

Modelos e meta-heurística para o problema de coleta e entrega com janelas de tempo, frota heterogênea e múltiplos produtos

Diógenes Gasque

Pedro Munari

Departamento de Engenharia da Produção
Universidade Federal de São Carlos, São Carlos, Brasil
diogeneshfg@gmail.com, munari@dep.ufscar.br

RESUMO

Neste trabalho, são propostos dois modelos de programação inteira-mista para o problema de coleta e entrega com janelas de tempo, frota heterogênea e múltiplos produtos. Os modelos são obtidos a partir da extensão dos modelos de dois- e três-índices propostos na literatura para a variante clássica do problema de coleta e entrega. Além disso, propõe-se uma meta-heurística adaptativa baseada em *large neighborhood search*. Os resultados de experimentos computacionais usando instâncias da literatura indicam que o modelo de dois-índices é superior ao de três-índices, sendo adequado para obtenção de soluções ótimas de instâncias de pequeno porte, por meio de software de otimização de propósito geral. A meta-heurística proposta se mostrou eficaz na resolução das instâncias conseguindo boas soluções em tempos aceitáveis.

PALAVRAS CHAVE. Roteamento de veículos; coleta e entrega; janelas de tempo; frota heterogênea; múltiplos produtos.

Área Principal: Logística e Transportes

ABSTRACT

In this paper, we propose two mixed-integer programming models to the pickup and delivery problem with time windows, heterogeneous fleet and multiple products. The models are obtained from the extension of the two- and three-index models proposed in the literature for classic variants of the pickup and delivery problem. In addition, we propose an adaptive meta-heuristic based on *large neighborhood search*. The results of computational experiments based on instances from literature, indicate that the two-index model is superior to three-index, being suitable for obtaining optimal solutions of small instances, through general purpose optimization software. The proposed meta-heuristic proved to be effective in solving the instances, obtaining good solutions in acceptable times.

KEYWORDS. Vehicle routing, pickup and delivery; time windows; heterogeneous fleet; multiple products.

Main Area: Logistics and Transportes

1. Introdução

Problemas de roteamento de veículos envolvendo a coleta e/ou entrega aparecem em várias aplicações práticas do cotidiano das empresas. Especificamente o problema de coleta e entrega com janelas de tempo (PCEJT), surge quando deseja-se coletar pessoas ou produtos em um cliente e posteriormente entregá-los a outro, cabendo ao veículo respeitar essa sequência de visitas e garantir que ambos os clientes pertençam a mesma rota. As situações reais envolvendo este tipo de problema são encontradas no transporte de passageiros envolvendo ônibus escolares, serviços de táxis e de pessoas idosas ou com deficiências a centros recreacionais e ambulatoriais; transporte de mercadorias; transporte marítimo de petróleo; transporte de animais vivos; entre outros [Furtado et al., 2015; Battarra et al., 2014; Cordeau e Laporte, 2007; Sigurd et al., 2004; Qu e Bard, 2013].

Há várias formulações bem como métodos de solução abordados na literatura para o PCEJT clássico que considera frota homogênea e único produto [Battarra et al., 2014; Parragh et al., 2008; Furtado et al., 2017]. Dentre as abordagens exatas podemos observar métodos do tipo *branch-and-cut* [Lu e Dessouky, 2004; Ropke et al., 2007], que se baseiam em modelos com número de restrições exponencial em relação ao número de requisições de coleta e entrega. Também existem abordagens do tipo *branch-and-price* [Ropke e Cordeau, 2009; Baldacci et al., 2011; Bettinelli et al., 2014], que usam modelos de particionamento de conjuntos com variáveis (colunas) geradas de forma iterativa pela técnica de geração de colunas. Dentre os métodos exatos, os mais efetivos são os propostos por Ropke e Cordeau [2009] e Baldacci et al. [2011], embora, para grandes instâncias, os tempos computacionais sejam elevados. Dessa forma, métodos heurísticos vêm sendo propostos por sua capacidade de prover soluções de qualidade em tempos aceitáveis [Li e Lim, 2001; Lu e Dessouky, 2006; Bent e Hentenryck, 2006; Ropke e Pisinger, 2006]. Até o momento, os métodos sugeridos por Bent e Hentenryck [2006] e Ropke e Pisinger [2006] são os mais eficientes para o PCEJT clássico, sendo que o último se mostrou superior ao primeiro. Exceto o trabalho de Bettinelli et al. [2014], que considera frota heterogênea, todos os outros citados abordam frota homogênea e um único tipo de produto.

Em relação a abordagens heurísticas para o PCEJT contendo frota heterogênea e múltiplos produtos, podemos citar os trabalhos de Qu e Bard [2013] e Ríos-Mercado et al. [2013] que abordam aplicações práticas específicas do problema, motivadas pela interação com a indústria. Ambos os artigos utilizam a meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*) [Feo e Resende, 1995]. O primeiro trabalho difere do segundo por ter desenvolvido um algoritmo em duas fases com a segunda fase de melhoria baseada na meta-heurística proposta por Ropke e Pisinger [2006]. Qu e Bard [2013] obtiveram soluções de 30 a 40% melhores que as da empresa enquanto Ríos-Mercado et al. [2013] conseguiram redução de 6% nos custos da empresa comparado a atual abordagem da mesma.

