Janelas de Tempo. Tal problema consiste em uma variante do roteamento de veículos clássico, em que a demanda de cada cliente deve ser atendida durante um intervalo temporal pré-estabelecido. Dado o caráter altamente combinatório do problema, que pertence à classe NP-Difícil, sua resolução por abordagens puramente exatas é, em muitos casos, computacionalmente impraticável. Este fato motiva o desenvolvimento de algoritmos heurísticos para sua resolução, que são mais rápidos, porém não garantem a obtenção da melhor solução para o problema. Neste trabalho é proposto um algoritmo heurístico híbrido, que combina os métodos *Iterated Local Search*, *Variable Neighborhood Descent* e um procedimento exato de particionamento de conjunto. Esse procedimento de programação matemática é acionado periodicamente com vistas a combinar da melhor forma as rotas geradas ao longo do algoritmo. Os resultados computacionais obtidos mostram que o algoritmo é promissor, visto que ele obteve soluções melhores ou iguais aos da literatura em quarenta e dois dos cinquenta e seis problemas-teste analisados. Além disso, o algoritmo melhorou os resultados da literatura em 21,4% dos problemas tratados.

1 INTRODUÇÃO

O Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT) é um dos problemas mais importantes de otimização combinatória e mais estudados na literatura de pesquisa operacional. Neste problema, uma frota de veículos deve partir do depósito, atender a demanda dos consumidores e retornar ao depósito de forma que o custo total de viagem seja mínimo e o atendimento aconteça dentro do intervalo temporal especificado por cada consumidor. Além disso, deve-se respeitar a capacidade dos veículos.

Há duas motivações principais para o investimento em pesquisas que visam a proposição de novos algoritmos para a resolução desse problema: uma sob o ponto de vista operacional e outra relacionada a aspectos teóricos. No ponto de vista prático/operacional, os custos relacionados ao transporte de pessoas e/ou mercadorias geralmente são elevados, com tendência ao crescimento, motivado pela expansão atual das fronteiras comerciais de todo tipo de negócio (Alvarenga e Mateus, 2004). Pesquisas sugerem que de 10% a 15% do valor final das mercadorias comercializadas no mundo correspondem ao custo de seu transporte (King e Mast, 1997). Quanto aos aspectos teóricos, a versão de otimização do PRV e variantes, incluindo o PRVJT, pertence à classe NP-Difícil (Lenstra e Kan, 2006). Desta forma, a resolução eficiente destes problemas corresponde a um desafio para os pesquisadores, que em sua maioria optam por adotar abordagens heurísticas. Tal desafio é comprovado pela grande quantidade trabalhos que tratam da resolução desses problemas.

São diversos os objetivos dos autores ao tratar o PRVJT. No presente trabalho, toma-se como objetivo a minimização da distância total percorrida que, segundo de Oliveira e Vasconcelos (2008), é o mais comum e mais encontrado na literatura. Backer e Furnon (1997), Riise e Stølevik (1999), Kilby et al. (1999), Ombuki et al. (2006), Alvarenga et al. (2007) e de Oliveira e Vasconcelos (2008) também adotam o mesmo objetivo. Outros autores consideram a geração de um conjunto mínimo de rotas como objetivo primário e a minimização da distância como um objetivo secundário. Os seguintes trabalhos seguem esta ideia Bent e Hentenryck (2004), Homberger e Gehring (2005), Pisinger e Ropke (2007), Lim e Zhang (2007) e Prescott-Gagnon et al. (2009).

Dada a complexidade do problema, sua resolução por abordagens puramente exatas é, em muitos casos, uma tarefa extremamente árdua, pois demanda um tempo computacional muito elevado. Este fato motiva o desenvolvimento de novos algoritmos heurísticos para a resolução do PRVJT. Vale observar que tais algoritmos não garantem a obtenção da solução ótima, porém são mais rápidos e geralmente fornecem soluções próximas da ótima.

Neste trabalho é proposto um algoritmo híbrido combinando conceitos da metaheurística *Iterated Local Search*, do método *Variable Neighborhood Descent* e de um modelo exato de particionamento de conjuntos que, periodicamente, combina da melhor forma as rotas geradas ao longo do algoritmo.

O algoritmo proposto foi aplicado à resolução do conjunto de 56 instâncias de 100 consumidores proposto em Solomon (1987), que é bastante conhecido e amplamente explorado na literatura para testar a eficiência de novos métodos tanto exatos quanto heurísticos. Os resultados computacionais obtidos mostram que o algoritmo é promissor, visto que ele foi capaz de melhorar os resultados da literatura em pouco mais que 20% dos problemas-teste tratados.

Este trabalho está organizado em cinco seções incluindo esta introdução. A Seção 2 define o PRVJT. A Seção 3 apresenta em detalhes o algoritmo proposto. A Seção 4 contém os resultados obtidos. Por fim, a Seção 5 apresenta as considerações finais.

2 DEFINIÇÃO DO PROBLEMA

O PRVJT pode ser definido num grafo completo orientado $G = (V, A)$ em que $V = \{0, \ldots, n+1\}$ é o conjunto de vértices e $A = \{(i, j)|i, j \in V\}$ é o conjunto de arcos. A cada arco (i, j) é associado um tempo t_{ij} e um custo de viagem c_{ij}.

