São José dos Campos - SP
LV SBPO
LV Simpósio Brasileiro de Pesquisa Operacional
São José dos Campos, SP - 6 a 9 de novembro de 2023

O problema de coleta e entrega com janelas de tempo e escalonamento nas arestas

Vítor A. Barbosa
Instituto de Computação, Universidade Federal da Bahia
Salvador, BA, Brasil
vitor.barbosa@ufba.br

Sunil Tiwari
Department of Operations Management and Decision Science, ESSCA School of Management
Lyon, França
sunil.tiwari@essca.fr

Rafael A. Melo
Instituto de Computação, Universidade Federal da Bahia
Salvador, BA, Brasil
rafael.melo@ufba.br

RESUMO

Neste trabalho, é introduzido e formalizado o problema de coleta e entrega com janelas de tempo e escalonamento nas arestas (PDPTW-SE, do inglês *pickup and delivery problem with time windows and scheduling on the edges*). Neste problema, deve-se realizar a coleta e entrega de um conjunto de pedidos (cada um com os seus pontos de coleta e entrega) utilizando uma frota heterogênea de veículos, sendo que algumas viagens incluem o uso de máquinas, cujos escalonamentos devem ser estabelecidos. O objetivo consiste em minimizar o tempo de conclusão total. É proposta uma formulação de programação inteira mista compacta baseada em arcos para o problema. Além disso, uma heurística gulosa é apresentada para geração de soluções viáveis, a qual também é usada no processo de geração de instâncias para o problema. Experimentos computacionais preliminares mostram que a formulação resolveu de forma ótima 44 das 72 instâncias testadas com 6, 8 e 10 pedidos, sendo que 29 dessas foram obtidas em instâncias com horizonte de planejamento mais curto. Além disso, a formulação obteve resultados em média 30% melhores que a heurística gulosa. Todo o trabalho foi desenvolvido pelo aluno sob supervisão do orientador.

PALAVRAS CHAVE. Roteamento de veículos. Escalonamento de máquinas. Programação inteira.

Tópicos (Logística e Transportes, Otimização Combinatória)

ABSTRACT

In this work, the pickup and delivery problem with time windows and scheduling on the edges (PDPTW-SE) is introduced and formalized. In this problem, the pickup and delivery

of a set of requests (each with its associated pickup and delivery points) are performed using a heterogeneous fleet of vehicles, with some trips involving machines whose schedules need to be established. The aim is to minimize the total completion time. An arc-based compact mixed integer programming formulation is proposed for the problem. Additionally, a greedy heuristic is presented for generating feasible solutions, which is also used in the instance generation process. Preliminary computational experiments show that the formulation optimally solved 44 out of the 72 instances tested with 6, 8, and 10 orders, and 29 of these were obtained in instances with a shorter planning horizon. Moreover, the formulation achieved solutions that are on average 30% better than those of the greedy heuristic. All the work was developed by the student under the advisor's supervision.

KEYWORDS. Vehicle routing. Machine scheduling. Integer programming.

Paper topics (Logistics and Transportation, Combinatorial Optimization)

1. Introdução

Este trabalho introduz uma generalização do conhecido problema de coleta e entregas com janelas de tempo (PDPTW, do inglês *pickup and delivery problem with time windows*). Nessa variante, denominada problema de coleta e entregas com janelas de tempo e escalonamento nas arestas (PDPTW-SE, do inglês *pickup and delivery problem with time windows and scheduling on the edges*), um conjunto de pedidos deve ser coletado e entregue utilizando uma frota heterogênea de veículos. Além das restrições relacionadas do PDPTW, no PDPTW-SE há um conjunto de arestas cuja travessia envolve o uso de máquinas, as quais necessitam ser escalonadas. O PDPTW-SE é um problema desafiador computacionalmente, uma vez que generaliza o PDPTW, que já foi demonstrado ser NP-difícil [Dumas et al., 1991]. O PDPTW-SE tem aplicabilidade em diferentes situações. Um exemplo é o planejamento de coletas e entregas, no qual os pedidos estão dispostos em ilhas e faz-se necessário a utilização de uma barca para realizar a travessia do veículo terrestre entre as ilhas. Outro exemplo é o planejamento de coletas e entregas por veículos guiados automaticamente na área da saúde, onde os pedidos estão situados em diferentes andares de um hospital e elevadores são utilizados para a travessia entre os andares.