O presente trabalho tem por objetivo propor abordagens de solução para PCEJT com frota heterogênea e múltiplos produtos. Para isso, são propostos dois modelos de programação inteira-mista, por meio da extensão de modelos conhecidos para a variante clássica do PCEJT com frota homogênea e único produto [Furtado et al., 2017; Ropke e Cordeau, 2009; Battarra et al., 2014]. Os modelos propostos são compactos, isto é, possuem número de variáveis e restrições que dependem polinomialmente do número de requisições de coleta e entrega e, portanto, podem ser resolvidos de forma direta por meio de softwares de otimização de propósito geral. As abordagens apresentadas foram testadas usando instâncias propostas por Ropke et al. [2007], modificadas para incluir frota heterogênea e múltiplos produtos.

O restante deste artigo está dividido em: definição do problema (Seção 2), proposta dos modelos de programação inteira-mista (Seção 3), descrição da meta-heurística ALNS (*Adaptive Large Neighborhood Search*) proposta (Seção 4), apresentação dos testes computacionais (Seção 5) e conclusão (Seção 6).

2. Definição do problema

Seja n o número de requisições de coleta e entrega dos clientes. O PCEJT pode ser descrito por meio de um grafo completo $G = (N, A)$, definido pelo conjunto de nós N e arcos A do problema, sendo $N = \{0, 1, \dots, 2n + 1\}$. Os nós do depósito inicial e final são representados por 0 e $2n + 1$, respectivamente. Embora haja um único depósito, essa diferenciação de índices é conveniente para a modelagem do problema. Os nós de coleta e entrega são representados pelos sub-conjuntos $P = \{1, \dots, n\} \subset N$ e $D = \{n + 1, \dots, 2n\} \subset N$ nessa ordem. Considera-se também o conjunto $L = \{1, \dots, l\}$ de todos os produtos e o conjunto $K = \{1, \dots, m\}$ de todos os veículos disponíveis com diferentes capacidades q_k . Todo nó $i \in N$ tem uma demanda $d_{il} \forall l \in L$ e um tempo de serviço s_i , onde $d_{0l} = d_{2n+1,l} = 0 \forall l \in L$ e $s_0 = s_{2n+1} = 0$. Cada requisição de coleta $i \in P$ está associada com um nó de entrega $n + i \in D$, de tal forma $d_{n+i,l} = -d_{il} \forall l \in L$. A cada produto $l \in L$ existe um parâmetro associado u_l que é o fator multiplicador para conversão da unidade de medida do produto para a capacidade dos veículos. Por exemplo, se uma unidade do produto l pesa 1kg e a capacidade dos veículos da frota é dada em toneladas, então o parâmetro u_l tem valor $\frac{1}{1000} = 0,001$. Existe para cada nó $i \in N$ uma janela de tempo $[w_i^a, w_i^b]$, tal que w_i^a e w_i^b representam o início mais cedo e mais tarde que um serviço pode ocorrer. Por fim tem-se a cada arco $(i, j) \in A$ o custo de roteamento c_{ij} e o tempo de deslocamento t_{ij} . Assume-se que a desigualdade triangular é válida para os custos de roteamento e tempos de deslocamento. Note que cada nó de coleta é pareado com um único de entrega (e vice-versa) e tudo que é coletado em um, deve ser entregue no outro, pelo mesmo veículo, não necessariamente de forma imediata (isto é, o veículo pode visitar outros nós entre uma coleta e sua respectiva entrega).

3. Modelos de programação inteira-mista

Na formulação de três-índices do PCEJT clássico [Battarra et al., 2014; Ropke e Cordeau, 2009] existe a variável binária x_{ijk} definida para cada arco $(i, j) \in A$ e para cada veículo $k \in K$, de forma que $x_{ijk} = 1$ se e somente se o veículo k visita o nó i e vai diretamente para j . É possível utilizar essa formulação considerando frota heterogênea e estendê-la para incorporar múltiplos produtos. Para isso, define-se: uma variável inteira positiva y_{ikl} que indica a carga do produto l no veículo k após deixar o nó i , para todo $i \in N$, para cada $k \in K$ e para cada $l \in L$; e uma variável contínua w_{ik} que indica o tempo de começo do serviço no nó i para cada $i \in N$ e $k \in K$. A seguir, a formulação de três-índices para o PCEJT com frota heterogênea e múltiplos produtos é apresentada.

$$\min \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk} \quad (1a)$$

$$\text{s.a} \quad \sum_{k \in K} \sum_{j \in N} x_{ijk} = 1 \quad \forall i \in P, \quad (1b)$$

$$\sum_{j \in N} x_{ijk} - \sum_{j \in N} x_{n+i,j,k} = 0 \quad \forall k \in K; i \in P, \quad (1c)$$

$$\sum_{j \in N} x_{0jk} \leq 1 \quad k \in K, \quad (1d)$$

$$\sum_{j \in N} x_{0jk} - \sum_{i \in N} x_{i,2n+1,k} = 0 \quad \forall k \in K, \quad (1e)$$

$$\sum_{j \in N} x_{jik} - \sum_{j \in N} x_{ijk} = 0 \quad \forall k \in K; i \in P \cup D, \quad (1f)$$

$$w_{jk} \geq w_{ik} + (s_i + t_{ij})x_{ijk} - M(1 - x_{ijk}) \quad \forall k \in K; i, j \in N, \quad (1g)$$

$$\begin{aligned}
w_{ik} + s_i + t_{i,n+i} &\leq w_{n+i,k} & \forall k \in K; i \in P \cup D, & (1h) \\
w_i^a &\leq w_{ik} \leq w_i^b & \forall k \in K; i \in P \cup D, & (1i) \\
y_{jkl} &\geq y_{ikl} + d_{jl} - M(1 - x_{ijk}) & \forall k \in K; i \in N; j \in N; l \in L, & (1j) \\
\sum_{l \in L} y_{ikl} \cdot u_l &\leq q_k & \forall k \in K; i \in N; l \in L, & (1k) \\
x_{ijk} &\in \{0, 1\} & \forall k \in K; i, j \in N. & (1l)
\end{aligned}$$

