

UM ESTUDO DA SOLUÇÃO DO PROBLEMA DE ROTEAMENTO ABERTO DE VEÍCULOS UTILIZANDO METAHEURÍSTICAS

José Maurício Costa

Centro Federal de Educação Tecnológica de Minas Gerais
jmcosta25@hotmail.com

Marcelus Xavier Oliveira

Centro Federal de Educação Tecnológica de Minas Gerais
marcelusxavier@gmail.com

Sérgio Ricardo de Souza

Centro Federal de Educação Tecnológica de Minas Gerais
sergio@dppg.cefetmg.br

Marcene Jamilson Freitas Souza

Universidade Federal de Ouro Preto
marcone@iceb.ufop.br

Resumo

Este artigo mostra a aplicação de metaheurísticas para resolver o Problema de Roteamento Aberto de Veículos com Janelas de Tempo (PRAVJT). Este é um problema NP-difícil, que consiste em definir o menor número de veículos para atender um conjunto de clientes, de forma que a distância total percorrida por eles também seja minimizada. A principal característica do PRAVJT está no fato dos veículos não retornarem ao depósito após atenderem os últimos clientes de suas rotas. Para sua solução, foram aplicadas duas heurísticas de construção de rotas, sendo uma oriunda da própria literatura e, outra, uma proposta original deste artigo. Ambas são testadas em conjunto com a metaheurística Iterated Local Search (ILS), utilizando instâncias da literatura. Estes testes mostram que os desenvolvimentos aqui propostos, em especial a nova heurística de construção de rotas, são promissores, obtendo-se resultados competitivos com os melhores da literatura.

Palavras-Chaves: Logística e Transportes; Otimização; Metaheurísticas.

Abstract

This paper presents the application of metaheuristics to solve the Open Vehicle Routing Problem with Time Windows (OVRPTW). This is a NP-hard problem. The objective is determining the minimum number of vehicles to meet a set of clients, so that the total distance traveled by them is also minimized. The main feature of OVRPTW lies in the fact that the vehicles do not return to the deposit after the last meet customers on their routes. To solve this problem, two route construction heuristics are applied, one coming from the literature itself and the other being a new proposal, presented in this article. Both are tested together with the metaheuristic Iterated Local Search (ILS). Tests were performed using instances from the literature. These tests showed that the development presented here, especially the new route construction heuristic, are promising, obtaining results competitive with the top of the literature.

Keywords: Logistica and Transport; Optimization; Metaheuristics;

1. INTRODUÇÃO

De acordo com Li *et al.* (2007), com a finalidade de reduzir os custos acerca do transporte de seus bens, muitas empresas contratam serviços de transporte terceirizados para realizar este trabalho. Assim, a empresa contratante não precisa se preocupar se os veículos irão retornar para o ponto de partida. Problemas como este caracterizam o Problema de Roteamento Aberto de Veículos (PRAV). O PRAV é uma variante do PRV clássico, que se difere basicamente devido ao fato do veículo não retornar ao depósito após servir o último cliente. Outra diferença entre o PRAV e o PRV é que, no caso do primeiro, as rotas são representadas por caminhos Hamiltonianos, ao invés de serem por ciclos, como é no PRV. Conforme Brandão (2004), o PRAV é um problema de otimização combinatória NP-difícil, que consiste em definir o menor número de veículos para atender um conjunto de clientes, de forma que a distância total percorrida por eles também seja minimizada. Além disso, este conjunto de cidades possui demanda e localização geográfica conhecidas. Neste trabalho, é estudada uma variante do PRAV, que é o Problema de Roteamento Aberto de Veículos com Janelas de Tempo (PRAVJT). No PRAVJT, além das restrições do PRAV, são consideradas restrições definidas por janelas de tempo, que indicam os períodos de atendimento dos clientes.

Segundo Brandão (2004), o PRAV foi mencionado primeiramente no trabalho desenvolvido por Schrage (1981), que descrevia problemas de roteamento reais e suas aplicações. Brandão (2004) desenvolveu em sua pesquisa um algoritmo baseado na metaheurística Busca Tabu para explorar a estrutura do PRAV. Este algoritmo utiliza os métodos *Nearest Neighbour Heuristic* e *Unstringing and Stringing* para a geração da solução inicial. Letchford *et al.* (2006) implementaram em seu trabalho um algoritmo exato para resolver o PRAV baseado no método *Branch-and-Cut*. Repoussis *et al.* (2006) usaram uma heurística de construção de rotas gulosa para solucionar o PRAVJT. Esta heurística utiliza as informações das janelas de tempo integrantes do problema através da combinação da seleção dos clientes e do critério de inserção de rotas. Guiyun (2009) desenvolveu um algoritmo baseado na metaheurística Colônia de Formigas para solucionar o PRAVJT, em que o processo de construção das soluções é feito pelas formigas de forma paralela. Repoussis *et al.* (2009) propuseram um algoritmo evolucionário para resolver o PRAVJT.