O problema de coleta e entrega (PDP, do inglês *pick-up and delivery problem*) vem sendo estudado há décadas na literatura [Savelsbergh e Sol, 1995]. O PDP também é encontrado como o problema de dial-a-ride (DARP, do inglês *dial-a-ride problem*) quando o transporte de pessoas é considerado [Bodin e Berman, 1979; Cullen et al., 1981]. O PDP com janelas de tempo (PDPTW, do inglês *PDP with time windows*) vem sendo estudado amplamente na literatura com suas extensões [Dumas et al., 1991; Ropke e Pisinger, 2006; Ropke e Cordeau, 2009; Dahle et al., 2019]. Revisões da literatura sobre problemas de coleta e entrega podem ser encontradas em Berbeglia et al. [2007] e Parragh et al. [2008].

Variantes de problemas de coleta e entrega, bem como de outros problemas de roteamento, têm considerado aspectos de diferentes possibilidades de travessia das arestas ou integração de roteamento e escalonamento. Exemplos incluem o PDPTW com transporte multi-modal [Ghilas et al., 2018], bem como problemas de roteamento e escalonamento de serviços de atendimento domiciliar [Fikar e Hirsch, 2017].

Neste trabalho, introduzimos e formalizamos o PDPTW-SE, o qual ainda não foi abordado na literatura. Propomos uma formulação de programação inteira e uma heurística construtiva para o problema. Um gerador de instâncias é apresentado, o qual adapta instâncias disponíveis na literatura para o PDPTW. O restante do artigo está organizado da seguinte forma. A Seção 2 formaliza o

PDPTW-SE. A Seção 3 descreve a formulação de programação inteira. A Seção 4 apresenta a heurística construtiva. A Seção 5 sumariza os resultados computacionais preliminares. Comentários finais são discutidos na Seção 6.

2. Formalização

O problema de coleta e entrega com janelas de tempo e escalonamento nas arestas (PDPTW-SE) pode ser definido formalmente usando um grafo não direcionado $G = (V, E)$ como segue. Seja $V = V_p \cup V_d \cup \{0\}$, tal que $V_p = \{1, \ldots, n\}$ é um conjunto de nós de coleta, $V_d = \{n+1, \ldots, 2n\}$ é um conjunto de nós de entrega, e o nó 0 representa o depósito. Além disso, seja $E = E^s \cup E^m$ o conjunto de arestas, tal que E^s é o conjunto de arestas que podem ser atravessadas diretamente por um veículo e E^m é o conjunto de arestas no qual a travessia de uma aresta envolve usar uma máquina que deve ser escalonada. Seja $R = \{r_1, r_2, \ldots, r_n\}$ um conjunto de pedidos indivisíveis, cada qual denotado por uma tupla $r_i = \langle v_i, v_{n+i}, q_i \rangle$, tal que $v_i \in V_p$ denota sua origem (nó de coleta ou carregamento), $v_{n+i} \in V_d$ seu destino (nó de entrega ou descarregamento), e q_i seu peso. É assumido que $q_i \in \mathbb{Z}_{>0}$ para $i \in V_p$ e $q_i \in \mathbb{Z}_{<0}$ para $i \in V_d$, com $q_i = -q_{n+i}$ para $i \in V_p$. Cada nó $i \in V_p \cup V_d$ tem uma janela de tempo estrita $[e_i, l_i]$ na qual os veículos devem começar a operação de carregamento/descarregamento correspondente. Existe também uma janela de tempo $[e_0, l_0]$ da qual os veículos devem partir e chegar no depósito. O tempo para carregar/descarregar um veículo no nó $i \in V_p \cup V_d$ é denotado por s_i.