A função objetivo representada por (1a) consiste em minimizar os custos de roteamento. As restrições (1b) garantem que os nós de coleta e entrega sejam visitados uma única vez. A paridade das visitas entre os nós de coleta e entrega é assegurada pelas restrições (1c). As restrições (1d) e (1e) restringem a saída dos veículos do depósito inicial e asseguram que todo veículo que deixa o mesmo deve voltar para o depósito final. O fluxo entre os nós do problema é dado pelas restrições (1f). As restrições (1g) garantem a progressividade dos tempos de visita enquanto as restrições (1h) asseguram a precedência entre as visitas. A satisfação as janelas de tempo é garantida pelas restrições (1i) e juntamente com (1g) previnem a geração de sub-rotas. O fluxo da carga dentro dos veículos após as visitas é dado pelas restrições (1j), a observação ao limite de capacidade dos veículos é feito pelas restrições (1k). A integralidade das variáveis de decisão é imposta pelas restrições (1l).

Uma formulação de dois-índices também pode ser utilizada para modelar o PCEJT com frota heterogênea e múltiplos produtos. Para isso, estende-se a formulação de Furtado et al. [2017], originalmente proposta para o PCEJT clássico. A formulação proposta pelos autores permite garantir a paridade e precedência entre as visitas aos pares de coleta e entrega por meio da inserção de novas variáveis contínuas v_i para todo nó $i \in P \cup D$. O papel dessas variáveis é armazenar o índice do primeiro nó de uma rota r e, por meio de restrições adicionais, garantir a propagação desse índice para as demais variáveis v_j tal que $j \in r$. A escolha desta formulação foi motivada pelo seu desempenho superior frente ao modelo clássico de literatura [Battarra et al., 2014; Ropke e Cordeau, 2009] e do apresentado por Lu e Dessouky [2004] para o PCJET clássico. Para mais informações sobre essa nova estratégia de formulação remete-se ao trabalho de Furtado et al. [2017].

Na formulação de dois-índices, as variáveis x_{ij} , y_{il} e w_i desempenham papel semelhante ao das variáveis x_{ijk} , y_{ikl} e w_{ik} da formulação (1) apresentada. A única diferença é ausência do índice k que corresponde ao veículo.

Como não há um índice representando os veículos, a construção da frota heterogênea dá-se pela inserção de nós artificiais de coleta e entrega [Ropke et al., 2007]. Seja $m = |K|$, onde cada veículo da frota tenha capacidade $q_1, \dots, q_m$ e seja $q = \max_{1 \leq i \leq m} \{q_i\}$. São criados dois conjuntos, P_a e D_a , contendo m requisições artificiais de coleta e de entrega, respectivamente. Com isso, os conjuntos de nós de coleta e entrega originais são redefinidos, de modo que $P_a \subset P$ e $D_a \subset D$. Contudo, ainda é necessário inserir um produto artificial l_a que tenha a mesma unidade de medida da capacidade q , e com o parâmetro $u_{l_a} = 1$. Assim é possível impor para cada requisição i artificial uma demanda associada ao produto l_a , tal que $d_{i,l_a} = q - q_i$ e $d_{n+i,l_a} = -d_{i,l_a}$. A demanda para todos os produtos reais dos respectivos nós artificiais deve ser 0, assim como para todos os nós reais a demanda pelo produto artificial também deve ser 0.

É necessário garantir que todos os arcos que deixam o depósito inicial e vão para os nós de coleta artificiais tenham custos e tempos de viagem iguais a zero. O mesmo deve ser feito para os custos e tempos de viagem para todos os arcos que ligam os nós de entrega artificiais ao depósito final. Arcos entre os nós artificiais, também devem ter custos e tempos de viagem iguais a zero. Os arcos entre os nós de coleta artificiais e os reais j devem ter custos c_{0j} e tempos de viagem t_{0j} . Enquanto os custos e tempos de viagem dos nós de entrega reais $n + j$ para os de entrega artificiais devem corresponder a $c_{n+j,2n+1}$ e $t_{n+j,2n+1}$ [Ropke et al., 2007]. A Figura 1 demonstra com mais clareza as imposições mencionadas anteriormente. Nesta figura, k e $n + k$ são os nós artificiais de

coleta e entrega, enquanto os demais são nós reais. A formulação de dois-índices para o PCEJT com frota heterogênea e múltiplos produtos é apresentada a seguir:

$$\min \quad \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \quad (2a)$$

$$\text{s.a} \quad \sum_{i \in N} x_{ij} = 1 \quad \forall j \in P \cup D, \quad (2b)$$