Neste ponto faz-se necessária uma definição precisa do termo *custo de viagem*. Em casos práticos, o custo de viagem pode considerar diversos fatores tais como: distância, tempo, desgaste do veículo ao percorrer determinado caminho, entre outros fatores. Entretanto, quando se trata de problemas teóricos envolvendo janelas de tempo, é comum converter todas as medidas relevantes em unidades de tempo para fins de padronização e também para facilitar a comparação entre diferentes métodos. Isto posto, adota-se aqui a mesma definição de custo que a maioria dos trabalhos teóricos da literatura, isto é, o custo de viagem consiste na distância convertida em unidades de tempo.

Ao todo, um conjunto K de veículos idênticos com capacidade Q devem atender n clientes, representados pelos vértices $1, \ldots, n$. Considera-se que $N = V - \{0, n+1\}$ representa o conjunto de clientes. Para atender tais clientes, os veículos devem partir do depósito e, após visitá-los, retornar ao mesmo local de onde partiram. Por questão de conveniência, o depósito é representado por dois vértices: o vértice 0, que representa a origem, e o vértice $n+1$, o destino. A cada cliente i, é associada uma demanda q_i que deve ser atendida por um único veículo. Além disso, todos os vértices possuem uma janela de tempo $[e_i, l_i]$, isto é, o serviço no vértice i deve ser iniciado dentro desse intervalo. Caso a chegada do veículo no cliente i ocorra antes do instante e_i, ele deve esperar a abertura da janela. O veículo não poderá chegar a i depois do instante l_i, pois isso faria violar a restrição de tempo do problema. Esse tipo de restrição é conhecido na literatura como janela de tempo rígida. A cada vértice é também associado um tempo de serviço, denotado por d_i. O objetivo é minimizar o custo total da rota $c(s)$, ou seja, minimizar a soma de todos os custos de viagem $\sum_{(i,j)\in s} c_{ij}$ que são associados aos arcos (i, j) presentes na solução s.

A formulação matemática do PRVJT, adaptada de Cordeau et al. (2001), é apresentada pelas expressões (1) a (9) a seguir. Nessa formulação, a variável binária x_{ijk} assume valor 1 se o veículo k passa pelo arco (i, j) e 0, caso contrário.

$$\text{Minimize} \sum_{k\in K} \sum_{i\in V} \sum_{j\in V} c_{ij} x_{ijk} \tag{1}$$

sujeito a:

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1, \forall i \in N \quad (2)$$

$$\sum_{j \in V} x_{0jk} = 1, \forall k \in K \quad (3)$$

$$\sum_{i \in V} x_{ijk} - \sum_{i \in V} x_{jik} = 0, \forall k \in K, \forall j \in N \quad (4)$$

$$\sum_{i \in V} x_{i(n+1)k} = 1, \forall k \in K \quad (5)$$

$$\sum_{i \in N} q_i \sum_{j \in V} x_{ijk} \leq Q, \forall k \in K \quad (6)$$

$$b_{ik} + d_i + t_{ij} - (1 - x_{ijk})M_{ij} \leq b_{jk}, \forall k \in K, \forall (i,j) \in A \quad (7)$$

$$e_i \leq b_{ik} \leq l_i, \forall k \in K, \forall i \in V \quad (8)$$

$$x_{ijk} \in \{0,1\}, \forall k \in K, \forall (i,j) \in A \quad (9)$$

A função objetivo (1) expressa o custo total a ser minimizado. As restrições (2) asseguram que somente um veículo k parte de cada cliente i. As restrições (3-5) garantem a continuidade do caminho a ser percorrido pelo veículo k, isto é, cada veículo parte do depósito, visita os clientes e em seguida retorna ao depósito. As restrições (6) fazem com que cada veículo k somente possa atender um conjunto de clientes cuja demanda total não ultrapasse sua capacidade Q. As restrições (7-8) asseguram a viabilidade das rotas no que diz respeito às restrições de janelas de tempo, onde b_{ik} representa o tempo em que o veículo k começa a atender o cliente i. As restrições (9) definem o domínio das variáveis de decisão.

3 METODOLOGIA

Esta seção apresenta em detalhes o algoritmo híbrido proposto. A Seção 3.1 apresenta a estrutura de dados utilizada para representar uma solução para o PRVJT. A Seção 3.2 descreve a função baseada em penalidades que avalia uma solução para o problema. Na Seção 3.3, é apresentado o procedimento usado para construir uma solução inicial. A Seção 3.4 descreve as estruturas de vizinhança utilizadas. A Seção 3.5 apresenta o algoritmo proposto.

3.1 Representação de uma solução

Uma rota r é definida por uma sequência de números inteiros, que correspondem aos identificadores dos clientes atendidos em r. Uma solução s consiste em um conjunto de rotas.

3.2 Função de avaliação

Uma solução s é avaliada pela função f, dada pela equação (10), a qual deve ser minimizada:

$$f(s) = \sum_{r \in s} g(r) = \sum_{r \in s} (c(r) + l(r).w_l + e(r).w_e) \quad (10)$$

em que: g é uma função que avalia rotas; $c(r)$ é o custo de deslocamento da rota r; $l(r)$ corresponde ao tempo de atraso em r; $e(r)$ é o excesso de carga na solução r; w_l e w_e são penalidades por unidade de atraso e excesso de carga, respectivamente (os quais foram fixados de forma empírica em $w_l = 200$ e $w_e = 300$). Observa-se que caso s seja viável, o valor reportado por f corresponde apenas ao custo total de deslocamento, pois, neste caso, $l(s) = e(s) = 0$.

