ESTUDO DA APLICAÇÃO DE METAHEURÍSTICAS AO PROBLEMA DE ROTEAMENTO ABERTO DE VEÍCULOS COM JANELAS DE TEMPO

José Maurício Costa[1], Sérgio Ricardo de Souza[1], Marcone Jamilson Freitas Souza[2]

[1]Programa de Mestrado em Modelagem Matemática e Computacional
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas, 7675, CEP 30510-000, Belo Horizonte (MG), Brasil

jmcosta25@gmail.com, sergio@dppg.cefetmg.br

[2]Departamento de Computação – Universidade Federal de Ouro Preto (UFOP)
Campus Universitário, Morro do Cruzeiro, CEP 35.400-000, Ouro Preto (MG), Brasil

marcone@iceb.ufop.br

Resumo. *Este artigo apresenta a aplicação de metaheurísticas para solucionar 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. O PRAVJT é uma variação do Problema de Roteamento de Veículos com Janelas de Tempo, em que o veículo não precisa retornar para o depósito após atender o último cliente. Para a solução, foram utilizados dois algoritmos baseados nas metaheurísticas MMAS, ILS e PFIH. Essas abordagens foram testadas por meio de experimentos computacionais, em que foram utilizadas instâncias encontrada na literatura. Os algoritmos mostraram ser competitivos, conseguindo gerar soluções melhores do que as presentes na literatura para a maioria dos testes.*

PALAVRAS-CHAVE: *Problema de Roteamento Aberto de Veículos com Janelas de Tempo, Metaheurísticas, Otimização*

Abstract. *This article presents the application of metaheuristics for solving the Open Vehicle Routing Problem with Time Windows (OVRPTW). This problem is NP-hard, consisting in defining the minimum number of vehicles to serve a set of customers, so that the total distance traveled by them is also minimized. The OVRPTW is a variant of the Vehicle Routing Problem with Time Windows, in which the vehicle does not need to return for the deposit after attending the last client. Two algorithms based on the metaheuristics MMAS, ILS and PFIH are developed for solving instances of the problem. Computational experiments with instances from literature are performed for testing these approaches. The algorithms shown to be competitive, obtaining better solutions than those found in the literature for most of the tests.*

KEYWORDS: *Open Vehicle Routing Problem with Time Windows, Metaheuristics, Optimization*

1 Introdução

Para Li et al. (2007), atualmente, devido aos custos de transporte de bens, várias empresas vêm contratando serviços de transporte terceirizados para realizarem a entrega de seus produtos. Com isso, é possível obter uma significante redução sobre os gastos da empresa para com os serviços de entrega de mercadorias para os seus clientes. Desta forma, não há uma preocupação por parte da empresa se os veículos irão retornar para o ponto de partida. Situações como essa caracterizam o Problema de Roteamento Aberto de Veículos (PRAV), que é uma variante do Problema de Roteamento de Veículos (PRV) Clássico. No PRAV o veículo não retorna para o depósito após visitar o último cliente da rota atendida por ele. Além disso, diferente do PRV em que as rotas representam ciclos Hamiltonianos, no PRAV essas rotas representam caminhos Hamiltonianos. Brandão (2004) diz que, o PRAV é um problema de otimização combinatória que consistem em encontrar as melhores rotas para uma frota de veículos, que, por sua vez, devem atender um conjunto de cidades com demanda e localização geográfica conhecidas. Devido às suas características que o tornam mais adequado descrever problemas do mundo real, o PRAV possui grande importância prática na área de logística. Apesar disso, conforme Li et al. (2009), o PRAV não tem recebido a mesma atenção que o PRV, havendo poucos trabalhos na literatura que tratam deste problema em específico. De acordo com Brandão (2004), o PRV é um problema de otimização NP−difícil, deste modo, o uso de heurísticas e metaheurísticas para a sua solução se justificam. 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 que são 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 dois métodos para a geração da solução inicial que são a heurística do vizinho mais próximo (*Nearest Neighbour Heuristic*) e o procedimento US (*Unstringing and Stringing*). 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 com "olhar à frente" 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) pesquisou o PRAVJT, em que foi desenvolvido um algoritmo baseado na metaheurística Colônia de Formigas para solucioná-lo. Este algoritmo foi desenvolvimento de forma que o processo de construção das soluções é feito pelas formigas de forma paralela, ou seja, várias soluções são criadas ao mesmo tempo, em que as formigas também interagem entre si, de forma que, elas utilizam informações referentes às trilhas de feromônio das soluções que são construídas por elas. Repoussis et al. (2009) propuseram um algoritmo evolucionário para resolver o PRAVJT. Nele, a cada iteração, uma nova população é gerada por meio do processo de mutação, que, por sua vez, é baseado nas arestas extraídas de pais individuais e as soluções geradas são melhoradas através do algoritmo Busca Tabu.