Neste estudo, é utilizado um novo algoritmo para a construção de rotas além das heurísticas PFIH e ILS para tentar solucionar o PRAVJT, em que o algoritmo desenvolvido e o PFIH foram combinados com o ILS. O artigo está organizado da seguinte forma: a seção 2 apresenta a descrição do problema. A seção 3 mostra a metodologia usada no trabalho. Os experimentos computacionais são discutidos na seção 4. As conclusões estão na seção 5.

1. CARACTERIZAÇÃO DO PROBLEMA

Para Li *et al.* (2009), o PRAVJT pode ser representado na forma de um grafo completo e não - direcionado, que possui um conjunto V de vértices e um conjunto A de arestas. Sendo o conjunto de vértices do grafo $V = \{0; 1; \dots; N\}$, o que inclui o depósito, e os demais vértices são os clientes a serem atendidos pelos veículos a partir do depósito. O conjunto A de arestas é definido como $A = \{(i; j) \mid i; j \in V; i \neq j\}$. Cada aresta $(i; j)$ está associada a uma distância d_{ij} entre os vértices respectivos. O conjunto de veículos é definido por K , em que o índice k conta os veículos, assumindo valores que vão de 1 a $|K|$. O conjunto de clientes é $C = \{1; 2; \dots; N\}$, em que cada cliente i possui uma demanda q_i , que deve ser atendida por apenas um dos veículos. No PRAVJT todos os veículos possuem a mesma capacidade Q de carga. Além disso, cada veículo deve atender um subconjunto de clientes em

sua rota, que começa no depósito e termina no último cliente. De acordo com MirHassani e Abolghasemi (2011), o PRAVJT é um problema de otimização que consiste na minimização do número de veículos que são necessários para atender aos clientes, para que, em seguida, seja feita a minimização da distância total de viagem. Nesse problema, cada cliente é visitado apenas uma vez por um veículo, e a capacidade de cada veículo não pode ser ultrapassada. Conforme Repoussis *et al.* (2006), o PRAVJT se difere do PRAV devido a adição das restrições de tempo que fazem parte dele. No PRAVJT, cada demanda q_i referente a um cliente C_i deve ser atendida de acordo com a sua respectiva janela de tempo, que modela o intervalo de tempo $[e_i; l_i]$ em que tal cliente pode ser atendido. Os valores e_i e l_i são os tempos inicial e final referentes ao período de atendimento do cliente, respectivamente. Além disso, cada cliente requer um tempo de serviço s_i , que é o período de tempo que o veículo deve aguardar para efetuar suas tarefas. Assim, a soma dos tempos de viagem e dos tempos de serviço das cidades já visitadas por um veículo deve ser maior ou igual ao horário inicial, e menor que o horário final de atendimento da janela de tempo associada ao próximo cliente a ser visitado. Caso o veículo chegue mais cedo ao consumidor, ele pode esperar o início da janela de tempo no local. O tempo de serviço só pode começar dentro da janela de tempo e assim que o veículo chega. Associados com a sequência i de clientes para os clientes j há um custo c_{ij} , um tempo de viagem t_{ij} e uma distância d_{ij} . Assume-se também que c_i , t_{ij} e d_{ij} são medidas equivalentes, com ajustamentos adequados. Um custo w_k (custo para a contratação de um veículo) é usado para a ativação do veículo $k \in K$. Todas as rotas devem satisfazer às restrições das janelas de tempo que definem que o veículo não pode atender um cliente i antes ou depois dos limites de tempo inferior e superior, respectivamente.

2. METODOLOGIA

2.1. REPRESENTAÇÃO DE UMA SOLUÇÃO

Uma solução para o PRAVJT é representada através de um vetor solução que armazena uma permutação de cidades, numeradas de 1 a n , que são separadas de acordo com o número de rotas criadas. O valor zero é utilizado como elemento separador, que indica o depósito. Cada rota percorrida por um veículo é chamada de pétala. Como exemplo, considere uma região onde existem 9 cidades, que podem ser atendidas por 3 caminhões. O vetor solução para este caso seria $S = [0, 2, 5, 7, 0, 3, 6, 4, 0, 8, 1, 9]$, em que $[0, 2, 5, 7]$, $[0, 3, 6, 4]$ e $[0, 8, 1, 9]$ são as rotas (pétalas) desta solução.

2.2. ESTRUTURA DE VIZINHANÇA

O espaço de busca consiste no conjunto de soluções $s \in S$ que pertencem ao PRAVJT, em que se procura obter novas soluções por meio de movimentos de troca e realocação para a geração de vizinhos de s . Neste trabalho foram utilizadas dez diferentes estruturas de vizinhança para explorar o espaço de busca. Com isso, são utilizados movimentos que consistem na modificação de apenas uma rota (intra-rotas) e movimentos que alteram duas rotas (inter-rotas). Os movimentos do tipo intra-rota são o Shift'(1), Shift'(2), Shift'(3) e Exchange. Os movimentos do tipo inter-rota são Shift (1,0), Shift (2,0), Shift (3,0), Swap (1,1), Swap (2,1) e Swap (2,2).