3.3 Construção de uma solução inicial

Para obter uma solução inicial para o PRVJT foi desenvolvido um método de inserção mais barata que explora o Princípio da Otimalidade Próxima (Resende e Ribeiro, 2010). Segundo este princípio, em uma sequência ótima de escolhas, cada subsequência deve também ser ótima. Vale observar que, apesar deste princípio tratar de casos ótimos, no algoritmo desenvolvido não há nenhuma garantia da obtenção de soluções ótimas ou mesmo de partes ótimas de soluções. Neste caso, tal princípio é utilizado apenas com o intuito de obter soluções iniciais de maior qualidade. O pseudocódigo do método construtivo é apresentado na Figura 1.

Procedimento `IMB-POP`

```
1  Seja s_0 uma solução com |K| rotas vazias
2  para cada cliente c ∈ {1,...,n} faça
3      melhor_custo ← ∞
4      para cada rota r ∈ s_0 faça
5          para cada posição p da rota r em que se possa inserir o cliente c faça
6              custo ← custo da inserção do cliente c na posição p da rota r
7              se custo < melhor_custo então
8                  melhor_r ← r ; melhor_p ← p ; melhor_custo ← custo
9              fim
10         fim
11     fim
12     Adicione o cliente c na posição melhor_p da rota melhor_r da solução s_0
13     se (c mod ⌈n/5⌉) = 0 então
14         Refine a solução parcial incompleta s_0 por uma heurística h
15     fim
16 fim
17 return s_0
```

Figura 1: *Inserção Mais Barata com Princípio da Otimalidade (IMB-POP)*

Seja $|K|$ o número de veículos disponíveis para utilização. Inicialmente, o algoritmo construtivo cria $|K|$ rotas vazias (linha 1) e uma lista de candidatos a serem inseridos no conjunto de rotas (linha 2). Observe que inicialmente a lista de candidatos deve conter todos os clientes que devem ser atendidos pela frota. A ideia do algoritmo é inserir, passo a passo, cada cliente da lista de candidatos à sua melhor localização no momento (linha 12), efetuando periodicamente uma busca local na solução que está sendo construída (linha 14). O algoritmo de construção termina quando a lista de candidatos se torna vazia. Mais especificamente, os parâmetros da linha 13 foram dimensionados de forma que ocorram cinco buscas locais ao longo da construção, por exemplo, caso haja um total de 100 clientes a busca local é efetuada a cada vinte clientes adicionados à solução. Neste caso, a busca local é feita pelo algoritmo RVND (*vide* descrição na Seção 3.5.2).

3.4 Estruturas de vizinhança

Para explorar o espaço de soluções, são utilizadas dez estruturas de vizinhança, das quais seis alteram duas rotas a cada movimento efetuado (inter-rota) e quatro alteram apenas uma rota (intra-rota). As vizinhanças inter-rotas são geradas a partir dos seguintes movimentos: **Shift**$(1,0)$, **Shift**$(2,0)$, **Shift**$(3,0)$, **Swap**$(1,1)$, **Swap**$(2,1)$ e **Swap**$(2,2)$. Um movimento da

estrutura **Shift**(k) consiste na transferência de k clientes adjacentes de uma rota r_1 para uma rota r_2 e um movimento do tipo **Swap**(k, l) permuta k clientes adjacentes de uma rota r_1 a l outros clientes adjacentes de outra rota r_2.

Quanto àquelas estruturas de vizinhança que alteram apenas uma rota por vez, são utilizadas as seguintes: **Exchange**, **Shift'**(1), **Shift'**(2) e **Shift'**(3). A vizinhança **Exchange** consiste na permutação entre dois clientes da mesma rota e, assim, é a versão intra-rota da vizinhança **Swap**$(1, 1)$. As outras três vizinhanças podem ser consideradas como versões intra-rotas das vizinhanças **Shift**$(1, 0)$, **Shift**$(2, 0)$ e **Shift**$(3, 0)$, respectivamente.

3.5 O algoritmo proposto

O algoritmo proposto, denotado por IILS-SP, consiste na construção de uma solução inicial pelo procedimento apresentado na Seção 3.3, seguida de um refinamento que combina versões adaptadas dos métodos *Iterated Local Search* (ILS) e *Variable Neighborhood Descent* (VND) com uma abordagem exata baseada na formulação matemática do problema de particionamento de conjunto (*Set Partition* - SP). O pseudocódigo do algoritmo proposto é apresentado pela Figura 2. Nesta figura: s_0 é uma solução inicial; s^* é a melhor solução obtida durante a execução do procedimento; s' é a solução perturbada e s'' é um ótimo local obtido pela aplicação da busca local RVND à solução perturbada.

Procedimento `IILS-SP`

```
1  s_0 ← IMB-POP()
2  s* ← RVND(s_0)
3  repita
4  |   s' ← Perturbação(s*, histórico)
5  |   s'' ← RVND(s')
6  |   se MomentoApropriado(histórico) então
7  |   |   s'' ← Intensificação (s'')
8  |   fim
9  |   s* ← CritérioAceitação(s'', s*, histórico)
10 até critério de parada satisfeito
11 return s*
```

Figura 2: *Intensified Iterated Local Search (IILS-SP)*

As seções seguintes detalham cada parte desse algoritmo.

