

UMA METODOLOGIA DE HIBRIDIZAÇÃO HEURÍSTICA PARA RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA E ENTREGA E JANELA DE TEMPO

Aline Aparecida de Carvalho Gonçalves, Sérgio Ricardo de Souza
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas, 7675, CEP 30510-000, Belo Horizonte (MG), Brasil
acg.aline@gmail.com, sergio@dppg.cefetmg.br

Carlos Alexandre Silva
Departamento de Computação - Instituto Federal de Minas Gerais
Avenida Serra da Piedade, 299, Morada da Serra, Sabará (MG), Brasil
carlos.silva@ifmg.edu.br

RESUMO

Este trabalho propõe uma metodologia para a resolução do Problema de Roteamento de Veículos com Coleta e Entrega e Janela de Tempo (PRVCEJT). A solução consiste em encontrar a frota mínima e as rotas de menor distância, capazes de atender clientes dispersos geograficamente que possuem demanda de coleta ou entrega de produtos. Cada atendimento é restrito pela ordem de precedência e emparelhamento das visitas e pela janela de tempo para início do serviço. A solução inicial da solução proposta é gerada através da implementação da fase de construção *Greedy Randomized Adaptive Search Procedure* (GRASP). Em seguida, essa solução é refinada pelos algoritmos ILS-VND e ILS-RVND. O primeiro combina os procedimentos heurísticos *Iterated Local Search* (ILS) e *Variable Neighborhood Descent* (VND) e o segundo utiliza o método de busca local *Random Variable Neighborhood Descent* (RVND). Os algoritmos foram avaliados utilizando instâncias clássicas da literatura e os resultados produzidos mostraram-se bastante competitivos.

PALAVRAS CHAVE. *Problema de Roteamento de Veículos com Coleta e Entrega e Janela de Tempo. Iterated Local Search. Variable Neighborhood Descent. Random Variable Neighborhood Descent.*

Área Principal: Metaheurísticas (MH) e Otimização Combinatória (OC)

ABSTRACT

This work addresses the Vehicle Routing Problem with Pickup and Delivery and Time Windows (VRPPDTW) through metaheuristics. The objective of this problem is to find the minimum fleet and a minimum total travel distance to serve customers who have demand for pickup or delivery of products. Each service is restricted by the order of precedence and pairing of visits and the time window to start the service. The initial solution is created by the implementation of the construction phase *Greedy Randomized Adaptive Search Procedure* (GRASP). This solution is refined by the algorithms ILS-VND and ILS-RVND. The first algorithm combines the procedures *Iterated Local Search* (ILS) and *Variable Neighborhood Descent* (VND) and the second differs from the ILS-VND by using local search method *Random Variable Neighborhood Descent* (RVND) instead of VND. The algorithms were tested in benchmark instances from literature for VRPPDTW and the results produced were very competitive.

KEYWORDS. *Vehicle Routing Problem with Pickup and Delivery and Time Window. Iterated Local Search. Variable Neighborhood Descent. Random Variable Neighborhood Descent. Main Area: Metaheuristics (MH) and Combinatorial Optimization (OC)*

1. Introdução

O Problema de Roteamento de Veículos (PRV), introduzido por Dantzig e Ramser [1959], é um dos problemas mais estudados em Otimização Combinatória. Seu objetivo é construir rotas de custo mínimo para atendimento a um conjunto de clientes dispersos geograficamente, por intermédio de uma frota de veículos inicialmente situada no depósito. A única restrição existente no PRV é a capacidade de carga dos veículos.

O Problema de Roteamento de Veículos com Coleta e Entrega e Janela de Tempo (PRV-CEJT) é uma das variações do PRV clássico. Neste problema, cada cliente possui um tipo de demanda por produtos: coleta ou entrega. Há, dessa maneira, um par de clientes coleta-entrega, de modo que produtos devem coletados em um e entregues em outro. Assim, o cliente com demanda de coleta deve ser visitado antes do respectivo cliente com demanda de entrega e o mesmo veículo deve visitar esse par de clientes, sem retornar ao depósito no meio do trajeto [Li e Lim, 2001]. Além disso, cada cliente, assim como o depósito, apresenta um instante de tempo mínimo e máximo para o atendimento ser iniciado, denominado janela de tempo. O tempo de serviço indica a duração da coleta ou da entrega a ser realizada. O objetivo do PRVCEJT é encontrar um conjunto de rotas para atendimento de um grupo de clientes, atendendo às restrições que definem o problema, minimize o número de veículos utilizados e a distância total percorrida por eles. Em Parragh et al. [2008a] e Parragh et al. [2008b] são apresentadas revisões a respeito dessa classe de problemas de roteamento.