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in P \cup D, \quad (2c)$$

$$w_j \geq w_i + t_{ij} - M(1 - x_{ij}) \quad \forall i \in N; j \in N, \quad (2d)$$

$$w_i^a \leq w_i \leq w_i^b \quad \forall i \in N, \quad (2e)$$

$$y_{jl} \geq y_{il} + d_{jl} - M(1 - x_{ij}) \quad \forall i \in N; j \in N; l \in L, \quad (2f)$$

$$\sum_{l \in L} y_{il} \cdot u_l \leq q \quad \forall i \in N, \quad (2g)$$

$$w_{n+i} \geq w_i + s_{ij} + t_{i,i+n} \quad \forall i \in P, \quad (2h)$$

$$v_{n+i} = v_i \quad \forall i \in P, \quad (2i)$$

$$v_j \geq j \cdot x_{0j} \quad \forall j \in P \cup D, \quad (2j)$$

$$v_j \leq j \cdot x_{0j} - n(x_{0j} - 1) \quad \forall j \in P \cup D, \quad (2k)$$

$$v_j \geq v_i + n(x_{ij} - 1) \quad \forall i, j \in P \cup D, \quad (2l)$$

$$v_j \leq v_i + n(1 - x_{ij}) \quad \forall i, j \in P \cup D, \quad (2m)$$

$$x_{0i} = 0 \quad \forall i \in N \setminus P_a, \quad (2n)$$

$$x_{ij} = 0 \quad \forall i \in P_a; j \in N \setminus P \cup D_a, \quad (2o)$$

$$x_{ij} = 0 \quad \forall i \in N \setminus D \cup P_a; j \in D_a, \quad (2p)$$

$$x_{i,2n+1} = 0 \quad \forall i \in N \setminus D_a. \quad (2q)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in N; j \in N \quad (2r)$$

A função objetivo (2a) consiste em minimizar os custos de roteamento. As restrições (2b) e (2c) asseguram que cada cliente deve ser visitado uma única vez. As restrições (2d) garante a progressão dos tempos de visita e, juntamente com as restrições (2e) asseguram que não haja sub-rotas. O fluxo da carga dos veículos após a visita aos clientes é garantida pelas restrições (2f). A não violação da capacidade dos veículos é imposta pelas restrições (2g). A paridade e precedência das visitas são asseguradas pelas restrições de (2h) à (2m). As restrições (2n) garantem que o depósito inicial só tenha ligações com os nós artificiais de coleta. As restrições (2o) determinam que os nós de coleta artificiais somente tenha ligações com os nós reais de coleta e com os artificiais de entrega. Restrições (2p) garantem que os nós artificiais de entrega só possam ser alcançados por meio dos nós de entrega reais e dos de coleta artificiais. Restrições do tipo (2q) garantem que o depósito final só tenha ligações com os nós de entrega artificiais. Por fim, as restrições (2r) determinam a integralidade das variáveis de decisão.

4. Meta-heurística ALNS

A meta-heurística ALNS proposta é baseada no trabalho de Ropke e Pisinger [2006] que, por sua vez, inspiraram-se em Shaw [1998] Bent e Hentenryck [2006]. A implementação realizada por Ropke e Pisinger [2006] usa um algoritmo de dois estágios. O primeiro estágio busca reduzir a frota homogênea de veículos enquanto o segundo estágio fica encarregado de minimizar as distâncias percorridas. Por vezes o segundo estágio é capaz de reduzir veículos também [Bent e

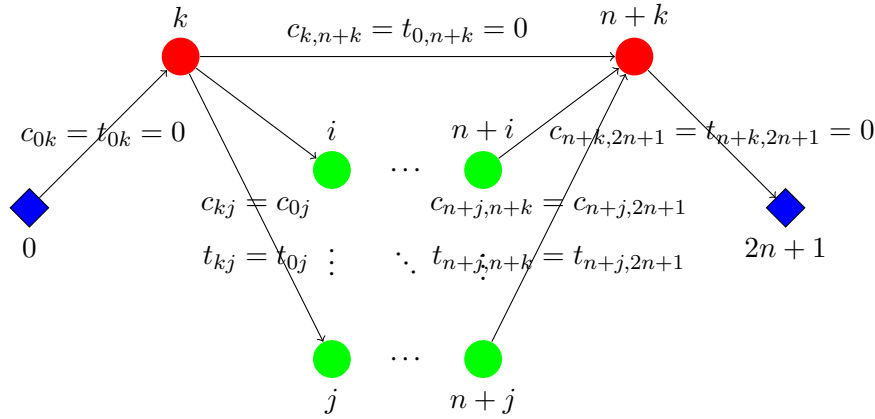


Figura 1: Nova estrutura do grafo contendo nós artificiais

Hentenryck, 2006]. Ambos os estágios utilizam a ALNS, ou seja, uma única meta-heurística. No trabalho de Bent e Hentenryck [2006] existem duas meta-heurísticas: a primeira uma *simulated annealing* (SA), é usada para minimizar a frota; a segunda é uma *large neighborhood search* (LNS) para minimizar as distâncias percorridas na segunda fase. Note que ambos os trabalhos consideram frota homogênea e um único tipo de produto.

A implementação proposta neste trabalho não considera dois estágios. Ropke e Pisinger [2006] mencionam que o algoritmo com duas fases é incapaz de lidar com frota heterogênea. Diante desse fato, optou-se por uma implementação contendo uma única fase, mas com uma heurística específica para redução de veículos baseada em Qu e Bard [2012, 2013] com frota heterogênea e múltiplos produtos.