Existe um conjunto de veículos heterogêneos K, com cada $k \in K$ tendo uma capacidade Q_k. Além disso, atravessar a aresta $ij \in E^s$ com o veículo $k \in K$ leva tempo $\hat{d}^k_{ij}$. O tempo $\hat{d}^k_{ij}$ para uma aresta $ij \in E^m$ define um limite inferior no tempo real para atravessar a aresta, uma vez que não leva o escalonamento da máquina em consideração. É assumido que esses tempos de travessia respeitam as desigualdades triangulares. Seja F o conjunto de regiões acessíveis por veículos, no qual se um par de nós $i, j \in V$ pertence a mesma região $f \in F$, então a aresta $ij \in E^s$. Considere um conjunto de máquinas H usadas para carregar os veículos entre localizações intermediárias para transpor certos obstáculos. Além disso, seja $H_e \subseteq H$ o conjunto de máquinas que podem ser usadas na travessia da aresta $e \in E^m$. Seja $F_h \subseteq F$ o conjunto de regiões que uma máquina $h \in H$ possui uma estação. Cada máquina $h \in H$ tem no máximo uma estação em cada região $f \in F$. Note que $|F_h| \geq 2, \ \forall h \in H$, de tal forma que cada máquina atende pelo menos duas regiões. Também considere que pelo menos a máquina $0 \in H$ possui uma estação em todas as regiões e uma máquina h pode viajar entre quaisquer duas regiões $f_1, f_2 \in F_h$. É assumido que uma máquina pode apenas carregar um veículo por vez, nenhuma preempção é permitida, e uma única máquina é necessária para atravessar uma aresta $ij \in E^m$. Seja f^h_i a estação da máquina $h \in H$ correspondente ao nó i. O parâmetro f^h_i mapeia $h \in H, i \in V \rightarrow \{0, ..., |F_h| - 1\}$, ou seja, os pontos das estações da máquina $h \in H$. Se a máquina não possui uma estação na região do nó i, f^h_i não é definido. Denote por $\bar{d}^k_{ih}$ o tempo para viajar entre o nó $i \in V$ e a máquina $h \in H$, usando o veículo k. Isso é apenas definido se f^h_i é definido. O tempo para a máquina $h \in H$ atravessar o obstáculo correspondente à aresta $ij \in E^m$ é denotado por $O^h_{f^h_i f^h_j}$. É assumido que cada máquina h começa sua jornada no ponto 0, o primeiro ponto da máquina.

Denote por C_k o tempo de conclusão do veículo k e seja $C_{total} = \sum_{k \in K} C_k$ o tempo de conclusão total. O problema, portanto, consiste em definir um roteamento e escalonamento viável dos veículos e das máquinas que minimize o tempo de conclusão total.

3. Formulação de programação inteira mista

Nesta seção, nós apresentamos uma formulação baseada em arcos para o PDPTW-SE. Considere o digrafo $G' = (V', A)$, tal que $V' = V \cup \{2n+1\}$, e o nó $2n+1$ é uma cópia do depósito (nó 0). O conjunto de arcos é definido como $A = \{(i, j) \mid i, j \in V_p \cup V_d, i \neq j, i \neq$

$n+j\} \cup \{(0,j) \mid j \in V_p \cup \{2n+1\}\} \cup \{(i,2n+1) \mid i \in V_d\}$. Para cada $k \in K$, estabeleça: $d^k_{ij} = d^k_{ji} = \hat{d}^k_{ij}$ para $ij \in E$ com $i,j \in V_p \cup V_d$; $d^k_{i,2n+1} = \hat{d}^k_{0i}$ para $i \in V_d$; $d^k_{0j} = \hat{d}^k_{0j}$ para $j \in V_p$; e $d^k_{0,2n+1} = 0$. Adicionalmente, sejam A^s e A^m os arcos correspondentes às arestas em E^s e E^m, respectivamente.

Considere as seguintes variáveis relacionadas ao roteamento. A variável binária x^k_{ij} é igual a um se o veículo k atravessa o arco $(i,j) \in A$ (i.e., vem diretamente do nó $i \in V$ para o nó $j \in V'$), zero caso contrário. A variável z^k_i denota o peso do veículo $k \in K$ logo após sair do nó $i \in V'$. A parte de roteamento do problema pode, portanto, ser formulada como:

$$\sum_{j \in V_p \cup \{2n+1\}} x^k_{0j} = 1, \qquad \forall\, k \in K, \tag{1}$$

$$\sum_{j:(j,i)\in A} x^k_{ji} - \sum_{j:(i,j)\in A} x^k_{ij} = 0, \qquad \forall\, k \in K,\ i \in V_p \cup V_d, \tag{2}$$