Considera-se que o PRVCEJT tem importância teórica e prática. Teórica, pois é um problema de otimização combinatória de difícil solução, que não pode ser resolvido em tempo polinomial. Em outras palavras, pertence à classe de problemas NP-Difícil, conforme Ropke e Cordeau [2009], e, na presença de janela de tempo, cada verificação de factibilidade é NP-Completo. Prática, pois existem muitas situações do setor logístico em que ele aparece, como, por exemplo, o serviço de coleta e entrega de documentos.

Muitos trabalhos presentes na literatura para a resolução desse problema apresentam abordagens por heurísticas e metaheurísticas [Li e Lim, 2001], [Braysy e Gendreau, 2005a], [Braysy e Gendreau, 2005b], [Lu e Dessouky, 2006], [Dondo et al., 2008], [Carabetti, 2010], [Gonçalves e de Souza, 2014], [Gonçalves et al., 2015].

Este trabalho apresenta uma metodologia para a solução do PRVCEJT baseada na utilização de heurísticas e metaheurísticas. Para a construção da solução inicial do problema, foi utilizada a fase de construção da metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP). Para o refinamento da solução, foram empregados dois algoritmos: ILS-VND e ILS-RVND. O primeiro foi desenvolvido através de uma combinação da metaheurística *Iterated Local Search* (ILS) associada à técnica de busca local *Variable Neighborhood Descent* (VND) e o segundo utiliza o método *Random Variable Neighborhood Descent* (RVND) ao invés do VND.

A próxima seção deste artigo apresenta uma visão global do problema abordado. A terceira seção descreve a metodologia aplicada ao desenvolvimento dos algoritmos propostos. A quarta seção mostra os resultados computacionais obtidos pelos dois algoritmos. A última seção apresenta as conclusões advindas deste artigo.

2. Problema de Roteamento de Veículos com Coleta e Entrega e Janela de Tempo

O objetivo do PRVCEJT é encontrar um conjunto de rotas para atendimento das demandas de coleta e entrega de mercadorias de um grupo de clientes dispersos geograficamente. As rotas serão percorridas por uma frota homogênea de veículos e o seu processo de construção deverá priorizar a minimização do número de veículos utilizados e a distância total percorrida por eles.

A modelagem matemática do PRVCEJT pode ser realizada através de um grafo $G = (V, A)$, em que o conjunto $V = \{v_0, v_1, \dots, v_n\}$ representa os clientes. O depósito é representado pelo elemento v_0 , ponto de partida e retorno de todos os veículos da frota. Define-se também P como o conjunto de clientes com demanda de coleta de mercadorias e D o conjunto de clientes com demanda de entrega. Cada $v_i \in V$ possui uma demanda q_i associada, sendo $q_i > 0$ para $v_i \in P$ e $q_i < 0$ para $v_i \in D$, um tempo de serviço s_i e uma janela de tempo $[e_i, l_i]$. Para o depósito v_0 ,

$q_0 = 0$ e $s_0 = 0$. Entre um par de clientes (v_i, v_j) , $(i \neq j, i, j = 1, 2, \dots, n)$, há uma distância d_{ij} e um tempo de viagem t_{ij} , sendo $d_{ij} > 0$ e $t_{ij} > 0$. Então, o conjunto A de arestas pode ser definido como $A = \{(v_i, v_j) \mid v_i, v_j \in V; v_i \neq v_j; t_{0i} + s_i + t_{ij} \leq l_j\}$ e representa o elo entre os clientes. Se um veículo chegar ao cliente v_i antes de e_i , então deve aguardar até e_i para iniciar o serviço. O tempo total de percurso de uma rota não pode ser maior que l_0 , que representa o momento de encerramento das atividades no depósito. Veja em Mitrovic-Mini [1998] a modelagem matemática adotada neste trabalho com maiores detalhes.

Nesse trabalho consideram-se duas funções objetivo principais para o PRVCEJT, organizadas de maneira hierarquizada: a primeira, consiste em minimizar o número de veículos empregados na solução; e a segunda, em minimizar a distância total percorrida pelos veículos, dada pela soma das distâncias entre as cidades adjacentes.