Neste estudo, foram utilizados os algoritmos MMAS+ILS e ILS para tentar solucionar o PRAVJT, em que o primeiro consiste na combinação entre as metaheurísticas *MAX−MIN Ant System* e ILS. O artigo está organizado da seguinte forma: a seção 2 apre-

senta a descrição do problema. A seção 3 mostra a metodologia utilizada neste trabalho. Os resultados dos experimentos computacionais são discutidos na seção 4. As conclusões sobre o estudo estão na seção 5.

2 Problema de Roteamento Aberto de Veículos com Janelas de Tempo

Para Li et al. (2009), o PRAVJT é considerado uma relaxação do PRAVJT clássico. Esse problema 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 $V = \{0, 1, \cdots, N\}$, 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 $A = \{(i, j) |\, i, j \in V, i \neq j\}$. Cada aresta (i, j) está associada a 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 $C = \{1, 2, \cdots, N\}$, 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 PRAV é um problema de otimização combinatória que consiste na minimização do número de veículos que são necessários para atender os 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_{ij}, 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 de capacidade e de tempo, e as restrições da janela de tempo definem que o veículo não pode atender um cliente i antes ou depois dos limites de tempo inferior e superior, respectivamente.

3 Metodologia Proposta

3.1 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.

3.2 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 se 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 a maior rota seja também analisada 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.

3.3 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.

$$f(s) = \sum_{(ij)\in A} d_{ij}x_{ij} \tag{1}$$

Sendo A é 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.

3.4 *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 um cidade é designada para ser inserida na solução o seu custo de inserção é verificado em todas as possíveis posições das rotas pertencentes a 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):

$$c_i = -\alpha.d_{0i} + \beta.b_i + \gamma.\left[\left(\frac{p_i}{360}\right).d_{0i}\right] \ \forall i \in C \tag{2}$$

Na função (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.

3.5 *MAX-MIN Ant System*

Proposta por Dorigo et al. (1996), a metaheurística Colônia de Formigas é um método baseado no comportamento de formigas para realizar a busca por alimento. Nele, um conjunto formigas artificiais constroem soluções para um problema de otimização, onde elas cooperam entre si trocando informações sobre a qualidade das soluções construídas pelas mesmas. A troca de informações é feita por meio do feromônio depositado por cada uma das formigas ao se moverem dentro do espaço de busca, onde a trilha deixada por elas é usada como informação pelas demais formigas. De acordo com Dorigo e Stützle (2003), as formigas utilizadas no método são procedimentos de construção de soluções estocásticas, que criam soluções por meio de critérios de escolha probabilísticos durante as iterações do processo. A cada iteração são adicionadas novas informações para as soluções parciais com base nas características da instância do problema a ser resolvido e nas trilhas

de feromônio que se alteram dinamicamente conforme a qualidade das soluções obtidas pelas formigas. Durante o processo de construção de soluções, tais formigas percorrem o espaço de busca do problema por meio da construção de caminhos sobre o mesmo. Os movimentos produzidos pelas formigas são realizados com base em uma política de decisão local estocástica que se utiliza das trilhas de feromônio e informações heurísticas. Durante a etapa de construção, a formiga avalia a solução parcial ou completa e deposita o feromônio sobre os componentes ou conexões que ela usou. Esta informação de feromônio será usada para direcionar as próximas buscas a serem feitas pelas formigas. Esta escolha é feita de acordo com a seguinte probabilidade:

$$p_{ij}^{k} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l \in N_i^k}(\tau_{il})^{\alpha}(\eta_{il})^{\beta}}, se\ j \in N_i^k, \tag{3}$$

Na equação 3, N_i^k representa o conjunto de cidades ainda não visitadas pela formiga k. Os parâmetros α e β servem para determinar a influência relativa da trilha de feromônio e da informação heurística, respectivamente. O parâmetro $N_{ij} = \frac{1}{d_{ij}}$ é uma informação heurística disponível a priori.