Os movimentos do tipo Shift'(k) consistem na transferência de k clientes adjacentes para outra posição da rota à qual pertencem. O movimento Exchange consiste na permutação entre dois clientes de uma mesma rota. Os movimentos do tipo Shift (k,0) são semelhantes aos do tipo Shift'(k), porém eles realizam a alteração de duas rotas por meio da realocação de k

clientes adjacentes de uma rota para outra. Os movimentos do tipo Swap (k,l) também alteram duas rotas quando realizados, permutando k clientes adjacentes de uma rota com l clientes adjacentes de outra.

Além dos movimentos descritos anteriormente, foram também utilizados outros dois movimentos que são o ER (Elimina Rota) e o ERFO (Elimina Rota Função Objetivo). O movimento ER (Elimina Rota) consiste na tentativa de eliminar uma rota da solução, em que primeiramente se escolhe a menor rota para tentar eliminá-la por meio da retirada dos clientes pertencentes a ela. Com isso, se busca inserir os clientes removidos da rota escolhida inserindo-os nas demais rotas da solução, dando-se preferência para as maiores rotas. Se todos os clientes retirados da rota escolhida forem inseridos nas demais rotas o procedimento termina. Quando não é possível inserir um dos clientes removidos da rota escolhida em nenhuma das outras rotas, tal rota permanece inalterada e o processo é aplicado para as demais rotas até que seja minimizado o número de rotas ou até que todas as rotas também sejam analisadas para a sua eliminação. O movimento ERFO (Elimina Rota Função Objetivo) é semelhante ao ER, porém, o procedimento termina quando todos os clientes da rota escolhida são retirados e inseridos nas demais rotas ou o valor da função objetivo é minimizado. O ERFO também termina quando todas as rotas a partir da menor para a maior são analisadas para serem eliminadas.

2.3. FUNÇÃO DE AVALIAÇÃO

Uma solução $s \in S$ gerada pelo algoritmo é avaliada pela função representada pela equação 1, que calcula o custo de deslocamento dos veículos, ou seja, o total das distâncias percorridas por todos os veículos em suas respectivas rotas.

$$f(s) = \sum_{(i,j) \in A} d_{ij} x_{ij} \quad (1)$$

Sendo A o conjunto das arestas (i; j). O custo de deslocamento de um cliente i para o cliente j é definido por d_{ij} . A variável x_{ij} serve para indicar se o arco (i; j) faz parte da solução, em que tal variável recebe o valor 1 caso o arco esteja na solução e 0, caso contrário.

2.4. PUSH FORWARD INSERTION HEURISTIC

A heurística PFIH (*Push Forward Insertion Heuristic*) foi proposta por Solomon (1987) para a construção de soluções para o PRVJT. O PFIH consiste em um algoritmo de construção que realiza a inserção das cidades nas rotas de acordo com o seu custo de inserção. O custo de inserção das cidades é calculado de acordo com a sua distância geográfica, limitante superior da janela de tempo e o ângulo polar entre a cidade e o depósito. Assim, quando uma cidade é designada para ser inserida na solução o seu custo de inserção é verificado em todas as possíveis posições das rotas pertencentes à solução corrente. Quando é analisado o custo de inserção de uma cidade, é também verificado se o atendimento da cidade em determinada posição não viola as restrições das janelas de tempo e de carga dos consumidores. Com isso, após a análise da inserção da cidade em todas as posições da solução, é escolhida a posição em que o atendimento da cidade apresentar o menor custo, ou seja, a menor distância total percorrida. Além disso, a sequência em que os consumidores são escolhidos para serem inseridos na solução é determinada através da expressão (2):

$$C_i = -\alpha \cdot d_{oi} + \beta \cdot b_i + \gamma \cdot \left[\left(\frac{p_i}{360} \right) \cdot d_{oi} \right], \forall i \in C \quad (2)$$

Na expressão (2) os parâmetros α , β e γ foram definidas por Solomon (1987) de

forma empírica com os valores 0.7, 0.1 e 0.2, respectivamente. O parâmetro d_{0i} é a distância entre o depósito e o consumidor i , b_i é o limitante superior da janela de tempo do consumidor i e p_i é o ângulo polar do consumidor i em relação ao depósito. A sequência definida para o atendimento dos consumidores na solução do PRAVJT é um fator importante para a geração de soluções de qualidade por meio do PFIH. Com isso, Solomon (1987) desenvolveu tal função para determinar a ordem de inserção dos consumidores na solução com base nas características do problema como a distância entre os clientes e suas janelas de tempo.