3. Metodologia proposta

Este artigo apresenta dois algoritmos propostos para solução do PRVCEJT. Nesta seção estão descritas as características do ILS-VND e ILS-RVND, divididas em: representação computacional da solução, funções de avaliação, construção da solução inicial, estruturas de vizinhança e detalhamento dos algoritmos propostos.

3.1. Representação computacional da solução

A representação computacional de uma solução do PRVCEJT é dada por uma estrutura de dados que armazena a soma das distâncias totais percorridas em cada rota, o número total de veículos empregados e o conjunto de rotas. Cada rota da solução, também definida como uma estrutura de dados, mantém uma lista de clientes, dispostos de acordo com a ordem de visita, e a distância total percorrida na rota.

Para cada cliente do conjunto a ser atendido são conhecidos: localização geográfica, tipo de demanda (coleta ou entrega), local de origem da carga (ponto de coleta), local de destino da carga (ponto de entrega), quantidade de carga (demanda de mercadorias), instante inicial e final da janela de tempo (horário mínimo e máximo para início do atendimento) e duração total do atendimento.

3.2. Funções de avaliação

Uma solução s para o PRVCEJT é avaliada por uma função $f(s)$, hierarquizada em dois níveis:

- (a) número total de veículos utilizados;
- (b) distância total percorrida, dada por:

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

A expressão (1) calcula a distância total percorrida por um veículo em uma determinada rota s , sendo que o custo de deslocamento de um cliente i para o cliente j é definido por d_{ij} . A distância total da solução é dada pela soma das distâncias percorridas em cada rota.

3.3. Construção da solução inicial

A metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP), proposta por Feo e Resende [1995], é um algoritmo iterativo, composto por duas fases: construção da solução e refinamento. Na fase de refinamento é empregado um mecanismo de busca local, com o objetivo de encontrar o ótimo local da solução inicial gerada. Cada iteração do algoritmo produz uma solução ótima local, que não é empregada em iterações futuras. Ao final, o procedimento efetua uma comparação e retorna a melhor solução obtida entre todas geradas.

A solução inicial do PRVCEJT é gerada através de uma variação da fase de construção do algoritmo GRASP. Para atender à característica gulosa da fase de construção dessa metaheurística é

utilizada a função de avaliação da heurística *Push-Forward Insertion Heuristic* (PFIH), com os valores dos parâmetros conforme propostos empiricamente por Solomon [1987]. Já para a característica aleatória, é realizado um sorteio de um dos clientes contidos na Lista Restrita de Candidatos (LRC).

A variação proposta refere-se à atualização e seleção dos clientes na LRC. Assim, os clientes são selecionados e removidos aos pares da lista, de modo a atender ao requisito de emparelhamento exigido no PRVCEJT. O pseudocódigo desse procedimento utilizado para a construção da solução é apresentado no Algoritmo 1.

Algoritmo 1: *Solucao GRASP()*

Entrada: lista de clientes
Saída: Solução do PRVCEJT

```

1 início
2   Calcule o custo de inserção de cada cliente;
3   Crie lista de clientes ordenados crescentemente pelo custo;
4   Crie a LRC;
5   enquanto existir algum cliente na lista faça
6     Selecione aleatoriamente um cliente da lista LRC;
7     Selecione o cliente com demanda emparelhada ao cliente selecionado;
8     Atualize a lista ordenada e a LRC;
9     enquanto existir rotas não vazias na solução faça
10      Percorra cada rota da solução;
11      Insira o cliente com demanda de coleta em uma posição da rota;
12      se inserção é factível então
13        Insira o cliente emparelhado em uma posição posterior;
14        se inserção é factível então
15          Calcule a distância total percorrida na rota;
16          fim
17      fim
18    fim
19    se não existir uma inserção factível para os dois clientes então
20      Crie uma nova rota com o par de clientes selecionados;
21    fim
22  fim
23 fim
  
```

3.4. Estruturas de vizinhança

A estrutura de vizinhança na busca local é obtida através dos seguintes movimentos e estratégias:

- (a) movimento de troca intrarota com cliente com demanda de coleta: consiste em trocar um cliente com demanda de coleta com todos os outros clientes da mesma rota que estão nas posições anteriores ao cliente com demanda de entrega emparelhada;
- (b) movimento de troca intrarota com cliente com demanda de entrega: consiste em trocar um cliente com demanda de entrega com todos os outros clientes da mesma rota que estão nas posições posteriores ao cliente com demanda de coleta emparelhada;
- (c) movimento de troca inter-rota: que consiste em permutar um par de clientes de uma rota com um par de rota diferente, respeitando-se a restrição de emparelhamento e de precedência;

- (d) movimento de realocação: que consiste em remover um par de clientes com demandas emparelhadas de uma rota da solução e inseri-lo em outra rota;
- (e) estratégia elimina rota: empregada na tentativa de reduzir uma rota através da realocação dos seus clientes nas demais rotas da solução;
- (f) estratégia esvazia rota: empregada para reduzir o tamanho da menor rota da solução, através do aproveitamento das realocações factíveis realizadas.