A metaheurística Colônia de Formigas possui outros dois procedimentos que são o de evaporação das trilhas de feromônio e as ações *daemon*. O processo de evaporação de feromônio consiste na sua diminuição ao longo do tempo. Tal processo de evaporação serve para evitar uma convergência muito rápida do algoritmo para regiões sub ótimas. Deste modo, é implementada uma forma útil de esquecimento que favorece a exploração de novas regiões do espaço de busca. As ações *daemon* são usadas para implementar ações centralizadas, que não podem ser realizadas pelas formigas apenas. A construção de uma solução termina após todas as formigas criarem os seus caminhos. Em seguida, as trilhas de feromônio são atualizadas, onde é feita a evaporação das mesmas, para que depois cada formiga possa realizar o depósito de feromônio sobre as arestas pertencentes ao seu caminho.

A metaheurística *MAX-MIN Ant System* é uma das variações do algoritmo *Ant System*, e foi desenvolvida por Stützle e Hoos (1996). Para Stützle e Hoos (2000), por meio de uma forte exploração das melhores soluções encontradas durante o processo de pesquisa e da análise do espaço de busca, é possível se obter melhores resultados, porém, a realização de uma pesquisa de forma gulosa faz com que ocorra uma convergência rápida do algoritmo. Com isso, o *MAX-MIN Ant System* veio da idéia de realizar um maior aproveitamento das melhores soluções encontradas no processo de busca e ao mesmo tempo utilizar um mecanismo para se evitar a estagnação precoce do processo de pesquisa. Segundo Stützle e Hoos (2000) a atualização das trilhas de feromônio no *MAX-MIN Ant System* é feita apenas pela formiga que encontrou a melhor solução da iteração (s^{ib}) ou a melhor solução desde o início da execução do algoritmo (s^{gb}) ao final de uma iteração. A equação 4 abaixo caracteriza a forma como é feita a atualização das trilhas de feromônio.

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}^{best} \tag{4}$$

Na equação 4, $0 < \rho < 1$ é a taxa de evaporação da trilha de feromônio, $\Delta\tau_{ij}^{best}$ é

definido por $\Delta\tau_{ij}^{best} = 1/f\left(s^{best}\right)$ e $f\left(s^{best}\right)$ é o valor referente ao custo de s^{ib} ou s^{gb}. Deve ser feito tanto o uso de s^{ib} quanto o de s^{gb}, ao invés de apenas uma delas para torna a exploração do espaço de busca mais diversificada, evitando uma convergência precoce de tal exploração, já que os elementos que compõem s^{ib} também terão as suas respectivas trilhas de feromônio atualizadas. O emprego de limitantes para controlar a concentração de feromônio e impedir a rápida estagnação do procedimento de pesquisa é outra importante característica do *MAX-MIN Ant System*. Ao ser feito o procedimento de atualização das trilhas de feromônio são usados dois limitantes, um inferior (τ_{min}) e outro superior (τ_{max}) para regular a quantidade de feromônio que as trilhas possuem. Ao final de cada iteração deve ser verificado se a concentração de feromônio nas trilhas respeita os limites τ_{min} e τ_{max} definidos. Tal verificação é feita de acordo com a seguinte a regra:

$$\tau_{ij}(t) = \begin{cases} \tau_{min} & \text{Se } \tau_{ij}(t) < \tau_{min} \\ \tau_{max} & \text{Se } \tau_{ij}(t) > \tau_{max} \end{cases} \tag{5}$$

Conforme Dorigo et al. (2006), os valores de τ_{min} e τ_{max} podem ser definidos tanto de forma empírica ou analítica. No *MAX-MIN Ant System*, normalmente, a quantidade de feromônio inicial (τ_0) das trilhas é igual a τ_{max}. Esta medida é adotada, pois assim é possível se ter uma melhor exploração do espaço de busca no início do algoritmo.