3.5.1 *Intensified Iterated Local Search*

Intensified Iterated Local Search é uma extensão da metaheurística *Iterated Local Search* – ILS (Lourenço et al., 2003). ILS explora o espaço de soluções por meio da aplicação de perturbações à solução ótima local corrente. Em linhas gerais, tal metaheurística parte de uma solução inicial s_0 e a esta aplica uma busca local, obtendo s^*. Posteriormente, o método efetua iterativamente os seguintes passos: (*i*) perturbação na solução corrente s^*, obtendo uma solução s' e (*ii*) busca local em s', obtendo uma solução s'', que é um ótimo local. Caso s'' seja melhor que a solução corrente s^* (isto é, melhora a melhor solução obtida até o momento), o método torna s'' a nova solução corrente. Caso contrário, o método efetua uma nova iteração. Este procedimento é repetido até que um critério de parada seja satisfeito.

É importante salientar que o sucesso do ILS depende fortemente das perturbações realizadas. Portanto, a perturbação aplicada a uma dada solução deve ser dosada de maneira tal que as alterações decorrentes de sua aplicação sejam suficientes para se escapar de ótimos locais e explorar diferentes regiões sem, contudo, causar um caos absoluto que implicaria na perda de características do ótimo local corrente.

Na adaptação feita, as perturbações realizadas (linha 4 da Figura 2) consistem na aplicação de $p+2$ movimentos aleatoriamente escolhidos na vizinhança **Shift**$(1,0)$, apresentada na Seção 3.4, onde $p \in \{0,1,2,\dots\}$ representa o nível de perturbação. Logo, quanto maior for este valor, maior será o número de modificações feitas em na solução. Neste trabalho, em um mesmo nível de perturbação são aplicadas *ILSmax* iterações sem melhora. Tão logo este valor seja alcançado, o nível de perturbação é aumentado.

Por outro lado, a busca local do ILS (linhas 2 e 5 da Figura 2) é feita pelo método *Variable Neighborhood Descent* com exploração randômica de vizinhanças, denotado por RVND e descrito na Seção 3.5.2.

Finalmente, o algoritmo proposto contém um módulo de intensificação (linha 7 da Figura 2). Este módulo é acionado em momentos "apropriados" da busca e consiste na chamada de um procedimento de programação matemática, baseado em Particionamento de Conjuntos, para encontrar o melhor conjunto de rotas dentre aquelas geradas ao longo da busca. Mais especificamente, o modelo de particionamento é aplicado ao conjunto formado por todas as rotas pertencentes às soluções geradas após a fase de busca local do algoritmo IILS. Isto é, a cada iteração do IILS, as rotas de s'' (linha 5 da Figura 2) são adicionadas ao conjunto a ser particionado. Observa-se que isto é feito de forma que não haja rotas repetidas neste conjunto, o qual possui tamanho ilimitado.

A descrição desse módulo é feita na Seção 3.5.3.

3.5.2 *Variable Neighborhood Descent* com exploração randômica de vizinhanças

O procedimento *Variable Neighborhood Descent* (VND), proposto por Mladenovic e Hansen (1997), consiste em explorar exaustivamente o espaço de soluções por meio de trocas sistemáticas de estruturas de vizinhança. No decorrer da busca, somente são aceitas as soluções de melhora em relação à solução corrente. Quando uma solução melhor é encontrada, o método retoma sua busca partindo da primeira estrutura de vizinhança.

O método VND baseia-se em três princípios: (*i*) um ótimo local com relação a uma dada estrutura de vizinhança não corresponde necessariamente a um ótimo local com relação a uma outra estrutura de vizinhança; (*ii*) um ótimo global corresponde a um ótimo local para todas as estruturas de vizinhança; e (*iii*) para muitos problemas, ótimos locais com relação a uma ou mais estruturas de vizinhança são relativamente próximas.

O último princípio, de natureza empírica, indica que um ótimo local frequentemente fornece algum tipo de informação sobre o ótimo global. Esse é o caso em que os ótimos local e global compartilham muitas variáveis com o mesmo valor, o que sugere uma investigação sistemática da vizinhança de um ótimo local até a obtenção de uma nova solução de melhor valor.

Na versão clássica, o VND efetua buscas por ótimos locais seguindo uma ordem pré-estabelecida de estruturas de vizinhança. Esta estratégia é vastamente aplicada na literatura e sua eficácia é comprovada em diversos trabalhos. No presente trabalho, porém, opta-se por não fixar a ordem de exploração da vizinhança. Dessa forma, ocorre maior diversidade durante a busca. Tal estratégia, adotada com sucesso em Subramanian et al. (2010), é denominada *Variable Neighborhood Descent with random neighborhood ordering*.

3.5.3 Modelo de particionamento de conjunto

A etapa de intensificação do algoritmo proposto consiste na resolução exata, por um resolvedor matemático, do PRVJT modelado como um Problema de Particionamento de Conjuntos (PPC). Seja $\mathcal{R}$ o conjunto de rotas e y_j, $\forall j \in \mathcal{R}$, as variáveis binárias que indicam se a rota $j \in \mathcal{R}$ faz parte da solução ($y_j = 1$) ou não ($y_j = 0$). A cada rota $j \in \mathcal{R}$ tem-se um custo g_j associado. O parâmetro m_{ij} é igual a 1 se o cliente $i \in N$ for atendido pela rota $j \in \mathcal{R}$ e 0 caso contrário. O modelo matemático do PRVJT na forma de um PPC utilizado neste trabalho é apresentado pelas expressões (11) a (14).

$$Minimize \sum_{j \in \mathcal{R}} g_j y_j \tag{11}$$

sujeito a:

$$\sum_{j \in \mathcal{R}} m_{ij} y_j = 1, \forall i \in N \tag{12}$$

$$\sum_{j \in \mathcal{R}} y_j \leq |K| \tag{13}$$

$$y_j \in \{0, 1\}, \forall j \in \mathcal{R} \tag{14}$$

O objetivo dessa formulação é encontrar o conjunto de rotas de custo mínimo (expressão 11) atendendo às restrições do problema. As restrições (12) garantem que exatamente uma rota passa por cada cliente e as restrições (13) asseguram que a solução contém no máximo $|K|$ rotas.