2.5. HEURÍSTICA SORTEIO DA COPA DO MUNDO

Neste artigo é proposto o uso de um novo algoritmo para a construção de soluções para o PRAVJT. Por utilizar uma estratégia de escolha dos clientes a serem atendidos baseada na forma de sorteio que designa quais times de futebol irão jogar em determinados grupos de um campeonato, como é o caso do campeonato mundial de futebol entre seleções, por exemplo, o algoritmo desenvolvido é chamado de Heurística Sorteio da Copa do Mundo (HSCM). O HSCM funciona de modo que sejam consideradas as características do PRAVJT como as restrições de carga e de tempo do problema, por exemplo. Considerando o fato de que, neste problema, a frota de veículos é homogênea, a quantidade R mínima de rotas que a solução irá possuir é definida de acordo o resultado da expressão (3), em que é feita divisão da demanda total dos clientes pela capacidade de carga dos veículos. Nesta expressão, q_i é a demanda de cada cliente i e Q a capacidade dos veículos. Além disso, os clientes são divididos em grupos de sorteio para serem atendidos, sendo esta divisão feita de acordo com as suas respectivas localizações geográficas e janelas de tempo, por meio da expressão (2) que é utilizada pelo PFIH. A quantidade de grupos de sorteio é estabelecida pela razão entre o número de cidades e de rotas inicial R . Com isso, são criados $(|N| / R)$ grupos que armazenam até R clientes. A ordem de inserção dos clientes em cada um dos grupos de sorteio criados é feita conforme a expressão (2), em que cada cliente i é designado para fazer parte de um grupo j . Com isso, os grupos de sorteio com menor índice j , tendem a receber os clientes que conforme a expressão (2) são os mais adequados para serem atendidos primeiro por um dos veículos da frota. Após a divisão dos clientes em grupos, é feito o sorteio de cada um dos clientes a serem inseridos na solução. O sorteio dos clientes é realizado de maneira que, ele se inicia no grupo de sorteio 1 e termina no grupo n . Desta forma, enquanto todos os clientes de um grupo j ainda não foram atendidos, é feita a escolha aleatória de um dos clientes i pertencente a tal grupo para ser inserido na solução. Ao ser sorteado para ser inserido na solução, o custo de inserção do cliente escolhido é analisado em todas as posições das rotas existentes. Durante a análise do custo de inserção, também é verificado se nenhuma das restrições de capacidade dos veículos e das janelas de tempo dos clientes são violadas. Assim, após a análise de tal custo de inserção, é escolhida a posição da solução que apresentar o menor custo de inserção, ou seja, tiver menor impacto no valor da distância total percorrida pelos veículos. Quando todos os clientes de um grupo j são atendidos, o processo de sorteio para o atendimento dos clientes é feito sobre o próximo grupo de sorteio e tal processo é repetido até que todos os clientes dos grupos de sorteio sejam atendidos.

$$R = \frac{\sum_{i=1}^{|N|} q_i}{Q}, \forall i \in N \quad (3)$$

A definição do número mínimo de rotas necessário, a divisão dos clientes em grupos de sorteio para o seu atendimento e o procedimento de inserção do cliente na posição de menor custo, auxiliam a designar os clientes a serem atendidos de forma mais uniforme.

```

1 Procedimento HSCM
2  $f(s^*) \leftarrow \infty$ ;
3 Defina o valor de  $R$  de acordo com a expressão (3);
4 Crie  $(N/R)$  grupos de sorteio;
5 Calcule para todo o cliente  $i$ , o custo PFIH conforme a expressão (2);
6 Ordenar em ordem crescente todos os clientes segundo o seu custo PFIH;
7  $i \leftarrow 0$ ,  $iter \leftarrow 0$ ;
8 Para (cada Grupo  $j = 1, \dots, (N/R)$ ) faça
9      $k \leftarrow 0$ ;
10    enquanto ( $k < R \ \&\& \ i < N$ ) faça
11         $Grupo(j) \leftarrow Grupo(j) \cup \{Cliente(i)\}$ ;
12         $k \leftarrow k+1$ ;  $i \leftarrow i+1$ ;
13    fim-enquanto;
14 fim-para;
15 enquanto ( $iter < iter_{max}$ ) faça
16     Inicialize uma solução  $s$  com  $R$  rotas;
17      $s' \leftarrow Construação(s, Grupo, N, R)$ ;
18     se ( $f(s') < f(s^*)$ ) então
19          $s^* \leftarrow s'$ ;
20          $f(s^*) \leftarrow f(s')$ ;
21     fim-se;
22      $iter \leftarrow iter + 1$ ;
23 fim-enquanto;
24 retorne  $s^*$ ;
25 fim HSCM;

```

Figura 1: Pseudocódigo do algoritmo HSCM.

```

1 Procedimento Construação( $s, Grupo, N, R$ );
2  $ImpactoInsercao_{best} \leftarrow \infty$ ,  $Posicao_{best} \leftarrow 0$ ;
3 Para (cada Grupo  $t = 1, \dots, (N/R)$ ) faça
4     enquanto (houverem clientes no Grupo( $t$ )) faça
5         Selecione, aleatoriamente, um cliente  $c$  do Grupo( $t$ );
6         enquanto (todas as rotas  $\in s$  não forem percorridas) faça
7             se (inserção de  $c$  entre os clientes  $i$  e  $j$  respeita todas as restrições) então
8                 se ( $f(s \cup \{c\}) < ImpactoInsercao_{best}$ ) então
9                      $ImpactoInsercao_{best} \leftarrow f(s \cup \{c\})$ ;
10                     $Posicao_{best} \leftarrow (i, j)$ ;
11                fim;
12            fim;
13        fim-enquanto;
14        se (não for possível a inserção  $c$  em nenhuma rota) então
15            Crie uma nova rota e insira o cliente  $c$  nela;
16             $f(s) \leftarrow f(s \cup \{c\})$ ;
17        senão
18            Insira o cliente  $c$  na posição de menor custo ( $Posicao_{best}$ );
19             $f(s) \leftarrow ImpactoInsercao_{best}$ ;
20        fim;
21    fim-enquanto;
22 fim-para;
23 retorne  $s$ ;
24 fim Construção;

```

Figura 2: Pseudocódigo do procedimento de construção do algoritmo HSCM.