3.6 *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 pertubações na solução ótima local. Neste trabalho, o ILS foi utilizado para refinar as soluções obtidas por meio do *MAX−MIN Ant System* e do PFIH, ou seja, eles foram utilizados como componentes de geração de soluções iniciais para o problema. O processo de pertubação do ILS consiste na aplicação de diferentes níveis de pertubação na solução corrente. Caso não seja obtida uma solução melhor, o nível de pertubação 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 pertubações realizadas nesta versão do ILS consistem na aplicação de 2 a 9 movimentos do tipo Shift'(k) ou Shift(k,0). No caso das pertubações feitas por meio dos movimentos ER ou ERFO, tais movimentos são aplicados apenas uma vez. Após a pertubação 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 pertubação.

3.7 *Algoritmos desenvolvidos*

Neste estudo foram desenvolvidos os algoritmos MMAS+ILS e ILS para solucionar o PRAVJT, em que o primeiro consiste na combinação entre as metaheurísticas MMAS e ILS e o segundo foi baseado na metaheurística ILS. No algoritmo MMAS+ILS, a metaheurística MMAS foi responsável pela geração da solução inicial e o ILS pelo seu refinamento. No algoritmo ILS, a solução inicial foi gerada pelo algoritmo PFIH. No MMAS+ILS, o MMAS foi desenvolvido de maneira que, a forma como as formigas utilizadas constroem as suas soluções foi baseada na heurística construtiva PFIH . Assim,

de acordo com a ordem de atendimento das cidades definida pela função (2) do PFIH, a formiga analisa todas as possíveis posições onde a cidade pode ser atendida e ao invés de escolher de maneira gulosa como no PFIH, tal formiga escolhe a posição de inserção da cidade na solução corrente de acordo com a regra de decisão (3). Na expressão (3), τ_{ij} se refere a concentração de feromônios do arco (i, j) que liga o cliente a ser atendido com a cidade ou depósito anterior a ele, na posição em que é verificada a sua inserção. O valor de η_{ij} é o custo de inserção da cidade em determinada posição da solução. Abaixo são apresentados os pseudocódigos dos algoritmos MMAS e MMAS+ILS.

1 **início**
2 Inicialize os parâmetros α, β, ρ, τ_{min} e τ_{max};
3 $\tau_0 = \tau_{max}$; $f^* \leftarrow \infty$; $iter \leftarrow 0$;
4 **faça** $\tau_{ij} \leftarrow \tau_0$ para todo o arco (i, j);
5 **enquanto** (iter $< iter_{\max}$) **faça**
6 **Para** (cada formiga $k = 1, ..., m$) **faça**
7 **Repita**
8 Selecione o próximo cliente para a formiga k;
9 **Até** (todos clientes serem atendidos);
10 **fim-para**;
11 $s^{ib} =$argmin$(f(s^1), f(s^2), ..., f(s^m))$;
12 **se** $(f(s^{ib}) < f(s^{gb}))$ **então**
13 $s^{gb} \leftarrow s^{ib}$;
14 $f(s^{gb}) \leftarrow f(s^{ib})$;
15 **fim-se**;
16 defina s^{best} igual a s^{ib} ou s^{gb}
17 **se** (arco(i, j) pertence à rota s^{best})
18 **então** $\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}^{best}$
19 **fim-se**;
20 **faça**
21 **if** $(\tau_{ij}(t) < \tau_{min})$ $\tau_{ij} = \tau_{min}$
22 **if** $(\tau_{ij}(t) > \tau_{max})$ $\tau_{ij} = \tau_{max}$
23 para todo arco (i, j) pertencente a s^{best};
24 $iter \leftarrow iter + 1$;
25 **fim-enquanto**;
26 **fim**.

(a) *MAX-MIN Ant System.*

1 **início**
2 $s_0 \leftarrow$ MaxMinAntSystem();
3 $s \leftarrow$ VND(s_0);
4 iter $\leftarrow$ 0;
5 **enquanto** (iter $< iter_{\max}$) **faça**
6 iter $\leftarrow$ iter +1;
7 $s' \leftarrow$ Perturbação(s, histórico);
8 $s'' \leftarrow$ VND(s');
9 $s \leftarrow$ CritérioAceitação(s, s', s'');
10 **fim-enquanto**;
11 **retorne** s;
12 **fim**.

(b) MMAS+ILS.