São inúmeros os problemas em otimização combinatória que podem ser descritos como um PPC. Especificamente para roteamento de veículos, o modelo tem sido adotado com sucesso por diversos autores, entre eles: Agarwal et al. (1989), Desrosiers et al. (1984), Kohl (1995) e Larsen (1995). Os dois últimos abordaram especificamente o PRVJT, obtendo novos resultados ótimos para as instâncias de Solomon (1987) até então desconhecidos.

Neste trabalho, o modelo em questão foi implementado utilizando a API Concert para C++ e resolvido, ao longo do algoritmo, pelo otimizador CPLEX, versão 11.

4 RESULTADOS

O algoritmo proposto (IILS-SP) foi desenvolvido na linguagem de programação C++ e os testes foram efetuados em um computador com microprocessador Intel Quad Core 2.4 GHz com 8 GB de memória RAM utilizando o sistema operacional Ubuntu Linux 9.10 Kernel 2.6.31.

O IILS-SP foi aplicado à resolução do conjunto de problemas-teste proposto por Solomon (1987), que é bastante conhecido e amplamente explorado na literatura. Este conjunto consiste em 56 problemas-testes de 100 clientes e é dividido em seis grupos: C1, C2, R1, R2, RC1 e RC2. Os problemas contidos nos grupos C1 e C2 possuem clientes alocados de forma geograficamente agrupada, enquanto que em R1 e R2 os clientes são alocados em posições aleatórias. Os problemas de RC1 e de RC2 mesclam agrupamento e aleatoriedade de localização de clientes. Além disso, os grupos C2, R2 e RC2 possuem horizontes de planejamento mais longos e os veículos são de maior capacidade que em C1, R1 e RC1. Isto significa que cada veículo de C2, R2 e RC2 é capaz de atender um número maior de clientes.

Foram feitas 5 execuções do algoritmo para cada um dos 56 problemas-teste utilizando como critério de parada 10 minutos de processamento para cada execução. O algoritmo foi calibrado

empiricamente e os parâmetros foram fixados como segue. Na construção de uma solução inicial, ao longo das inserções dos clientes são efetuadas 5 buscas locais conforme descreve a Seção 3.3. O número de iterações sem melhora em um dado nível de perturbação do IILS foi fixado em 20. O procedimento é executado iterativamente, seguindo a mesma ideia do método *Multi-Start* (Martí, 2003). Isto é, a cada iteração a solução inicial é construída pelo método não determinístico descrito na Seção 3.3 e a busca local é feita pelo IILS-SP. Além disto, o tempo máximo de processamento de cada chamada do resolvedor matemático na etapa de intensificação foi limitado a 5 segundos.

As Tabelas 1, 2, 3, 4, 5 e 6 apresentam os melhores resultados disponíveis na literatura para cada um dos problemas-teste analisados e os resultados obtidos pela aplicação do algoritmo proposto. A primeira coluna indica o nome do problema. As três seguintes fornecem respectivamente o número de veículos ("$|K|$"), a distância total percorrida ("Distância") da melhor solução obtida para o problema e o primeiro trabalho que a obteve ("Trabalho"). O próximo grupo de colunas apresenta os resultados obtidos pela aplicação do método proposto, apresenta-se o número de veículos e a distância da melhor solução obtida nas 5 execuções do algoritmo, a média das distâncias obtidas ("Média") e o desvio da média em relação à melhor solução ("Desvio"), conforme a seguinte definição: $Desvio = (Media - Melhor)/Melhor$, sendo $Melhor$ o melhor valor encontrado para o problema-teste em questão. A última coluna apresenta a melhora do valor das soluções obtidas pelo IILS-SP em relação às melhores soluções da literatura, de acordo com $Melhora = (MelhorLit - Melhor)/Melhor$. Os resultados de melhora e os empates estão destacados na tabela com sublinhado e negrito, respectivamente.

Problema	Melhores da literatura			IILS-SP				Melhora
	$\|K\|$	Distância	Trabalho	$\|K\|$	Distância	Média	Desvio	
C101	10	**828,94**	RT95	10	**828,94**	**828,94**	0,00%	0,00%
C102	10	**828,94**	RT95	10	**828,94**	**828,94**	0,00%	0,00%
C103	10	**828,06**	RT95	10	**828,06**	**828,06**	0,00%	0,00%
C104	10	**824,78**	RT95	10	**824,78**	**824,78**	0,00%	0,00%
C105	10	**828,94**	RT95	10	**828,94**	**828,94**	0,00%	0,00%
C106	10	**828,94**	RT95	10	**828,94**	**828,94**	0,00%	0,00%
C107	10	**828,94**	RT95	10	**828,94**	**828,94**	0,00%	0,00%
C108	10	**828,94**	RT95	10	**828,94**	**828,94**	0,00%	0,00%
C109	10	**828,94**	RT95	10	**828,94**	**828,94**	0,00%	0,00%