2.6. ITERATED LOCAL SEARCH

Segundo Lourenço *et al.* (2003) o algoritmo de Busca Local Iterativa (ILS), trabalha de modo que o processo de busca local pode ser melhorado, gerando novas soluções de partida, de forma que tais soluções são obtidas por meio de perturbações na solução ótima local. Neste trabalho, o ILS foi utilizado para refinar as soluções obtidas por meio do HSCM e do PFIH, ou seja, eles foram utilizados como componentes de geração de soluções iniciais para o problema. O processo de perturbação do ILS consiste na aplicação de diferentes níveis de perturbação na solução corrente. Caso não seja obtida uma solução melhor, o nível de perturbação sobre tal solução é aumentado até um determinado limite. No algoritmo ILS implementado foram utilizados os movimentos do tipo Shift'(k), Shift(k,0), ER e ERFO. A perturbações realizadas nesta versão do ILS consistem na aplicação de 2 a 9 movimentos do tipo Shift'(k) ou Shift(k,0). No caso das perturbações feitas por meio dos movimentos ER ou ERFO, tais movimentos são aplicados apenas uma vez. Após a perturbação é aplicado o processo de busca local por meio do método VND (*Variable Neighborhood Descent*) proposto por Mladenovic e Hansen (1997), em que são utilizados todos os movimentos descritos na seção (3.2), exceto o ER e o ERFO. A cada iteração do ILS, a solução obtida é avaliada por meio do procedimento de aceitação para definir de qual será a solução no próximo procedimento de perturbação.

3. RESULTADOS COMPUTACIONAIS

Os algoritmos propostos foram desenvolvidos em linguagem C++, usando o compilador gnu GCC. O algoritmo foi executado em um computador com processador Pentium Intel(R) Core(TM)2 Quad Q8400, com clock de 2.66 GHz, 3.7 GB de RAM, sob a plataforma Windows Seven. Para testar os algoritmos desenvolvidos, foram utilizadas as instâncias propostas por Solomon (1987). Estas instâncias se dividem nos grupos C, R e RC. No conjunto de instâncias C, os clientes são localizados próximos uns dos outros, formando grupos. No conjunto de instâncias R, os clientes estão localizados em posições aleatórias sem agrupamento. No caso do grupo de instâncias RC, há tanto clientes localizados próximos uns dos outros como no conjunto C quanto em posições distantes como no conjunto R. Foram utilizados os conjuntos de instâncias do tipo C1, R1, RC1, C2, R2 e RC2. Os conjuntos de instâncias utilizadas para a realização deste trabalho possuem 100 clientes. A capacidade Q dos veículos nos conjuntos de instâncias usadas é igual a 200 unidades. O algoritmo proposto foi executado trinta vezes para cada uma das instâncias utilizadas. Os parâmetros utilizados para a execução do algoritmo, como o número de iterações do HSCM e os níveis de perturbação utilizados durante a execução do ILS foram definidos de forma empírica, em que foram feitos testes para definir os mesmos.

Nas tabelas 1 e 2 as colunas que vão de 2 a 7 apresentam os dados referentes à distância percorrida e ao número de veículos das rotas construídas pela heurística desenvolvida por Repoussis *et al.* (2009), pelo HSCM+ILS e PFIH+ILS. Nas tabelas 3 e 4 são mostrados os demais dados referentes aos resultados obtidos pelos algoritmos HSCM+ILS e PFIH+ILS, ao serem aplicados sobre as instâncias utilizadas, onde a primeira coluna indica o nome da instância. As colunas de 2 a 4 se referem à média (MDT), desvio padrão (DP), o desvio da melhor solução (DMS) e desvio do resultado médio (DRM) acerca dos resultados obtidos pelo HSCM+ILS. As colunas de 6 a 9 contém os dados acerca da média, desvio padrão, o desvio da melhor solução e desvio do resultado médio em relação aos resultados encontrados pelo PFIH+ILS. O DMS e o DRM são calculados de acordo com as equações 3 e 4, respectivamente. Nas equações abaixo, fo^* refere-se ao melhor resultado encontrado por

Repoussis *et al.* (2009) e foi ao resultado encontrado na i -ésima execução dos algoritmos.

$$DMS = \frac{fo^* - \min_{i=1, \dots, 30} \{fo_i\}}{fo^*} \quad (4)$$

$$DMS = \frac{fo^* - \frac{1}{30} \left(\sum_{i=1}^{30} fo_i \right)}{fo^*} \quad (5)$$