3.5. Algoritmos propostos

3.5.1. Algoritmo ILS-VND

O algoritmo híbrido ILS-VND combina os procedimentos heurísticos *Iterated Local Search* (ILS), proposto por Lourenço et al. [2010], com o método de busca local *Variable Neighborhood Descent* (VND) [Hansen e Mladenović, 2014] para refinamento de soluções do PRVCEJT.

O ILS consiste na aplicação de perturbações em ótimos locais obtidos através de um mecanismo de busca local e consequente exploração de todo o espaço de busca do problema. O algoritmo ILS está organizado em três etapas:

- (1) Técnica de Busca Local aplicada a uma solução inicial;
- (2) Metodologia de perturbação;
- (3) Critério de Aceitação.

Já o VND busca o refinamento de soluções, através da exploração do espaço de soluções de um determinado problema, realizando trocas sistemáticas nas estruturas de vizinhança definidas.

O algoritmo ILS-VND aplica dois níveis de perturbação em uma solução, utilizando os movimentos descritos na seção 3.4, conforme sequência a seguir:

- (a) nível 1: movimento de troca intrarota (uma troca de cliente com demanda de coleta e uma troca de cliente com demanda de entrega), seguido de movimento de troca inter-rota;
- (b) nível 2: movimento de realocação.

A perturbação aplicada só é aceita se a solução perturbada é factível. As rotas que sofrerão perturbação são escolhidas aleatoriamente a cada iteração, o que garante a diversificação da solução para execução do procedimento de busca local. O Algoritmo 2 apresenta o pseudocódigo da implementação realizada para o ILS-VND.

O processo de busca local é repetido em todas as iterações do algoritmo, após a aplicação de uma perturbação. O Algoritmo ILS-VND utiliza o método de busca local VND, que emprega todos os movimentos e estratégias descritos na seção 3.4.

Para realizar a busca pelo melhor vizinho, é utilizado o método da descida completa (toda a vizinhança da solução é analisada) em cada uma das estruturas de vizinhança empregadas. Ao final de cada iteração, a solução obtida é analisada de acordo com as funções de avaliação. Se ela for melhor que a anterior, será escolhida como ponto de partida para a próxima iteração. Se a solução obtida for de piora, o procedimento de perturbação é reaplicado na mesma solução.

O Algoritmo 3 mostra o método de busca local VND indicado nas linhas 3, 6 e 11 do Algoritmo 2.

Algoritmo 2: Procedimento ILS-VND

Entrada: $MaxIter, s, f()$

```

1 início
2    $iter = 0;$ 
3   GeraSolucaoInicial( $s_0$ );
4   BuscaLocalVND( $s_0, s$ );
5   enquanto  $iter \leq MaxIter$  faça
6     PerturbaSolucaoNivel1( $s, s_1$ );
7     BuscaLocalVND( $s_1, s'_1$ );
8     se  $f(s'_1) < f(s)$  então
9       |  $s = s'_1;$ 
10    senão
11      PerturbaSolucaoNivel2( $s, s_2$ );
12      BuscaLocalVND( $s_2, s'_2$ );
13      se  $f(s'_2) < f(s)$  então
14        |  $s = s'_2;$ 
15      senão
16        |  $iter = iter + 1;$ 
17      fim
18    fim
19  fim
20  Retorne  $s;$ 
21 fim
    
```

3.5.2. Algoritmo ILS-RVND

O processo iterativo no algoritmo ILS-RVND é idêntico ao processo do ILS-VND, descrito na Seção 3.5.1. A modificação no algoritmo refere-se ao método de busca local implementado nas linhas 3, 6 e 11 do Algoritmo 2, denominado *Random Variable Neighborhood Descent* (RVND).

O RVND é uma variação do VND, proposta por Souza et al. [2010] e Subramanian et al. [2010], em que há uma escolha aleatória das estruturas de vizinhança empregadas durante o processamento da busca. Ou seja, sempre que uma determinada vizinhança não melhorar a solução corrente, o método seleciona aleatoriamente outra vizinhança para continuar a busca pelo espaço de soluções.