Seja S uma solução para o PCEJT. Considere também o conjunto R de rotas factíveis r da solução S , e a variável I_r que assume valor 0 caso a rota seja vazia (a rota não contém nenhum nó de coleta e entrega) e valor 1 caso contrário. Seja U o conjunto de todas as requisições i ainda não roteadas e U_i a variável que recebe valor 1 se a requisição i não se encontra na solução S , ou 0 caso contrário. Aqui assumi-se que t_{ij} é equivalente à distância entre os nós i e j . Dessa forma, pode-se expressar a função objetivo usada na ALNS por:

$$c(s) = w_1 \sum_{r \in R} \sum_{(i,j) \in r} t_{ij} + w_2 \sum_{r \in R} I_r + w_3 \sum_{i \in U} U_i \quad (3)$$

Considera-se a ordem dos pesos ($w_1 \ll w_2 \ll w_3$), fazendo com que a função objetivo minimize primeiramente a quantidade de requisições não roteadas, depois o número de veículos e por fim as distâncias viajadas.

A cada iteração, a meta-heurística ALNS remove μ requisições de S reinserindo-as novamente na expectativa de gerar uma outra solução S_* com menor custo. O algoritmo é executado por um número finito de iterações ou até que o tempo limite seja atingido. O Algoritmo 1 demonstra o processo da ALNS. Para a construção da solução inicial, utiliza-se uma heurística de inserção que seleciona requisições ao acaso e as insere na melhor posição possível dentre todas as rotas. Os veículos alocados às rotas são ordenados em uma lista decrescente de capacidade e selecionados nesta ordem.

Algoritmo 1: ALNS

Entrada: $S, \mu \in \mathbb{N}, \maxIterations \in \mathbb{N}, timeToSpend \in \mathbb{N}$;

Saída: S_{best} ;

$countIt = 0, S_{best} = S$;

repita

$S_* = S$;

se $countIt = 100$ **então**

 atualize os pesos das heurísticas;

$countIt = 0$;

fim

 selecione uma heurística de remoção h_r e inserção h_i ;

 remova μ requisições de S_* utilizando h_r ;

 insira μ requisições em S_* utilizando h_i ;

se $S_* \leq S_{best}$ **então**

$S_{best} = S_*$;

fim

se $aceito(S_*, S)$ **então**

$S = S_*$;

fim

$countIt = countIt + 1$;

até $\maxIterations \geq 0$ E $timeToSpend > 0$;

O critério de aceite é o da meta-heurística SA, dado por $e^{-\frac{S_* - S}{T}}$, sendo T a temperatura inicial obtida como em Ropke e Pisinger [2006]. Esse critério é usado para tentar escapar de ótimos locais, juntamente com a adição de um ruído, *noise*, na função objetivo. O *noise* não é implementado em todas as heurística de inserção. Dessa forma ele é selecionado de acordo com o critério adaptativo da ALNS explicado à frente. O *noise* é selecionado aleatoriamente, dentro de um intervalo $[-\max R, \max R]$ posteriormente utilizado para calcular a nova função objetivo $c' = \max\{0, c + \text{noise}\}$. Os parâmetros T e $\max R$ são calculados conforme o proposto por Ropke e Pisinger [2006].

O termo *adptative* vem do processo de coleta de informações de desempenho das heurísticas de inserção e remoção em iterações anteriores. Assumindo que temos k heurísticas de inserção e remoção, cada uma tem um peso W_i , $i \in \{1, 2, \dots, k\}$, uma heurística j é selecionada com probabilidade:

$$\frac{W_j}{\sum_{i=1}^k W_i}. \quad (4)$$

Primeiramente, um segmento é um número pré-estabelecido de iterações. Assim como em Ropke e Pisinger [2006], neste trabalho um segmento é definido por 100 iterações. A busca é dividida em segmentos e os pesos das heurísticas atualizados a cada um. Denotando por W_{ij} o peso da heurística i no segmento j , tem-se que o novo peso é calculado por:

$$W_{i,j+1} = (1 - r)W_{ij} + r \frac{\pi_i}{\theta_i}, \quad (5)$$

sendo π_i a pontuação calculada no último segmento para uma heurística i ; θ_i é o número de vezes que a heurística i foi usada no último segmento; r é um fator de reação: se $r = 0$, então somente os pesos iniciais são usados para a seleção das heurísticas, enquanto para $r = 1$ passa-se a dar importância somente à pontuação para a decisão de qual heurística selecionar. A cada segmento somente duas heurísticas são pontuadas e usadas, uma de remoção outra de inserção. As heurísticas são pontuadas por σ_1 , caso encontrem uma solução melhor que a global; σ_2 caso encontrem uma

solução melhor que a atual; e σ_3 caso encontrem uma solução pior do que a atual. Os dois últimos casos só são pontuados caso as soluções passem no critério de aceite da SA. Somente soluções ainda não encontradas são pontuadas.

4.1. Heurística de remoção

Nossa implementação considera oito heurísticas de remoção baseadas em Ropke e Pisinger [2006]; Qu e Bard [2012]; Ghilas et al. [2016]. Todas removem μ requisições, sendo μ selecionado aleatoriamente dentro do intervalo de $[0,1n; 0,4n]$.

Primeiramente, tem-se a *Shaw removal* que remove requisições parecidas entre elas. Um fator de semelhança R_{ij} é calculado por:

$$R_{ij} = \varphi(t_{ij} + t_{n+i,n+j}) + \chi(|w_i - w_j| + |w_{n+i} - w_{n+j}|) + \psi|d_i - d_j|. \quad (6)$$

As requisições são colocadas em uma lista ordenada OL , de forma ascendente por R_{ij} . Um número $p \in (0,1)$ é selecionado aleatoriamente e usando um parâmetro $\alpha \geq 1$ seleciona-se uma requisição $OL[p^\alpha \cdot |OL|]$ para ser removida.