Tabela 1: Resultados para os conjuntos de instâncias R1, C1 e RC1

Conjunto	Repoussis (2009)		HSCM+ILS		PFIH+ILS	
	DT	NV	DT	NV	DT	NV
R101	1192,85	19	1195,13	18	1195,13	18
R102	1079,39	17	1046,79	17	1046,4	17
R103	1016,78	13	982,42	13	986,16	13
R104	869,63	9	809,11	9	801	9
R105	1055,04	14	1090,62	13	1082,37	13
R106	1000,95	12	979,64	12	988,3	12
R107	912,99	10	946,06	10	925,5	10
R108	760,3	9	780,31	9	787,6	9
R109	934,53	11	912,18	12	981,45	11
R110	846,49	10	898,12	10	883,08	10
R111	895,21	10	872,96	10	906,85	10
R112	811,73	9	826,96	9	753,07	10
C101	556,18	10	556,03	10	556,03	10
C102	556,18	10	556,03	10	556,03	10
C103	556,18	10	556,03	10	556,03	10
C104	555,41	10	555,27	10	555,27	10
C105	556,18	10	556,03	10	556,03	10
C106	556,18	10	556,03	10	556,03	10
C107	556,18	10	556,03	10	556,03	10
C108	555,8	10	555,65	10	555,65	10
C109	555,8	10	555,65	10	555,65	10
RC101	1227,37	14	1130,06	14	1131,64	14
RC102	1203,05	12	1071,65	12	1043,87	13
RC103	923,15	11	931,97	11	920,89	11
RC104	787,02	10	806,19	10	801,52	10
RC105	1195,2	13	1047,62	13	1052,92	13
RC106	1095,65	11	997,22	11	1004,88	11
RC107	861,28	11	841,77	11	844,95	11
RC108	831,09	10	841,27	10	804,79	10

Tabela 2: Resultados para os conjuntos de instâncias R2, C2 e RC2

Conjunto	Repoussis (2009)		HSCM+ILS		PFIH+ILS	
	DT	NV	DT	NV	DT	NV
R201	1182,43	4	1188,48	4	1214,68	4
R202	1149,59	3	1035,47	4	1038,07	4
R203	889,12	3	913,72	3	933,96	3
R204	801,46	2	731,48	3	839,52	2
R205	943,33	3	1000,41	3	995,76	3
R206	869,27	3	881,46	3	895,74	3
R207	857,08	2	803,33	3	795,54	3
R208	700,53	2	714,89	2	725,69	2
R209	851,69	3	874,78	3	855,15	3
R210	892,45	3	917,74	3	922,8	3
R211	886,9	2	773,91	3	771,89	3
C201	548,51	3	548,3	3	548,3	3
C202	548,51	3	548,3	3	548,3	3
C203	548,13	3	546,99	3	635,74	3
C204	547,55	3	549,26	3	651,1	3
C205	545,83	3	545,61	3	545,61	3
C206	545,45	3	545,22	3	545,22	3
C207	545,24	3	545,01	3	545,01	3
C208	545,28	3	545,05	3	545,05	3
RC201	1303,73	4	1343,77	4	1345,6	4
RC202	1321,43	3	1095,83	4	1112,3	4
RC203	993,29	3	1039,15	3	1027,37	3
RC204	718,97	3	816,05	3	796,82	3
RC205	1189,84	4	974,4	3	986,9	3
RC206	1091,79	3	1140,86	3	1118,31	3
RC207	998,7	3	1049,26	3	1030,02	3
RC208	769,4	3	845,48	3	835,93	3

Tabela 3: Resultados para R1, C1 e RC1

Conjunto	HSCM+ILS				PFIH+ILS			
	MDT	DP	DMS	DM	MDT	DP	DMS	DM
R101	1194,74	10,66	0,00	0,00	1198,85	5,23	0,00	-0,01
R102	1052,25	4,26	0,03	0,03	1050,55	3,93	0,03	0,03
R103	904,72	272,81	0,03	0,11	978,30	28,52	0,03	0,04
R104	815,15	28,00	0,07	0,06	821,05	33,87	0,08	0,06
R105	1053,70	15,20	-0,03	0,00	1077,52	39,58	-0,03	-0,02
R106	1003,99	9,50	0,02	0,00	1009,31	18,94	0,01	-0,01
R107	928,16	38,22	-0,04	-0,02	936,45	49,83	-0,01	-0,03
R108	768,48	21,11	-0,03	-0,01	763,13	18,24	-0,04	0,00
R109	934,77	18,84	0,02	0,00	931,18	20,56	-0,05	0,00