4 Resultados

Os algoritmos propostos foram desenvolvidos em linguagem C++, por meio do 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 Ultimate. Para testar os algoritmos MMAS+ILS e ILS, foram utilizadas as instâncias proposta 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 valor de α, β, ρ iguais a 1, 5, 0.5 respectivamente para a execução do *MAX−MIN Ant System* e os níveis de perturbação utilizados durante a execução do ILS foram definidos de forma empírica, em que foram feitos testes para o estabelecimento dos mesmos.

Para fins de comparação, as colunas 2, 3, 4, 5, 10 e 11 das tabelas 1 e 2 apresentam os dados referentes à distância percorrida e ao número de veículos das rotas construídas pela heurística desenvolvida por Repoussis et al. (2009), pelo MMAS+ILS e pelo ILS.

Nas colunas restantes são mostrados os demais dados referentes aos resultados obtidos pelos algoritmos MMAS+ILS e ILS, ao serem aplicados sobre as instâncias utilizadas neste trabalho, onde a primeira coluna indica o nome da instância utilizada. As colunas de 6 a 9 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 MMAS+ILS. As colunas de 12 a 15 contém os dados acerca da média, desvio padrão, o desvio da melhor solução e desvio do resultado médio em relação aos resultados encontrados pelo ILS. O DMS e o DRM são calculados de acordo com as equações 6 e 7, respectivamente. Nas equações abaixo, fo^* refere-se ao melhor resultado encontrado por Repoussis et al. (2009) e fo_i ao resultado encontrado na *i*-ésima execução do algoritmo proposto.

$$DMS = \frac{fo^* - min_{i=1,\ldots,30}\{fo_i\}}{fo^*} \tag{6}$$

$$DRM = \frac{fo^* - \frac{1}{30}\left(\sum_{i=1}^{30} fo_i\right)}{fo^*} \tag{7}$$

	Repoussis (2009)		MMAS+ILS						ILS					
Conjunto	**DT**	**NV**	**DT**	**NV**	**MDT**	**DP**	**DMS**	**DM**	**DT**	**NV**	**MDT**	**DP**	**DMS**	**DM**
R101	1192,85	19	**1195,13**	**18**	1193,12	19,96	0,00	0,00	**1195,13**	**18**	1198,85	5,23	0,00	-0,01
R102	1079,39	17	1046,46	17	1051,21	5,72	0,03	0,03	**1046,4**	**17**	1050,55	3,93	0,03	0,03
R103	1016,78	13	**981,43**	**13**	987,78	19,76	0,03	0,03	986,16	13	978,30	28,52	0,03	0,04
R104	869,63	9	804,86	9	805,79	30,63	0,07	0,07	**801**	**9**	821,05	33,87	0,08	0,06
R105	1055,04	14	1091,63	13	1068,00	28,01	-0,03	-0,01	**1082,37**	**13**	1077,52	39,58	-0,03	-0,02
R106	1000,95	12	**982,17**	**12**	1006,53	21,29	0,02	-0,01	988,3	12	1009,31	18,94	0,01	-0,01
R107	912,99	10	**899,05**	**10**	914,91	37,16	0,02	0,00	925,5	10	936,45	49,83	-0,01	-0,03
R108	760,3	9	776,87	9	767,34	21,32	-0,02	-0,01	787,6	9	763,13	18,24	-0,04	0,00
R109	934,53	11	958,27	11	934,36	23,83	-0,03	0,00	981,45	11	931,18	20,56	-0,05	0,00
R110	846,49	10	889,86	10	866,48	31,93	-0,05	-0,02	883,08	10	863,41	28,78	-0,04	-0,02
R111	895,21	10	901,19	10	895,10	52,43	-0,01	0,00	906,85	10	899,67	49,32	-0,01	0,00
R112	811,73	9	832,88	9	787,49	20,86	-0,03	0,03	753,07	10	783,37	18,16	0,07	0,03
C101	556,18	10	**556,03**	**10**	581,34	48,60	0,00	-0,05	**556,03**	**10**	558,33	8,75	0,00	0,00
C102	556,18	10	**556,03**	**10**	586,19	49,89	0,00	-0,05	**556,03**	**10**	570,72	29,63	0,00	-0,03
C103	556,18	10	**556,03**	**10**	602,48	63,18	0,00	-0,08	**556,03**	**10**	577,91	40,18	0,00	-0,04
C104	555,41	10	**555,27**	**10**	631,03	49,71	0,00	-0,14	**555,27**	**10**	619,60	65,70	0,00	-0,12
C105	556,18	10	**556,03**	**10**	560,31	13,11	0,00	-0,01	**556,03**	**10**	556,03	0,00	0,00	0,00
C106	556,18	10	**556,03**	**10**	562,41	14,70	0,00	-0,01	**556,03**	**10**	567,02	23,04	0,00	-0,02
C107	556,18	10	**556,03**	**10**	620,72	77,35	0,00	-0,12	**556,03**	**10**	567,02	23,04	0,00	-0,02
C108	555,8	10	**555,65**	**10**	558,97	18,21	0,00	-0,01	**555,65**	**10**	566,97	29,57	0,00	-0,02
C109	555,8	10	**555,65**	**10**	577,38	39,81	0,00	-0,04	**555,65**	**10**	584,33	51,34	0,00	-0,05
RC101	1227,37	14	**1130,36**	**14**	1164,74	19,66	0,08	0,05	1131,64	14	1147,58	11,95	0,08	0,07
RC102	1203,05	12	**1149,76**	**12**	1056,78	31,84	0,04	0,12	1043,87	13	1079,02	32,59	0,13	0,10
RC103	923,15	11	**906,73**	**11**	954,25	30,55	0,02	-0,03	920,89	11	939,69	14,32	0,00	-0,02
RC104	787,02	10	799,56	10	830,67	26,60	-0,02	-0,06	801,52	10	826,68	27,59	-0,02	-0,05
RC105	1195,2	13	**1038,78**	**13**	1055,68	19,22	0,13	0,12	1052,92	13	1067,51	19,83	0,12	0,11
RC106	1095,65	11	**997,35**	**11**	989,09	29,09	0,09	0,10	1004,88	11	993,33	24,80	0,08	0,09
RC107	861,28	11	**842,09**	**11**	885,07	34,25	0,02	-0,03	844,95	11	894,11	34,51	0,02	-0,04
RC108	831,09	10	834,83	10	835,53	26,19	0,00	-0,01	**804,79**	**10**	840,35	30,62	0,03	-0,01