No ILS-RVND são utilizadas as seis estruturas de vizinhança propostas neste trabalho. Entretanto, não é estabelecida a ordem de aplicação de cada uma dessas estruturas, ou seja, a cada chamada do procedimento, uma ordem de processamento é definida aleatoriamente. O critério de aceite da nova solução são os valores das funções de avaliação. Se a nova solução for melhor que a solução corrente, ela é utilizada no próximo passo do algoritmo e a busca é reiniciada com a primeira estrutura de vizinhança; caso contrário, a busca continua na estrutura subsequente. O processo se encerra quando todas as vizinhanças forem analisadas.

O Algoritmo 4 mostra o funcionamento do método a partir de uma solução s para o PRVCEJT.

4. Resultados computacionais

Nesta seção são apresentados os resultados computacionais obtidos pelos algoritmos ILS-VND e ILS-RVND. Eles foram desenvolvidos na linguagem C, utilizando o ambiente C++ Builder, versão 2006. Os testes foram executados em um computador com processador Intel Core i3 2,4 GHz, 4GB de memória RAM, sistema operacional Windows 7 Home Premium de 32 bits.

Algoritmo 3: Procedimento Busca Local VND

```

Entrada:  $s, f()$ 
1 início
2    $k = 1;$ 
3   enquanto  $k \leq 6$  faça
4     se  $k = 1$  então
5        $s' = \text{BuscaLocalTrocaInterRota}(s);$ 
6     fim
7     se  $k = 2$  então
8        $s' = \text{BuscaLocalTrocaIntrarotaColeta}(s);$ 
9     fim
10    se  $k = 3$  então
11       $s' = \text{EstrategiaEsvaziaRota}(s);$ 
12    fim
13    se  $k = 4$  então
14       $s' = \text{BuscaLocalTrocaIntrarotaEntrega}(s);$ 
15    fim
16    se  $k = 5$  então
17       $s' = \text{BuscaLocalRealocacao}(s);$ 
18    fim
19    se  $k = 6$  então
20       $s' = \text{EstrategiaEliminaRota}(s);$ 
21    fim
22    se  $f(s') < f(s)$  então
23       $s = s';$ 
24       $k = 1;$ 
25    fim
26    senão
27       $k = k + 1;$ 
28    fim
29  fim
30  Retorne  $s;$ 
31 fim

```

Para testar o algoritmo, foram utilizadas as instâncias de Li e Lim [2001], usadas como referência de desempenho de algoritmos para solução do PRVCEJT. Essas instâncias são divididas em seis classes distintas: LC1, LC2, LR1, LR2, LRC1 e LRC2. Todos os problemas possuem 100 clientes com depósito, capacidade do veículo e janela de tempo. Clientes das instâncias LC são distribuídos na forma de *cluster*. Nos problemas LR, os clientes são postos de forma aleatória. Nas instâncias LRC, os clientes são parcialmente colocados na forma de *cluster* e parcialmente aleatórios. De acordo com Li e Lim [2001], as instâncias LC1, LR1 e LRC1 são problemas de curto horizonte de programação, enquanto as instâncias LC2, LR2 e LRC2 são de longo horizonte de programação.

Os resultados obtidos estão apresentados nas Tabelas 1 a 6. Nestas tabelas, NV significa o número de veículos utilizados e DT a distância total percorrida. Esses valores estão comparados com os melhores resultados da literatura determinados para cada instância e encontrados em <http://www.sintef.no/Projectweb/TOP/PDPTW/Li--Lim-benchmark/100-customers/>.

Algoritmo 4: Procedimento Busca Local RVND

```

Entrada:  $s, f()$ 
1 início
2    $v = 1, 2, 3, 4, 5, 6;$ 
3   Embaralhe( $v$ );
4    $k = 1;$ 
5   enquanto  $k \leq 6$  faça
6     se  $k = v(1)$  então
7        $s' = \text{BuscaLocalTrocaIntrarrotaColeta}(s);$ 
8     fim
9     se  $k = v(2)$  então
10       $s' = \text{BuscaLocalTrocaIntrarrotaEntrega}(s);$ 
11    fim
12    se  $k = v(3)$  então
13       $s' = \text{BuscaLocalTrocaInterRota}(s);$ 
14    fim
15    se  $k = v(4)$  então
16       $s' = \text{BuscaLocalRealocacao}(s);$ 
17    fim
18    se  $k = v(5)$  então
19       $s' = \text{EstrategiaEliminaRota}(s);$ 
20    fim
21    se  $k = v(6)$  então
22       $s' = \text{EstrategiaEsvaziaRota}(s);$ 
23    fim
24    se  $f(s') < f(s)$  então
25       $s = s';$ 
26       $k = 1;$ 
27    fim
28    senão
29       $k = k + 1;$ 
30    fim
31  fim
32  Retorne  $s;$ 
33 fim

```

Um resumo dos resultados finais obtidos pelos algoritmos ILS-VND e ILS-RVND, em cada classe de instâncias, é mostrado na Tabela 7.