$$\sum_{j \in V_d \cup \{0\}} x^k_{j,2n+1} = 1, \qquad \forall\, k \in K, \tag{3}$$

$$\sum_{k \in K} \sum_{j:(j,i)\in A} x^k_{ji} = 1, \qquad \forall\, i \in V_p \cup V_d, \tag{4}$$

$$\sum_{j:(j,i)\in A} x^k_{ji} = \sum_{j:(j,n+i)\in A} x^k_{j,n+i}, \qquad \forall\, k \in K,\ i \in V_p, \tag{5}$$

$$z^k_0 = 0, \qquad \forall\, k \in K, \tag{6}$$

$$z^k_j \geq z^k_i + q_j - M_1(1 - x^k_{ij}), \qquad \forall\, k \in K,\ (i,j) \in A, \tag{7}$$

$$z^k_j \leq z^k_i + q_j + M_1(1 - x^k_{ij}), \qquad \forall\, k \in K,\ (i,j) \in A, \tag{8}$$

$$z^k_i \leq \min\{Q^k, \max\{0, Q^k + q_i\}\} \sum_{j:(j,i)\in A} x^k_{ji}, \qquad \forall\, k \in K,\ i \in V_p \cup V_d, \tag{9}$$

$$z^k_i \geq q_i \sum_{j:(j,i)\in A} x^k_{ji}, \qquad \forall\, k \in K,\ i \in V_p, \tag{10}$$

$$x^k_{ij} \in \{0,1\}, \qquad \forall\, k \in K,\ (i,j) \in A, \tag{11}$$

$$z^k_i \geq 0, \qquad \forall\, k \in K,\ i \in V'. \tag{12}$$

As restrições (1)-(3) são restrições de balanceamento para os veículos. As restrições (4) asseguram que cada nó de coleta e de entrega é visitado. As restrições (5) garantem que os nós de coleta e entrega correspondentes a um dado pedido são visitados pelo mesmo veículo. As restrições (6)-(10) impõem que as capacidades dos veículos são respeitadas. As restrições (11)-(12) limitam os domínios das variáveis $\boldsymbol{x}$ e $\boldsymbol{z}$. O limite M_1 pode ser calculado como $\max\{Q_k \mid k \in K\} + \max\{q_i \mid i \in R\} + 1$.

Considere as seguintes variáveis relacionadas ao escalonamento de veículos. A variável t_i fornece o tempo de início para servir o nó $i \in V_p \cup V_d$. As variáveis t^k_0 e t^k_{2n+1} representam os tempos de partida e chegada do veículo $k \in K$ no depósito. As variáveis C_k dão o tempo de conclusão do veículo $k \in K$. Ademais, considere as seguintes variáveis relacionadas ao escalonamento de máquinas. A variável binária ϕ^h_{ij} é igual a um se a máquina $h \in H_{ij}$ é usada para atravessar o arco $(i,j) \in A^m$, zero caso contrário. A variável binária $\gamma^h_{iji'j'}$ é igual a um se o uso da máquina $h \in H_{ij} \cap H_{i'j'}$ para atravessar o arco $(i,j) \in A^m$ precede seu uso para atravessar o arco $(i',j') \in A^m$, com $(i',j') \neq (i,j)$, zero caso contrário. A variável α^h_{ij} denota o tempo de início para usar a máquina $h \in H_{ij}$ na travessia do arco $(i,j) \in A^m$. Dessa maneira, a parte de escalonamento do problema pode ser formulada como:

$$t_j \geq t_i + s_i + d^k_{ij} - M_2(1 - x^k_{ij}), \qquad \forall\, k \in K,\ (i,j) \in A,\ \text{com } i \neq 0 \text{ e } j \in V_p \cup V_d, \tag{13}$$

$$t_j \geq t^k_0 + d^k_{0j} - M_3(1 - x^k_{0j}), \qquad \forall\, k \in K,\ (0,j) \in A,\ \text{com } j \in V_p \cup V_d, \tag{14}$$

$$t_i + s_i + \sum_{k \in K} \sum_{l:(l,i) \in A} d^k_{i,n+i} x^k_{li} \leq t_{n+i}, \qquad \forall\, i \in V_p, \tag{15}$$

$$\sum_{h \in H} \phi^h_{ij} = \sum_{k \in K} x^k_{ij}, \qquad \forall\, (i,j) \in A^m, \tag{16}$$