Tabela 1. Resultados para os conjuntos R1, C1 e RC1

	Repoussis (2009)		MMAS+ILS						ILS					
Conjunto	**DT**	**NV**	**DT**	**NV**	**MDT**	**DP**	**DMS**	**DM**	**DT**	**NV**	**MDT**	**DP**	**DMS**	**DM**
R201	1182,43	4	1188,95	4	1249,22	27,55	-0,01	-0,06	1214,68	4	1264,28	34,37	-0,03	-0,07
R202	1149,59	3	1216,8	3	1086,90	36,76	-0,06	0,05	1038,07	4	1071,99	25,31	0,10	0,07
R203	889,12	3	919,46	3	991,09	38,33	-0,03	-0,11	933,96	3	925,47	156,57	-0,05	-0,04
R204	801,46	2	747,75	3	799,78	30,72	0,07	0,00	839,52	2	781,90	28,61	-0,05	0,02
R205	943,33	3	1002,13	3	1056,53	34,93	-0,06	-0,12	995,76	3	1035,03	23,23	-0,06	-0,10
R206	869,27	3	905,18	3	965,14	28,85	-0,04	-0,11	895,74	3	934,98	21,87	-0,03	-0,08
R207	857,08	2	809,73	3	853,52	25,75	0,06	0,00	795,54	3	824,70	19,78	0,07	0,04
R208	700,53	2	709,27	2	757,38	24,16	-0,01	-0,08	725,69	2	762,63	23,76	-0,04	-0,09
R209	851,69	3	875,27	3	934,33	32,47	-0,03	-0,10	855,15	3	985,92	48,49	0,00	-0,16
R210	892,45	3	920,37	3	986,55	54,04	-0,03	-0,11	922,8	3	965,81	33,95	-0,03	-0,08
R211	886,9	2	782,78	3	823,56	21,62	0,12	0,07	771,89	3	809,56	14,16	0,13	0,09
C201	548,51	3	**548,3**	**3**	638,16	63,82	0,00	-0,16	**548,3**	**3**	548,30	0,00	0,00	0,00
C202	548,51	3	**548,3**	**3**	704,69	87,58	0,00	-0,28	**548,3**	**3**	550,27	10,81	0,00	0,00
C203	548,13	3	**546,99**	**3**	695,69	81,73	0,00	-0,27	635,74	3	760,76	53,10	-0,16	-0,39
C204	547,55	3	553,67	3	743,11	99,23	-0,01	-0,36	651,1	3	697,03	39,20	-0,19	-0,27
C205	545,83	3	**545,61**	**3**	673,60	75,24	0,00	-0,23	**545,61**	**3**	545,61	0,00	0,00	0,00
C206	545,45	3	**545,22**	**3**	636,97	72,71	0,00	-0,17	**545,22**	**3**	597,26	52,35	0,00	-0,09
C207	545,24	3	**545,01**	**3**	674,92	65,44	0,00	-0,24	**545,01**	**3**	554,33	21,22	0,00	-0,02
C208	545,28	3	**545,05**	**3**	674,92	65,44	0,00	-0,24	**545,05**	**3**	546,75	9,30	0,00	0,00
RC201	1303,73	4	1323,07	4	1406,19	42,37	-0,01	-0,08	1345,6	4	1378,06	19,37	-0,03	-0,06
RC202	1321,43	3	1370,85	3	1203,87	66,53	-0,04	0,09	1112,3	4	1134,79	20,22	0,16	0,14
RC203	993,29	3	999,87	3	1105,63	47,52	-0,01	-0,11	1027,37	3	1079,60	27,38	-0,03	-0,09
RC204	718,97	3	781,14	3	841,06	24,25	-0,09	-0,17	796,82	3	852,84	26,86	-0,11	-0,19
RC205	1189,84	4	**980,01**	**3**	1051,39	32,60	0,18	0,12	986,9	3	1044,67	22,67	0,17	0,12