As heurísticas de remoção *Time related removal*, *Distance related removal* e *Demand related removal* são todos casos especiais da *Shaw removal* com pesos $\chi = 1, \varphi = 1$ e $\psi = 1$, respectivamente, enquanto os outros pesos são 0.

Utiliza-se também a *Random route removal*, que remove a cada iteração todas as requisições de uma rota deixando-a vazia. Ao se fazer isso aumenta-se a chance de encontrar soluções com um número de veículos menor. Caso não tenha atingido o número de requisições μ a serem removidas, a heurística *Shaw removal* é usada para completá-lo. A *Random route removal* remove aleatoriamente μ requisições da solução. *Worst removal* remove a requisição que mais piora a função objetivo. Por fim, tem-se a *Late removal* que remove a requisição que tenha a maior diferença entre o tempo de início de serviço e a abertura da janela de tempo nos nós de coleta e entrega.

4.2. Heurística de inserção

Neste trabalho, utilizam-se 10 heurísticas de inserção, todas baseadas em Ropke e Pisinger [2006], ou seja, são *regret-heuristics* (heurística de arrependimento). Esse tipo de heurística considera a diferença entre inserir uma requisição i na sua melhor posição c_{i1} e na sua k -ésima melhor posição, c_{ik} . O *regret* associado para cada requisição i é dado por:

$$\max_{i \in U} \left\{ \sum_{j=1}^k (c_{ij} - c_{i1}) \right\}. \quad (7)$$

Escolhe-se a cada iteração da heurística de inserção a requisição que tenha o maior *regret*. Dessa forma, tem-se uma família de cinco *regret-heuristics* variando-se k em $\{1, 2, 3, 4, m\}$. Outras cinco implementações incluem a adição de *noise* na função objetivo como descrito anteriormente. Para mais detalhes sobre as implementações, remete-se aos trabalhos de Ropke e Pisinger [2006] e Shaw [1997].

5. Resultados

Nesta seção são apresentados os resultados de testes computacionais com os modelos e a meta-heurística. Os testes foram realizados em um computador com processador Intel Core i7-7500U, 2.70GHz, contendo 16GB de memória RAM. O solver utilizado foi o IBM Cplex 12.7.0. A codificação dos modelos e da meta-heurística foi feita em C++. A saída dos veículos do depósito são penalizadas em 10^4 , sendo este também o valor do parâmetro w_2 da ALNS. O valor dos outros parâmetros são $w_1 = 1$ e $w_3 = 10^6$.

As classes de instâncias utilizadas (AA, BB, CC, DD) foram as propostas por Ropke et al. [2007], contendo novas instâncias criadas para cada classe de acordo com Furtado et al. [2017]. A frota heterogênea foi incluída gerando-se aleatoriamente o parâmetro q_k entre a maior demanda da classe, definida como $maxD$, e 20% maior do que $maxD$, ou seja, $q_k = [maxD; 1,2 \cdot maxD]$. Uma frota foi criada para as classes AA e CC contendo 7 veículos com capacidades $\{15, 18, 17, 15, 15, 17, 16\}$ e outra para as classes BB e DD contendo 9 veículos com capacidade $\{23, 22, 21, 20, 24, 24, 20, 21, 22\}$.

Foram gerados três produtos para os testes, todos com os parâmetros $u_l = 1 \forall l \in L$. Para geração dos produtos, a demanda de cada requisição d_i é dividida pelo número de produtos, neste caso por 3. Caso a divisão $d_i/3$ tenha resto positivo esse valor é adicionado a um dos produtos. Assim $\sum_{l \in L} d_{il} = d_i \forall i \in N$.

5.1. Resultados para os modelos

A Tabela 1 apresenta os resultados dos modelos para as instâncias utilizadas. A primeira coluna, apresenta o nome da instância e as demais colunas apresentam, para cada modelo, o valor da função objetivo (UB) para a melhor solução encontrada, gap relativo (Gap) correspondente e o tempo computacional em segundos (Tempo). São reportados apenas os resultados que tiveram o *Gap* disponível pelo Cplex, para pelo menos um dos modelos. O *Gap*, é calculado pela diferença entre o valor da função objetivo e um limitante inferior, dividindo-se essa diferença pelo valor da função objetivo. Para todas as outras instâncias não foram encontradas soluções dentro do limite estabelecido de uma hora. Caso os tempos das soluções obtidas tenha atingido o máximo permitido, reporta-se apenas como TL. É possível observar que o modelo de dois-índices é superior ao de três-índices, encontrando soluções para mais instâncias, para todas as classes de problemas. Esses resultados corroboram com os reportados por Furtado et al. [2017] sobre a eficiência obtida com o uso desta formulação frente a outras da literatura.