Tabela 1: Resultados para o grupo de problemas C1

Problema	Melhores da literatura			IILS-SP				Melhora
	$\|K\|$	Distância	Trabalho	$\|K\|$	Distância	Média	Desvio	
C201	3	**591,56**	RT95	3	**591,56**	**591,56**	0,00%	0,00%
C202	3	**591,56**	RT95	3	**591,56**	**591,56**	0,00%	0,00%
C203	3	**591,17**	RT95	3	**591,17**	**591,17**	0,00%	0,00%
C204	3	**590,60**	RT95	3	**590,60**	596,42	0,99%	0,00%
C205	3	**588,88**	RT95	3	**588,88**	**588,88**	0,00%	0,00%
C206	3	**588,49**	RT95	3	**588,49**	**588,49**	0,00%	0,00%
C207	3	**588,29**	RT95	3	**588,29**	**588,29**	0,00%	0,00%
C208	3	**588,32**	RT95	3	**588,32**	**588,32**	0,00%	0,00%

Tabela 2: Resultados para o grupo de problemas C2

Problema	Melhores da literatura			IILS-SP				Melhora
	$\|K\|$	Distância	Trabalho	$\|K\|$	Distância	Média	Desvio	
R101	20	**1642,88**	AM04	20	**1642,88**	**1642,88**	0,00%	0,00%
R102	18	**1472,62**	AM04	18	1472,81	1472,81	0,01%	-0,01%
R103	14	**1213,62**	RT95	14	**1213,62**	1214,40	0,06%	0,00%
R104	11	**986,10**	AM07	11	**982,30**	988,65	0,26%	0,39%
R105	15	**1360,78**	AM07	15	**1360,78**	**1360,78**	0,00%	0,00%
R106	13	**1241,52**	AM07	13	**1239,37**	1242,48	0,08%	0,17%
R107	11	**1076,13**	AM07	11	**1075,21**	1076,23	0,01%	0,09%
R108	10	**948,57**	AM07	10	951,22	955,39	0,72%	-0,28%
R109	13	**1151,84**	AM07	13	**1151,84**	1152,06	0,02%	0,00%
R110	11	**1080,36**	RT95	12	**1072,41**	**1072,41**	0,00%	0,74%
R111	12	**1053,50**	AM07	12	**1053,50**	1054,43	0,09%	0,00%
R112	10	**953,63**	RT95	10	956,36	961,66	0,84%	-0,29%

Tabela 3: Resultados para o grupo de problemas R1

Problema	Melhores da literatura			IILS-SP				Melhora
	$\|K\|$	Distância	Trabalho	$\|K\|$	Distância	Média	Desvio	
R201	8	**1147,80**	OV08	8	**1147,80**	1149,94	0,19%	0,00%
R202	8	**1039,32**	OV08	8	**1034,35**	**1039,19**	0,47%	0,48%
R203	6	**874,87**	OV08	6	881,12	892,38	2,00%	-0,71%
R204	5	**735,80**	OV08	4	745,12	759,48	3,22%	-1,25%
R205	5	**954,16**	OV08	5	955,96	967,51	1,40%	-0,19%
R206	5	**884,25**	OV08	5	**879,89**	891,15	0,78%	0,50%
R207	4	**797,99**	OV08	4	808,23	820,73	2,85%	-1,27%
R208	4	**705,62**	OV08	3	724,98	734,94	4,16%	-2,67%
R209	5	**860,11**	OV05	5	**859,39**	866,29	0,72%	0,08%
R210	5	**910,98**	OV08	7	914,84	919,89	0,98%	-0,42%
R211	4	**755,82**	OV08	4	762,38	772,97	2,27%	-0,86%

Tabela 4: Resultados para o grupo de problemas R2

Problema	Melhores da literatura			IILS-SP				Melhora
	$\|K\|$	Distância	Trabalho	$\|K\|$	Distância	Média	Desvio	
RC101	15	**1623,58**	RT95	15	**1623,58**	1626,82	0,20%	0,00%
RC102	14	**1466,84**	AM07	14	**1461,23**	1472,15	0,36%	0,38%
RC103	11	**1261,67**	S98	11	1262,02	1273,51	0,94%	-0,03%
RC104	10	**1135,48**	C00	10	1135,52	1135,79	0,03%	0,00%
RC105	16	**1518,60**	AM04	16	**1518,58**	**1518,58**	0,00%	0,00%
RC106	13	**1377,35**	AM04	13	**1376,99**	1378,25	0,07%	0,03%
RC107	12	**1212,83**	AM04	12	**1212,83**	**1212,83**	0,00%	0,00%
RC108	11	**1117,53**	AM04	11	**1117,53**	1120,99	0,31%	0,00%

Tabela 5: Resultados para o grupo de problemas RC1

Problema	Melhores da literatura			IILS-SP				Melhora
	$\|K\|$	Distância	Trabalho	$\|K\|$	Distância	Média	Desvio	
RC201	9	**1266,11**	OV08	9	**1265,90**	1268,59	0,20%	0,02%
RC202	8	**1096,75**	OV05	8	**1095,64**	1099,17	0,22%	0,10%
RC203	5	**926,89**	OV08	5	937,45	949,94	2,49%	-1,13%
RC204	4	**786,38**	OV08	4	796,55	810,88	3,12%	-1,28%
RC205	7	**1157,55**	OV08	7	1157,66	1161,82	0,37%	-0,01%
RC206	6	**1056,21**	OV08	6	1069,00	1076,84	1,95%	-1,20%
RC207	6	**966,08**	OV08	6	**966,08**	982,07	1,66%	0,00%
RC208	4	**779,84**	OV05	5	785,07	788,78	1,15%	-0,67%