R110	872,95	24,29	-0,06	-0,03	863,41	28,78	-0,04	-0,02
R111	909,15	17,62	0,02	-0,02	899,67	49,32	-0,01	0,00
R112	775,22	19,75	-0,02	0,04	783,37	18,16	0,07	0,03
C101	556,03	0,00	0,00	0,00	558,33	8,75	0,00	0,00
C102	556,61	3,19	0,00	0,00	570,72	29,63	0,00	-0,03
C103	556,48	2,47	0,00	0,00	577,91	40,18	0,00	-0,04
C104	577,50	26,89	0,00	-0,04	619,60	65,70	0,00	-0,12
C105	556,03	0,00	0,00	0,00	556,03	0,00	0,00	0,00
C106	557,27	6,77	0,00	0,00	567,02	23,04	0,00	-0,02
C107	556,03	0,00	0,00	0,00	567,02	23,04	0,00	-0,02
C108	555,65	0,00	0,00	0,00	566,97	29,57	0,00	-0,02
C109	566,79	17,69	0,00	-0,02	584,33	51,34	0,00	-0,05
RC101	1148,19	19,68	0,08	0,06	1147,58	11,95	0,08	0,07
RC102	1070,61	38,84	0,11	0,11	1079,02	32,59	0,13	0,10
RC103	980,39	20,61	-0,01	-0,06	939,69	14,32	0,00	-0,02
RC104	856,27	30,46	-0,02	-0,09	826,68	27,59	-0,02	-0,05
RC105	1080,26	18,14	0,12	0,10	1067,51	19,83	0,12	0,11
RC106	1005,69	40,69	0,09	0,08	993,33	24,80	0,08	0,09
RC107	883,55	40,32	0,02	-0,03	894,11	34,51	0,02	-0,04
RC108	851,18	30,02	-0,01	-0,02	840,35	30,62	0,03	-0,01

Tabela 4: Resultados para R2, C2 e RC2

Conjunto	HSCM+ILS				PFIH+ILS			
	MDT	DP	DMS	DM	MDT	DP	DMS	DM
R201	1208,86	18,32	-0,01	-0,02	1264,28	34,37	-0,03	-0,07
R202	1070,83	23,13	0,10	0,07	1071,99	25,31	0,10	0,07
R203	947,12	29,54	-0,03	-0,07	925,47	156,57	-0,05	-0,04
R204	747,54	13,59	0,09	0,07	781,90	28,61	-0,05	0,02
R205	1062,73	35,88	-0,06	-0,13	1035,03	23,23	-0,06	-0,10
R206	935,11	32,68	-0,01	-0,08	934,98	21,87	-0,03	-0,08
R207	839,86	20,81	0,06	0,02	824,70	19,78	0,07	0,04
R208	726,83	9,08	-0,02	-0,04	762,63	23,76	-0,04	-0,09
R209	915,71	27,90	-0,03	-0,08	985,92	48,49	0,00	-0,16
R210	938,08	16,96	-0,03	-0,05	965,81	33,95	-0,03	-0,08
R211	799,15	18,23	0,13	0,10	809,56	14,16	0,13	0,09
C201	548,30	0,00	0,00	0,00	548,30	0,00	0,00	0,00
C202	548,30	0,00	0,00	0,00	550,27	10,81	0,00	0,00
C203	548,56	3,02	0,00	0,00	760,76	53,10	-0,16	-0,39
C204	553,91	2,48	0,00	-0,01	697,03	39,20	-0,19	-0,27
C205	545,61	0,00	0,00	0,00	545,61	0,00	0,00	0,00
C206	545,22	0,00	0,00	0,00	597,26	52,35	0,00	-0,09
C207	545,01	0,00	0,00	0,00	554,33	21,22	0,00	-0,02
C208	545,05	0,00	0,00	0,00	546,75	9,30	0,00	0,00
RC201	1148,19	19,68	-0,03	0,12	1378,06	19,37	-0,03	-0,06

RC202	1070,61	38,84	0,17	0,19	1134,79	20,22	0,16	0,14
RC203	1081,34	28,66	-0,05	-0,09	1079,60	27,38	-0,03	-0,09
RC204	863,98	29,47	-0,14	-0,20	852,84	26,86	-0,11	-0,19
RC205	1065,38	38,83	0,18	0,10	1044,67	22,67	0,17	0,12
RC206	1174,45	64,66	-0,04	-0,08	1173,08	40,75	-0,02	-0,07
RC207	1107,44	40,05	-0,05	-0,11	106,77	36,59	-0,03	0,89
RC208	875,38	17,01	-0,10	-0,14	890,13	27,76	-0,09	-0,16

Segundo os dados das tabelas acima, os algoritmos propostos demonstraram ser competitivos em comparação com os resultados apresentados por Repoussis *et al.* (2009). O algoritmo HSCM+ILS foi capaz de encontrar novas soluções para mais de 58,33% das instancias do conjunto R1, apesar de não ter encontrado soluções melhores do que as apresentadas na literatura para o conjunto R2, os resultados alcançados pelo algoritmo foram próximos. Para os conjuntos de instâncias C1 e C2, o HSCM+ILS foi capaz de encontrar novas soluções em 100% e 87,5% das instancias testadas, respectivamente. Ao ser aplicado sobre os problemas-teste das instancias RC1 e RC2, o HSCM+ILS apresentou novas soluções em 62,5% e 12,5% das instancias testadas, respectivamente.