Pode-se notar, na comparação dos dois algoritmos, ILS-VND e ILSRVND, que o método de busca local VND combinado ao ILS foi capaz de produzir melhores resultados que o método RVND, ambos partindo de uma solução inicial gerada através da fase de construção GRASP.

Finalmente, os resultados computacionais obtidos pelos dois algoritmos comprovam a viabilidade da metodologia proposta neste trabalho. Os melhores resultados foram atingidos pelo algoritmo ILS-VND, que está somente 3,98% acima dos melhores resultados conhecidos para o número total de veículos e 2,88% acima em relação à distância total percorrida. Observa-se ainda que o algoritmo ILS-RVND obteve o melhor resultado no objetivo de minimizar a distância total percorrida. Porém, ele não pode ser considerado como o melhor resultado obtido pela metodologia proposta, pois uma solução para o PRVCEJT é avaliada por dois objetivos distintos em ordem

Tabela 1: Resultados das instâncias da classe LC1

Instância	Literatura			ILS-VND		ILS-RVND	
	NV	DT	Autor	NV	DT	NV	DT
LC101	10	828,94	Li e Lim [2001]	10	828,94	10	828,94
LC102	10	828,94	Li e Lim [2001]	10	828,94	10	828,94
LC103	09	1035,35	Bent e Hentenryck [2003]	09	1065,11	10	827,86
LC104	09	860,01	Hasle e Kloster [2007]	09	907,43	09	1093,41
LC105	10	828,94	Li e Lim [2001]	10	828,94	10	828,94
LC106	10	828,94	Li e Lim [2001]	10	828,94	10	828,94
LC107	10	828,94	Li e Lim [2001]	10	828,94	10	828,94
LC108	10	826,44	Li e Lim [2001]	10	826,44	10	826,44
LC109	09	827,82	Bent e Hentenryck [2003]	09	827,82	09	827,82
Total	87	7694,32		87	7771,50	88	7720,23

Tabela 2: Resultados das instâncias da classe LC2

Instância	Literatura			ILS-VND			
	NV	DT	Autor	NV	DT	NV	DT
LC201	03	591,56	Li e Lim [2001]	03	591,56	03	591,56
LC202	03	591,56	Li e Lim [2001]	03	591,56	03	591,56
LC203	03	585,56	Li e Lim [2001]	03	601,85	03	585,56
LC204	03	590,6	Hasle e Kloster [2007]	03	669,52	04	680,58
LC205	03	588,88	Li e Lim [2001]	03	588,88	03	588,88
LC206	03	588,49	Li e Lim [2001]	03	588,49	03	588,49
LC207	03	588,29	Li e Lim [2001]	03	588,29	03	588,29
LC208	03	588,32	Li e Lim [2001]	03	588,32	03	588,32
Total	24	4713,26		24	4808,46	25	4803,24

Tabela 3: Resultados das instâncias da classe LR1

Instância	Literatura			ILS-VND			
	NV	DT	Autor	NV	DT	NV	DT
LR101	19	1650,8	Li e Lim [2001]	19	1650,8	19	1650,80
LR102	17	1487,57	Li e Lim [2001]	17	1489,29	17	1487,57
LR103	13	1292,68	Li e Lim [2001]	13	1292,68	13	1292,68
LR104	09	1013,39	Li e Lim [2001]	10	1062,28	10	1033,54
LR105	14	1377,11	Li e Lim [2001]	14	1377,11	14	1377,11
LR106	12	1252,52	Li e Lim [2001]	12	1252,52	12	1252,52
LR107	10	1111,31	Li e Lim [2001]	10	1111,31	10	1111,31
LR108	09	968,97	Li e Lim [2001]	09	968,97	09	968,97
LR109	11	1208,96	Hasle e Kloster [2007]	12	1241,25	12	1253,38
LR110	10	1159,35	Li e Lim [2001]	12	1191,65	12	1197,27
LR111	10	1108,9	Li e Lim [2001]	11	1114,95	11	1131,71
LR112	09	1003,77	Li e Lim [2001]	10	1042,06	09	1003,77
Total	143	14635,33		149	14794,87	148	14760,63

de prioridade: minimização do número de veículos empregados e minimização da distância total percorrida.