Instâncias	Modelo três-índices			Modelo dois-índices		
	UB	Gap	Tempo (s)	UB	Gap	Tempo (s)
AA5	10184,8	0,00	0,17	10184,8	0,00	0,08
AA10	10383,6	0,00	2,70	10383,6	0,00	0,41
AA15	20542,4	0,00	TL	20542,4	0,00	11,72
AA20	20762,9	0,01	TL	20757,9	0,00	TL
AA25	-	-	-	30976,2	0,33	TL
BB5	10339,7	0,00	0,13	10339,7	0,00	0,08
BB10	20512,5	0,00	25,47	20512,5	0,00	0,92
BB15	20667,6	0,00	TL	20667,6	0,00	15,86
BB20	-	-	-	20786	0,00	106,52
BB25	-	-	-	20946,1	0,01	TL
CC5	10212,5	0,00	5,61	10212,5	0,00	0,22
CC10	10394,3	0,01	TL	10394,3	0,00	6,49
CC15	-	-	-	20557,8	0,49	TL
CC20	-	-	-	20748,9	0,49	TL
DD5	10324,8	0,00	0,28	10324,8	0,00	0,09
DD10	20567,8	0,49	TL	20567,8	0,00	41,49
DD15	-	-	-	20616,4	0,49	TL
DD20	-	-	-	20679,2	0,49	TL

Tabela 1: Resultados para os modelos de três- e dois-índices

5.2. Resultados para a meta-heurística

Cada instância foi executada cinco vezes na ALNS, com um limite de tempo de 10 minutos e 25000 iterações. Os valores para os parâmetros $\sigma_1, \sigma_2, \sigma_3, \varphi, \chi, \psi, \alpha$, e r , são respectivamente, 27; 30; 22; 0,3; 0,4; 0,3; 6 e 0,1. Todos os parâmetros foram calibrados de forma semelhante como

apresentado por Ropke e Pisinger [2006]. O peso para todas as heurísticas foi estabelecido inicialmente como um.

De acordo com a Tabela 2 é possível identificar que a ALNS obteve melhores resultados em tempos iguais ou menores que os dois modelos propostos, como esperado. Para o cálculo dos desvios padrões das soluções, utilizou-se os cinco resultados obtidos durante os testes para cada instância. Estes resultados indicam a estabilidade da ALNS para as soluções obtidas a cada execução. Há casos, como nas instâncias DD75 e BB75 em que os desvios assumiram grandes valores devido a soluções com menor número de veículos no cálculo da média.

6. Conclusão

O presente trabalho propôs dois modelos para o PCEJT com frota heterogênea e múltiplos produtos, baseados nas formulações de dois- e três-índices. Os modelos se mostraram adequados para resolução de instâncias de pequeno porte, sendo que o modelo com dois-índices é superior ao de três. Uma meta-heurística ALNS foi proposta para o mesmo problema e obteve resultados superiores aos encontrados pelos modelos, conseguindo encontrar boas soluções em tempos aceitáveis para problemas com 150 nós de coleta e entrega. Para os próximos passos pretende-se incorporar coleta/entrega fracionada ao problema, concebendo uma formulação compacta para o problema, e uma meta-heurística ALNS capaz de lidar com essas restrições práticas. Para a meta-heurística, mais testes serão realizados com a intenção de melhorar os desvios padrões da média das soluções e também tentar paralelizar seu processamento para assim alcançar a redução dos tempos computacionais. Ao fim, deseja-se obter um método *branch-and-cut* para problema PCEJT com frota heterogênea, múltiplos produtos e coleta/entrega fracionada.

7. Agradecimentos

Os autores agradecem o apoio financeiro da FAPESP (Projeto 2016/24763-1) e da CAPES e aos revisores anônimos, por comentários importantes que contribuíram para a melhoria deste trabalho.

Referências

- Baldacci, R., Bartolini, E., e Mingozzi, A. (2011). An exact algorithm for the pickup and delivery problem with time windows. *Operations research*, 59(2):414–426.
- Battarra, M., Cordeau, J.-F., e Iori, M. (2014). Pickup-and-delivery problems for goods transportation. In Toth, P. e Vigo, D., editors, *Vehicle routing: Problems, methods, and applications*, MOS/SIAM Ser Optim, p. 161–191.
- Bent, R. e Hentenryck, P. V. (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers and Operations Research*, 33(4):875–893. ISSN 03050548.
- Bettinelli, A., Ceselli, A., e Righini, G. (2014). A branch-and-price algorithm for the multi-depot heterogeneous-fleet pickup and delivery problem with soft time windows. *Mathematical Programming Computation*, 6(2):171–197.
- Cordeau, J.-F. e Laporte, G. (2007). The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153(1):29–46.
- Feo, T. A. e Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133.
- Furtado, M. G. S., Munari, P., e Morabito, R. (2015). O problema de coleta e entrega na indústria petrolífera: modelagem e método de solução branch-and-cut. In *Anais do XLVII Simpósio Brasileiro de Pesquisa Operacional*, p. 1581–1591. SOBRAPO.