Tabela 6: Resultados para o grupo de problemas RC2

Trabalho		C1	C2	R1	R2	RC1	RC2	Total
(1) Caseau *et al*. (1999)	NV	10,00	3,00	12,42	3,09	12,00	3,38	420
	DT	**828,38**	596,63	1233,34	990,99	1403,74	1220,99	58927
(2) Kilby *et al*. (1999)	NV	10,00	3,00	12,67	3,00	12,13	3,38	423
	DT	830,75	592,24	1200,33	966,56	1388,15	1133,42	57423
(3) Riise e Stlevik (1999)	NV	10,56	3,88	13,92	4,91	13,75	5,63	502
	DT	846,88	598,70	1211,22	917,54	1399,76	1055,61	56682
(4) Schrimpf et al. (2000)	NV	10,00	3,00	12,08	2,82	11,88	3,38	412
	DT	**828,38**	**589,86**	1211,53	949,27	1361,76	1097,63	56830
(5) Cordone *et al*. (2001)	NV	10,00	3,00	12,50	2,91	12,38	3,38	422
	DT	834,05	591,78	1241,89	995,39	1408,87	1139,70	58481
(6) Braysy (2003)	NV	10,00	3,00	12,17	2,82	11,88	3,25	412
	DT	832,88	593,49	1253,24	1039,56	1408,44	1244,96	59945
(7) Fraga (2006)	NV	10,00	3,00	13,83	5,18	13,38	6,13	493
	DT	**828,38**	**589,86**	1204,42	901,67	1374,29	1037,74	55809
(8) Alvarenga *et al*. (2007)	NV	10,00	3,00	13,25	5,55	12,88	6,50	489
	DT	**828,38**	**589,86**	1183,38	899,90	1341,67	1015,90	55134
(9) de Oliveira e	NV	10,00	3,00	13,33	5,36	13,25	6,13	488
Vasconcelos (2008)	DT	**828,38**	**589,86**	1186,94	**878,79**	1362,44	**1004,59**	55020
(10) Presente trabalho	NV	10,00	3,00	13,17	5,36	12,75	6,13	482
	DT	**828,38**	**589,86**	**1181,03**	883,10	**1338,54**	1009,17	**54842**

Tabela 7: Comparação entre diferentes trabalhos que otimizam a distância total percorrida

Em suma, quanto às melhores soluções encontradas ao longo das execuções, o algoritmo proposto obteve: 100% (9/9) de empates para C1; 100% (8/8) de empates para C2; 33,3% (4/12) de melhoras e 41,6% (5/12) de empates para R1; 27,3% (3/11) de melhoras e 9,1% (1/11) de empates para R2; 37,5% (3/8) de melhoras e 37,5% (3/8) de empates para RC1; e 25% (2/8) de melhoras e 12,5% (1/8) de empates para RC2. Ao todo em 21,4% (12/56) dos casos houve melhora, em 48,2% (27/56) houve empates e em 30,4% (17/56) dos casos houve derrotas ao se considerar todos os grupos.

O algoritmo se mostrou robusto, visto que ele apresentou desvios relativamente baixos. Em 80,4% (45/56) dos problemas-teste analisados, o desvio foi inferior a 1.0%. Quando extrapolou esse valor, o desvio não passou de 4,16% (como é o caso em R208). Dessa forma, pode-se concluir que o algoritmo produz soluções finais com baixa variabilidade em relação à qualidade. Além disso, em alguns casos (R110, R202 e RC105) o algoritmo obteve soluções em média melhores que as melhores soluções da literatura.

A Tabela 7 apresenta os resultados de diferentes trabalhos que têm como objetivo primário a minimização da distância total percorrida. Os resultados de cada trabalho são apresentados em duas linhas. A linha superior ("NV") indica o número de veículos e a inferior ("DT") indica a distância percorrida. Nas colunas de "C1" a "RC2" são apresentados os resultados para cada um dos grupos de problemas-teste analisados. Cada célula de tais colunas apresentam a média do número de veículos e a média das distâncias das melhores soluções de cada grupo. A última coluna apresenta o número total veículos e a distância total percorrida considerando-se todos os problemas, isto é, os valores desta coluna são calculados somando-se o número de veículos e a distância percorrida das melhores soluções obtidas pelo IILS-SP para os 56 problemas-teste de Solomon (1987). Ao se observar os resultados separadamente para cada grupo, conclui-se que o algoritmo empatou com os melhores resultados nos grupos clusterizados (C1 e C2) e superou todos os trabalhos nos grupos R1 e RC1. Porém, apesar de apresentar resultados próximos aos melhores nos grupos R2 e RC2, ele não foi capaz de superá-los nesses dois grupos. Por outro lado, ao se considerar todos os grupos, o IILS-SP superou os demais algoritmos.

5 CONCLUSÃO

Neste trabalho, é proposto um algoritmo heurístico híbrido para a resolução do Problema de Roteamento de Veículos com Janelas de Tempo. O algoritmo proposto combina as metaheurísticas *Iterated Local Search*, *Variable Neighborhood Descent* e um modelo exato de particionamento de conjunto que, de tempos em tempos, combina da melhor forma as rotas geradas ao longo do algoritmo.