Tabela 4: Resultados das instâncias da classe LR2

Instância	Literatura			ILS-VND			
	NV	DT	Autor	NV	DT	NV	DT
LR201	04	1253,23	Hasle e Kloster [2007]	04	1321,65	05	1275,95
LR202	03	1197,67	Li e Lim [2001]	04	1368,62	04	1289,38
LR203	03	949,4	Li e Lim [2001]	03	949,40	03	1017,67
LR204	02	849,05	Li e Lim [2001]	03	1018,95	03	1006,74
LR205	03	1054,02	Li e Lim [2001]	03	1054,02	03	1054,05
LR206	03	931,63	Li e Lim [2001]	03	931,63	03	931,63
LR207	02	903,06	Li e Lim [2001]	03	1023,31	03	1028,18
LR208	02	734,85	Li e Lim [2001]	03	890,22	02	734,85
LR209	03	930,59	Hasle e Kloster [2007]	03	1032,76	04	1007,75
LR210	03	964,22	Li e Lim [2001]	03	964,22	03	964,22
LR211	02	911,52	Hasle e Kloster [2007]	03	989,86	03	939,23
Total	30	10679,24		35	11544,64	36	11249,62

Tabela 5: Resultados das instâncias da classe LRC1

Instância	Literatura			ILS-VND			
	NV	DT	Autor	NV	DT	NV	DT
LRC101	14	1708,8	Li e Lim [2001]	15	1703,21	14	1715,56
LRC102	12	1558,07	Hasle e Kloster [2007]	12	1558,07	13	1613,25
LRC103	11	1258,74	Li e Lim [2001]	11	1272,97	11	1258,74
LRC104	10	1128,74	Li e Lim [2001]	10	1128,74	10	1139,50
LRC105	13	1637,62	Li e Lim [2001]	13	1637,62	14	1645,49
LRC106	11	1424,73	Hasle e Kloster [2007]	12	1500,84	11	1424,73
LRC107	11	1230,15	Li e Lim [2001]	11	1230,15	11	1230,15
LRC108	10	1147,43	Hasle e Kloster [2007]	10	1147,43	11	1176,71
Total	92	11094,28		94	11179,03	95	11204,13

Tabela 6: Resultados das instâncias da classe LRC2

Instância	Literatura			ILS-VND1		ILS-VND2	
	NV	DT	Autor	NV	DT	NV	DT
LRC201	04	1406,94	Hasle e Kloster [2007]	04	1406,94	05	1518,66
LRC202	03	1374,27	Li e Lim [2001]	04	1396,80	04	1432,23
LRC203	03	1089,07	Li e Lim [2001]	04	1240,30	04	1220,15
LRC204	03	818,66	Hasle e Kloster [2007]	03	818,66	03	818,66
LRC205	04	1302,2	Li e Lim [2001]	04	1302,20	04	1302,20
LRC206	03	1159,03	Hasle e Kloster [2007]	04	1219,98	04	1244,23
LRC207	03	1062,05	Hasle e Kloster [2007]	03	1062,05	03	1062,05
LRC208	03	852,76	Li e Lim [2001]	03	1004,85	04	1045,35
Total	26	9064,98		29	9451,78	31	9643,53

5. Conclusões

Este artigo apresentou dois algoritmos heurísticos híbridos para resolver o PRVCEJT. Esses algoritmos realizam o refinamento de uma solução inicial do problema, gerada através da fase de construção GRASP.

Para avaliar a viabilidade da metodologia proposta, foram utilizadas as 56 instâncias de teste apresentadas por Li e Lim [2001]) para o problema. Os algoritmos ILS-VND e ILS-RVND produziram, de forma isolada, alguns resultados idênticos aos melhores da literatura, sendo que o

Tabela 7: Resumo dos resultados finais do algoritmo ILS-VND

Classe	Literatura		ILS-VND		ILS-RVND	
	NV	DT	NV	DT	NV	DT
LC1	87	7694,32	87	7771,50	88	7720,23
LC2	24	4713,26	24	4808,47	25	4803,24
LR1	143	14635,33	149	14794,87	148	14760,63
LR2	30	10679,24	35	11544,64	36	11249,62
LRC1	92	11094,28	94	11179,03	95	11204,13
LRC2	26	9064,98	29	9451,78	31	9643,53
Total	402	57881,41	418	59550,29	423	59381,38

ILS-VND foi capaz de atingir 32 resultados idênticos.