RC206	1091,79	3	**1090,43**	**3**	1194,96	87,42	0,00	-0,09	1118,31	3	1173,08	40,75	-0,02	-0,07
RC207	998,7	3	1016,33	3	1089,90	47,63	-0,02	-0,09	1030,02	3	106,77	36,59	-0,03	0,89
RC208	769,4	3	818,78	3	883,28	26,18	-0,06	-0,15	835,93	3	890,13	27,76	-0,09	-0,16

Tabela 2. Resultados para os conjuntos R2, C2 e RC2

Com base nos resultados apresentados, percebe-se que, os algoritmos MMAS+ILS e ILS demonstraram ser competitivos, conseguindo obter novas soluções na maioria dos conjuntos de instâncias testados. Ao ser aplicado sobre os conjuntos de instâncias R1 e R2, o algoritmo MMAS+ILS encontrou novas soluções em mais de 58.33 % das instâncias do conjunto R1, já em relação ao conjunto R2, o MMAS+ILS não foi capaz de encontrar soluções melhores do que as presentes na literatura, apesar dos resultados para alguns dos testes terem sido próximos. No caso dos conjuntos C1 e C2, o algoritmo encontrou soluções melhores do que as da literatura para mais de 100% e 87.5% dos casos, respectivamente. Em relação aos conjuntos RC1 e RC2, o MMAS+ILS encontrou novas soluções em mais de 75% e 25% dos casos.

Ao ser testado, utilizando o conjunto de instâncias R1, o algoritmo ILS apresentou soluções melhores do que as da literatura em 50% das instâncias testadas. O algoritmo ILS também foi aplicado sobre o conjunto R2, apesar de ter apresentado resultados próximos, o 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 ILS encontrou novas soluções em 100% e 75% dos instâncias testadas, respectivamente. O algoritmo 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.

Por meio da comparação entre os resultados obtidos pelos algoritmos MMAS+ILS e ILS, percebeu-se que para o conjunto de instâncias R1 o primeiro foi melhor do que segundo algoritmo em mais de 58.33% das instâncias e empatando em 8.33%. Para o conjunto de instâncias R2 o algoritmo ILS encontrou soluções melhores do que o MMAS+ILS

em 54.55% dos casos. Em relação ao conjunto de instâncias C1 houve empate entre os dois algoritmos em 100% das instâncias. No caso dos conjunto de instâncias C2, o MMAS+ILS encontrou soluções melhores do que o ILS para 25% das instâncias e houve empate entre eles em para 75% das demais instâncias. Considerando os conjuntos RC1 e RC2, o MMAS+ILS apresentou resultados melhores do que ILS em 87.5% e 100% de tais conjuntos de instâncias, respectivamente.