Ao ser testado, utilizando o conjunto de instâncias R1, o algoritmo PFIH+ILS apresentou soluções melhores do que as da literatura em 50% das instâncias testadas. O algoritmo PFIH+ILS também foi aplicado sobre o conjunto R2, apesar de ter apresentado resultados próximos, o PFIH+ILS não foi capaz de encontrar soluções melhores do que as existentes na literatura neste caso. Quando foi usado para solucionar os conjuntos C1 e C2, o PFIH+ILS encontrou novas soluções em 100% e 75% das instâncias testadas, respectivamente. O algoritmo PFIH+ILS demonstrou também ser eficiente ao ser testado para solucionar as instâncias do conjunto RC1, encontrando soluções melhores do que as da literatura em 75% dos casos, já para o conjunto RC2, o ILS apresentou resultados melhores para apenas 12,5% dos casos.

Ao analisar os resultados obtidos pelo PFIH+ILS e o HSCM+ILS, percebe-se que em relação ao conjunto R1 o primeiro foi melhor do que o segundo em 50% das instâncias e empatando em 8,33% dos casos. Para o conjunto de instâncias R2 o HSCM+ILS obteve soluções melhores do o PFIH+ILS em 54,55% dos casos. No caso do conjunto de instâncias C1 houve empate entre os algoritmos em 100% das instâncias. Para o conjunto C2, o HSCM+ILS encontrou soluções melhores do o PFIH+ILS para 25% das instâncias, porém houve empate entre eles em 75% dos demais casos. Em relação ao conjunto RC1, o algoritmo HSCM+ILS apresentou melhores resultados do que o PFIH+ILS para 62,5% das instâncias, mas para o conjunto RC2, o PFIH+ILS obteve melhores soluções em 62,5% dos testes.

4. CONCLUSÕES

Este trabalho apresentou um estudo sobre o uso de heurísticas para solucionar o PRAVJT que é um problema NP - difícil, bem como a proposta de um novo algoritmo de construção de soluções para o mesmo, o HSCM. Foram aplicados sobre o PRAVJT os algoritmos HSCM+ILS e PFIH+ILS que consistem no uso das heurísticas construtivas HSCM e PFIH para gerar a solução inicial utilizada pelo ILS, respectivamente. Com isso, as soluções geradas pelas as heurísticas utilizadas são refinadas pelo ILS. O HSCM+ILS e o PFIH+ILS foram capazes de gerar soluções melhores do que as presentes na literatura para mais de 51,8% e 50% das instâncias. Ao se comparar os algoritmos desenvolvidos, percebe-se que o

HSCM+ILS demonstrou ser competitivo em relação ao PFIH+ILS, conseguindo obter soluções melhores do que o segundo algoritmo em grande parte das instâncias dos conjuntos R1, R2 e RC1, por exemplo. Espera-se que este estudo possa contribuir para a realização de trabalhos que tratem do PRAVJT e de novos estudos que visem o aperfeiçoamento da heurística proposta, como o uso de um método de busca local.

5. AGRADECIMENTOS

Os autores agradecem pelo apoio oferecido pelo CEFET-MG, CAPES e FAPEMIG.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BRANDÃO, J. A tabu search algorithm for the open vehicle routing problem. *European Journal of Operation Research*, v. 157, p. 552–564, 2004.
- [2] GUIYUN, LI. An improved ant colony algorithm for open vehicle routing problem with time windows. *International Conference on Information Management, Innovation Management and Industrial Engineering*, v. 2, p. 616–619, 2009.
- [3] LETCHFORD, A.N.; LYSGAARD, J. E EGLESE, R.W. A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society*, v. 58, p. 1642–1651, 2006.
- [4] LI, F.; GOLDEN, B. EWASIL, E. The open vehicle routing problem: Algorithms, large scale test problems, and computational results. *Computers and Operations Research*, v.34, p. 2918–2930, 2007.
- [5] LI, X.Y.; TIAN, P. E LEUNG, S.C.H. An ant colony optimization metaheuristic hybridized with tabu search for open vehicle routing problems. *Journal of the Operational Research Society*, v. 60, p. 1012–1025, 2009.
- [6] LOURENÇO, H.R.; MARTIN, O.C. E STÜTZLE, THOMAS. Iterated local search. Glover, G.F. e Kochenberger, editor, *Handbook of Metaheuristics*, p. 321–353. Kluwer Academic Publishers, Boston, 2003.
- [7] MIRHASSANI, S.A. E ABOLGHASEMI, N. A particle swarm optimization algorithm for open vehicle routing problem. *Expert Systems with Applications*, v. 38, p. 11547–11551, 2011.
- [8] MLADENOVIC, N. E HANSEN, P. Variable neighborhood search. *Computers and Operations Research*, v. 24, p. 1097–1100, 1997.
- [9] REPOUSSIS, P.P.; TARANTILIS, C.D. E IOANNOU, G. The open vehicle routing problem with time windows. *Journal of the Operational Research Society*, v. 58, p. 1–13, 2006.
- [10] REPOUSSIS, P.P.; TARANTILIS, C.D. E IOANNOU, G. An evolutionary algorithm for the open vehicle routing problem with time windows. *Bio-inspired algorithms for the vehicle routing problems*, v. 161, p. 55–75, 2009.
- [11] SCHRAGE, L. Formulation and structure of more complex/realistic routing and scheduling problems. *Networks*, v. 11, p. 229–232, 1981.
- [12] SOLOMON, M.M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, v. 35, p. 2544–265, 1987.