Dado o caráter combinatório do problema, que pertence à classe NP-Difícil, sua resolução por abordagens puramente exatas é computacionalmente impraticável, pois levaria muito tempo. Esse fato motiva o desenvolvimento de algoritmos heurísticos para sua resolução, que são mais rápidos, mas que, entretanto, não garantem a obtenção da melhor solução para o problema. O algoritmo proposto, ao combinar a flexibilidade dos métodos heurísticas e o poderio dos métodos de programação matemática, busca usar o que há de melhor nas abordagens heurística e exata.

O algoritmo desenvolvido foi aplicado aos 56 problemas-teste de Solomon e seus resultados foram comparados com os melhores da literatura. Os resultados obtidos mostram que o algoritmo é promissor, visto que ele obteve soluções melhores ou iguais à da literatura em quarenta e dois problemas-teste. Além disso o algoritmo melhorou os resultados da literatura em 21,4% dos problemas-teste analisados.

REFERÊNCIAS

Agarwal Y., Mathur K., e Salkin H.M. A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks*, 19:731–740, 1989.

Alvarenga G.B. e Mateus G.R. A two-phase genetic and set partitioning approach for the vehicle routing problem with time windows. In *HIS '04: Proceedings of the Fourth International Conference on Hybrid Intelligent Systems*, páginas 428–433. IEEE Computer Society, Washington, DC, USA, 2004. ISBN 0-7695-2291-2. doi:http://dx.doi.org/10.1109/ICHIS.2004.13.

Alvarenga G.B., Mateus G.R., e de Tomi G. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers and Operations Research*, 34:1561–1584, 2007.

Backer B.D. e Furnon V. Meta-heuristics in constraint programming experiments with tabu search on the vehicle routing problem. In *MIC'97: Proceedings of the Second International Conference on Metaheuristics*. Sophia Antipolis, France, 1997.

Bent R. e Hentenryck P.V. A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science*, 38(4):515–530, 2004. ISSN 1526-5447.

Cordeau J.F., Desaulniers G., Desrosiers J., Solomon M.M., e Soumis F. *The Vehicle Routing Problem*, capítulo The VRP with Time Windows, páginas 157–193. Paolo Toth and Daniele Vigo, SIAM Monographs on Discrete Mathematics and Applications, 2001.

de Oliveira H. e Vasconcelos G. A hybrid search method for the vehicle routing problem with time windows. *Annals of Operations Research*, 2008. doi:10.1007/s10479-008-0487-y.

Desrosiers J., Soumis F., e Desrochers M. Routing with time windows by column generation. *Networks*, 14(4):545–565, 1984.

Homberger J. e Gehring H. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, 162(1):220–238, 2005.

Kilby P., Prosser P., e Shaw P. Guided local search for the vehicle routing problem with time windows. In *META-HEURISTICS advances and trends in local search paradigms for optimization*, páginas 473–486. S. Voss, S. Martello, I. H. Osman, & C. Roucairol (Eds.), Boston: Kluwer Academic, 1999.

King G.F. e Mast C.F. Excess travel: causes. extent and consequences. *Transportation Research Record*, (1111):126–134, 1997.

Kohl N. *Exact methods for Time Constained Routing and Related Scheduling Problems*. Tesis de Doutorado, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1995.

Larsen J. *Parallelization of the vehicle routing problem with time windows*. Tesis de Doutorado, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1995.

Lenstra J.K. e Kan A.H.G.R. *Complexity of vehicle routing and scheduling problems*. Wiley Subscription Services, Inc., A Wiley Company, 2006.

Lim A. e Zhang X. A two-stage heuristic with ejection pools and generalized ejection chains for the vehicle routing problem with time windows. *INFORMS J. on Computing*, 19(3):443–457, 2007. ISSN 1526-5528.

Lourenço H.R., Martin O.C., e Stützle T. Iterated local search. In F. Glover e G. Kochenberger, editores, *Handbook of Metaheuristics*, capítulo 11. Kluwer Academic Publishers, Boston, 2003.

Martí R. Multi-start methods. In F. Glover e G. Kochenberger, editores, *Handbook of Metaheuristics*, capítulo 12. Kluwer Academic Publishers, Boston, 2003.

Mladenovic N. e Hansen P. A variable neighborhood search. *Computers and Operations Research*, 24:1097–1100, 1997.

Ombuki B., Ross B.J., e Hanshar F. Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, 24:17–30, 2006.

Pisinger D. e Ropke S. A general heuristic for vehicle routing problems. *Comput. Oper. Res.*, 34(8):2403–2435, 2007. ISSN 0305-0548.

Prescott-Gagnon E., Desaulniers G., e Rousseau L.M. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Netw.*, 54(4):190–204, 2009. ISSN 0028-3045. doi:http://dx.doi.org/10.1002/net.v54:4.

Resende M.G.C. e Ribeiro C.C. Grasp. In E.K. Burke e G. Kendall, editores, *Search Methodologies*. Springer (to appear), 2 edição, 2010. Available at: http://www.ic.uff.br/∼celso/artigos/grasp.pdf.

Riise A. e Stølevik M. Implementation of guided local search for the vehicle routing problem. Relatório Técnico, Department of Computer Science, Michigan State University, SINTEF Applied Mathematics, Norway, 1999.

Solomon M.M. Algorithms for the vehicle routing and scheduling problem with time window contraints. *Operational Research*, 35:254–265, 1987.

Subramanian A., Drummond L., Bentes C., Ochi L., e Farias R. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers and Operations Research*, 37:1899–1911, 2010.