Em vista das análises feitas nos resultados obtidos, pode-se concluir que o algoritmo ILS-VND obteve os melhores resultados finais e variou, em média, 3,43% acima das melhores metodologias existentes. Trata-se, portanto, de um algoritmo eficiente que supera os resultados obtidos por outras metodologias propostas na literatura para solução do PRVCEJT. Apesar dos resultados satisfatórios, ainda existem estratégias que não foram incorporadas à metodologia e que podem ser suficientes para melhorar seu desempenho, por exemplo, a criação de diferentes movimentos de exploração da vizinhança e eliminação de rotas.

Referências

- Bent, R. e Hentenryck, P. V. (2003). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33:875–893.
- Braysy, O. e Gendreau, M. (2005a). Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Science*, 39(1):104–118.
- Braysy, O. e Gendreau, M. (2005b). Vehicle routing problem with time windows, part ii: Metaheuristics. *Transportation Science*, 39(1):119–139.
- Carabetti, E. G. (2010). Metaheurística colônia de formigas aplicada ao problema de roteamento de veículos com coleta e entrega e janela de tempo. Dissertação de mestrado, Programa de Pós-Graduação em Modelagem Matemática e Computacional (CEFET-MG), Belo Horizonte.
- Dantzig, G. B. e Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1): 80–91.
- Dondo, R., Mendez, C., e Cerda, J. (2008). Optimal management of logistic activities in multi-site environments. *Computers and Chemical Engineering*, 32:2547–2569.
- Feo, T. A. e Resende, M. G. C. (1995). Greedy randomized adaptive search. *Journal of Global Optimization*, 3:109–133.
- Gonçalves, A. A. C. e de Souza, S. R. (2014). Resolução do problema de roteamento de veículos com coleta e entrega e janela de tempo via metaheurística iterated local search com busca local em vizinhança variável. In *Proceedings of the XXXV Ibero-Latin American Congress on Computational Methods in Engineering*.
- Gonçalves, A. A. C., de Souza, S. R., e Silva, C. A. (2015). Um algoritmo híbrido para a solução do problema de roteamento de veículos com coleta e entrega e janela de tempo. In *Anais do XLVII Simpósio Brasileiro de Pesquisa Operacional*.

- Hansen, P. e Mladenović, N. (2014). Variable neighborhood search. In Burke, E. K. e Kendall, G., editors, *Search Methodologies*, p. 313–337. Springer US.
- Hasle, G. e Kloster, O. (2007). Industrial vehicle routing problem. *Geometric Modelling, Numerical Simulation, and Optimization*, 3:397–435.
- Li, H. e Lim, A. (2001). A metaheuristic for the pickup and delivery problem with time windows. In *Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence – ICTAI-2001*, p. 160–167, Washington, DC, USA.
- Lourenço, H. R., Martin, O. C., e Stutzle, T. (2010). Iterated local search: Framework and applications. In Gendreau, M. e Potvin, J., editors, *Handbook of Metaheuristics*, p. 363–397. Springer US.
- Lu, Q. e Dessouky, M. M. (2006). A new insertion-based construction heuristic for solving the pickup and delivery problem with time-windows. *European Journal of Operational Research*, 175:672–687.
- Mitrovic-Mini, S. (1998). Pickup and delivery problem with time windows: a survey. Technical report, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada.
- Parragh, S., Doerner, K., e Hartl, R. (2008a). A survey on pickup and delivery problems, part i: Transportation between customers and depot. *Journal for Betriebswirtschaft*, 58(2):21–51.
- Parragh, S., Doerner, K., e Hartl, R. (2008b). A survey on pickup and delivery problems, part ii: Transportation between pickup and delivery locations. *Journal for Betriebswirtschaft*, 58(2): 81–117.
- Ropke, S. e Cordeau, J. F. (2009). Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3):267–286.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.
- Souza, M. J. F., Coelho, I. M., Ribas, S., Santos, H. G., e Merschmann, L. H. C. (2010). A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research*, 207(2):1041–1051.
- Subramanian, A., Drummond, L. M. A., Bentes, C., Ochi, L. S., e Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–1911.