5 Conclusão

Este artigo apresentou um estudo sobre o uso de metaheurísticas para solucionar o PRAVJT, que é um problema NP-difícil, em que foram propostos os algoritmos MMAS+ILS e ILS. O algoritmo MMAS+ILS consiste no combinação das metaheurísticas *MAX-MIN Ant System* e ILS, em que o primeiro é responsável pela geração da solução inicial, de forma que, ela é refinada pelo ILS que utiliza o VND como método de busca local. No caso do algoritmo ILS, ele é semelhante ao método ILS utilizado no MMAS+ILS, porem a solução inicial usada é gerada pela heurística PFIH. Em ambas abordagens, diferentes tipos de estruturas de vizinhança foram usadas. Tanto o MMAS+ILS quanto o ILS demonstraram ser competitivos, sendo capazes de encontrar soluções melhores do que as existentes na literatura para mais de 55.35% e 50% das instâncias. O MMAS+ILS apresentou resultados promissores conseguindo obter soluções melhores do que as da literatura para mais de 58.33 % das instâncias do conjunto R1, 100 % das instâncias do conjunto C1, 87.5 % das instâncias do conjunto C2 e 75 % das instâncias do conjunto RC1. O ILS conseguiu obter um bom desempenho, gerando resultados melhores do que os da literatura para mais de 50%, 100% e 75% das instâncias dos conjuntos R1, C1 e RC1, respectivamente. Ao se comparar tais algoritmos, percebe-se que o MMAS+ILS foi capaz de encontrar resultados melhores do que os alcançados pelo o ILS para mais de 51.8 % das instâncias testadas. Porém houve empate entre esses algoritmos em 28.6 % dos casos. Como há poucos trabalhos na literatura que tratam deste problema em específico, espera-se que este trabalho possa contribuir para com a realização de mais estudos acerca deste tipo de problema. Pretende-se também realizar novos trabalhos que abordam o PRAVJT, tendo em vista, o desenvolvimento de novas metaheurísticas.

Agradecimentos

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

Referências

Brandão, J. (2004). A tabu search algorithm for the open vehicle routing problem. *European Journal of Operation Research*, v. 157, p. 552–564.

Dorigo, M.; Birattari, M. e Stützle, Thomas. (2006). Ant colony optimization: Artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, v. 1(4), p. 28–39.

Dorigo, M.; Maniezzo, V. e Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, v. 26(1), p. 29–41.

Dorigo, M. e Stützle, T. (2003). The ant colony optimization metaheuristic: Algorithms, applications, and advances. Glover, In F. e Kochenberger, G. A., editors, *Handbook of Metaheuristics*, Capítulo 9, p. 251–285. Kluwer Academic Publishers.

Guiyun, Li. (2009). 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.

Letchford, A.N.; Lysgaard, J. e Eglese, R.W. (2006). A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society*, v. 58, p. 1642–1651.

Li, F.; Golden, B. e Wasil, E. (2007). The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers and Operations Research*, v. 34, p. 2918–2930.

Li, X.Y.; Tian, P. e Leung, S.C.H. (2009). 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.

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

MirHassani, S.A. e Abolghasemi, N. (2011). A particle swarm optimization algorithm for open vehicle routing problem. *Expert Systems with Applications*, v. 38, p. 11547–11551.

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

Repoussis, P.P.; Tarantilis, C.D. e Ioannou, G. (2006). The open vehicle routing problem with time windows. *Journal of the Operational Research Society*, v. 58, p. 1–13.

Repoussis, P.P.; Tarantilis, C.D. e Ioannou, G. (2009). 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.

Schrage, L. (1981). Formulation and structure of more complex/realistic routing and scheduling problems. *Networks*, v. 11, p. 229–232.

Solomon, MM. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, v. 35, p. 2544–265.

Stützle, Thomas. e Hoos, H. H. (1996). Improving the ant system: A detailed report on the max-min ant system. *FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep.*

Stützle, Thomas. e Hoos, H. H. (2000). Max-min ant system. *Future Generation Computer Systems*, v. 16, p. 889–914.