Instâncias	Média UB	Desvio Padrão UB	Média Tempos (s)	Melhor UB	Tempo Melhor Solução (s)
AA5	10184,80	0,00	0,00	10184,80	0
AA10	10383,60	0,00	0,00	10383,60	0
AA15	20542,40	0,00	1,80	20542,40	0
AA20	20757,90	0,00	17,20	20757,90	0
AA25	21041,08	0,11	4,20	21041,00	1
AA30	31117,20	0,70	126,00	31116,00	136
AA35	31279,10	2,94	286,20	31277,10	182
AA40	31448,90	0,27	272,60	31448,70	195
AA45	31673,46	4,82	404,20	31666,60	445
AA50	41770,50	34,62	475,80	41733,80	487
AA55	41939,18	9,15	540,20	41930,90	436
AA60	42133,80	18,63	410,60	42108,40	599
AA65	42268,96	19,75	312,60	42238,60	449
AA70	52491,64	28,00	405,00	52448,50	150
AA75	52625,50	30,37	341,20	52575,10	495
BB5	10339,70	0,00	0,00	10339,70	0
BB10	20512,50	0,00	0,00	20512,50	0
BB15	20667,60	0,00	0,20	20667,60	0
BB20	20786,00	0,00	1,80	20786,00	0
BB25	20947,40	2,91	6,40	20946,10	0
BB30	31038,14	2,05	127,20	31036,70	127
BB35	31228,68	5,75	137,00	31225,00	174
BB40	31449,06	2,01	177,00	31447,30	46
BB45	41535,02	15,77	447,40	41522,40	450
BB50	41681,20	39,79	511,20	41652,60	513
BB55	41866,54	12,81	462,00	41853,70	555
BB60	62361,98	12,86	579,40	62344,20	583
BB65	62673,42	32,14	498,60	62635,20	596
BB70	62987,24	49,00	356,40	62920,80	544
BB75	71104,30	4513,17	280,00	63031,10	175
CC5	10212,50	0,00	0,00	10212,50	0
CC10	10394,30	0,00	0,00	10394,30	0
CC15	20554,90	0,00	0,80	20554,90	0
CC20	20721,40	0,00	20,40	20721,40	2
CC25	20908,80	0,00	64,80	20908,80	38
CC30	31036,02	16,69	94,00	31010,20	93
CC35	31168,04	35,82	270,20	31105,00	576
CC40	31283,42	7,32	294,80	31276,40	335
CC45	31430,64	15,24	351,00	31419,70	423
CC50	41671,38	58,48	492,80	41628,80	505
CC55	41771,42	15,74	551,00	41745,00	523
CC60	42012,26	14,17	423,80	41997,80	438
CC65	52196,38	34,93	452,60	52147,60	562
CC70	52382,60	39,00	433,40	52353,60	534
CC75	52564,38	24,97	364,00	52519,80	369
DD5	10324,80	0,00	0,00	10324,80	0
DD10	20567,80	0,00	0,00	20567,80	0
DD15	20616,40	0,00	0,00	20616,40	0
DD20	20679,20	0,00	15,80	20679,20	11
DD25	20902,84	1,21	84,00	20902,30	66
DD30	21016,68	1,56	113,60	21015,10	65
DD35	31095,08	7,79	209,40	31083,50	201
DD40	31209,54	5,50	260,60	31202,20	280
DD45	31326,74	10,67	413,20	31310,90	442
DD50	31444,38	12,79	508,80	31430,20	536
DD55	31620,22	22,72	484,20	31581,60	515
DD60	41794,32	26,24	549,00	41762,30	513
DD65	42145,56	15,72	488,60	42129,80	442
DD70	42249,50	20,02	330,20	42228,20	446
DD75	50446,74	4463,62	485,60	42462,70	478

Tabela 2: Resultados para a ALNS

- Furtado, M. G. S., Munari, P., e Morabito, R. (2017). Pickup and delivery problem with time windows: A new compact two-index formulation. *Operations Research Letters*, 45(4):334–341. ISSN 0167-6377.
- Ghilas, V., Demir, E., e Van Woensel, T. (2016). An adaptive large neighborhood search heuristic for the Pickup and Delivery Problem with Time Windows and Scheduled Lines. *Computers and Operations Research*, 72:12–30.
- Li, H. e Lim, A. (2001). A metaheuristic for the pickup and delivery problem with time windows. In *Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001*, p. 160–167. IEEE Comput. Soc. ISBN 0-7695-1417-0.
- Lu, Q. e Dessouky, M. (2004). An exact algorithm for the multiple vehicle pickup and delivery problem. *Transportation Science*, 38(4):503–514.
- 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(2):672–687.
- Parragh, S. N., Doerner, K. F., e Hartl, R. F. (2008). A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58 (2):81–117. ISSN 0344-9327.
- Qu, Y. e Bard, J. F. (2012). A grasp with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers and Operations Research*, 39(10):2439–2456. ISSN 03050548.
- Qu, Y. e Bard, J. F. (2013). The heterogeneous pickup and delivery problem with configurable vehicle capacity. *Transportation Research Part C: Emerging Technologies*, 32:1–20. ISSN 0968-090X.
- Ríos-Mercado, R. Z., López-Pérez, J. F., e Castrillón-Escobar, A. (2013). A grasp for a multi-depot multi-commodity pickup and delivery problem with time windows and heterogeneous fleet in the bottled beverage industry. In Pacino, D., Voß, S., e Jensen, R. M., editors, *Computational Logistics: 4th International Conference, ICCL 2013, Copenhagen, Denmark, September 25-27, 2013. Proceedings*, p. 143–157. Springer Berlin Heidelberg. ISBN 978-3-642-41019-2.
- 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.
- Ropke, S., Cordeau, J.-F., e Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, 49(4):258–272.
- Ropke, S. e Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472.
- Shaw, P. (1997). A new local search algorithm providing high quality solutions to vehicle routing problems. *APES Group, Dept of Computer Science, University of Strathclyde*, p. 1–12.
- Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. *Computer*, 1520(Springer):417–431. ISSN 03029743.
- Sigurd, M., Pisinger, D., e Sig, M. (2004). Scheduling transportation of live animals to avoid the spread of diseases. *Transportation Science*, 38(2):197–209.