$$\alpha^h_{ij} \geq t_i + s_i + \bar{d}^k_{ih} - M_4(1 - \phi^h_{ij}), \qquad \forall\, h \in H_{ij},\ k \in K,\ (i,j) \in A^m, i \neq 0, \tag{17}$$

$$\alpha^h_{0j} \geq t^k_0 + \bar{d}^k_{0h} - M_5(1 - \phi^h_{0j}), \qquad \forall\, h \in H_{0j},\ k \in K,\ (0,j) \in A^m, \tag{18}$$

$$t_j \geq \alpha^h_{ij} + O^h_{f^h_i f^h_j} + \bar{d}^k_{jh} - M_6(2 - \phi^h_{ij} - x^k_{ij}), \qquad \forall\, h \in H_{ij},\ k \in K,\ (i,j) \in A^m,\ j \in V_p \cup V_d, \tag{19}$$

$$\gamma^h_{iji'j'} + \gamma^h_{i'j'ij} \geq \phi^h_{ij} + \phi^h_{i'j'} - 1, \qquad \forall\, h \in H_{ij} \cap H_{i'j'},\ (i,j),(i',j') \in A^m, (i,j) \neq (i',j'), \tag{20}$$

$$\gamma^h_{iji'j'} \leq \phi^h_{ij}, \qquad \forall\, h \in H_{ij} \cap H_{i'j'},\ (i,j),(i',j') \in A^m, (i,j) \neq (i',j'), \tag{21}$$

$$\gamma^h_{i'j'ij} \leq \phi^h_{ij}, \qquad \forall\, h \in H_{ij} \cap H_{i'j'},\ (i,j),(i',j') \in A^m, (i,j) \neq (i',j'), \tag{22}$$

$$\gamma^h_{iji'j'} + \gamma^h_{i'j'ij} \leq 1, \qquad \forall\, h \in H_{ij} \cap H_{i'j'},\ (i,j),(i',j') \in A^m, (i,j) \neq (i',j'), \tag{23}$$

$$\begin{aligned} \alpha^h_{i'j'} \geq \alpha^h_{ij} + O^h_{f^h_i f^h_j} + O^h_{f^h_j f^h_{i'}} - M_7(1 - \gamma^h_{iji'j'}), \\ \forall\, h \in H_{ij} \cap H_{i'j'},\ (i,j),(i',j') \in A^m, (i,j) \neq (i',j'), \end{aligned} \tag{24}$$

$$\alpha^h_{ij} \geq O^h_{0f^h_i} - M_8(1 - \phi^h_{ij}), \qquad \forall\, h \in H_{ij} \cap H_{0i},\ (i,j) \in A^m, \tag{25}$$

$$t^k_{2n+1} \geq t_i + s_i + d^k_{i,2n+1} - M_2(1 - x^k_{i,2n+1}), \qquad \forall\, k \in K,\ (i,2n+1) \in A,\ i \neq 0, \tag{26}$$

$$\begin{aligned} t^k_{2n+1} \geq \alpha^h_{i,2n+1} + O^h_{f^h_i f^h_{2n+1}} + \bar{d}^k_{2n+1,h} - M_4(2 - \phi^h_{i,2n+1} - x^k_{i,2n+1}), \\ \forall\, h \in H_{i,2n+1},\ k \in K,\ (i,2n+1) \in A^m, \end{aligned} \tag{27}$$

$$C_k \geq t^k_{2n+1} - t^k_0, \qquad \forall\, k \in K, \tag{28}$$

$$e_i \leq t_i \leq l_i, \qquad \forall\, i \in V_p \cup V_d, \tag{29}$$

$$e_0 \leq t^k_0 \leq t^k_{2n+1} \leq l_0, \qquad \forall\, k \in K, \tag{30}$$

$$C_k \geq 0, \qquad \forall\, k \in K, \tag{31}$$

$$\alpha^h_{ij} \geq 0, \qquad \forall\, h \in H_{ij},\ (i,j) \in A^m, \tag{32}$$

$$\phi^h_{ij} \in \{0,1\}, \qquad \forall\, h \in H_{ij},\ (i,j) \in A^m, \tag{33}$$

$$\gamma^h_{iji'j'} \in \{0,1\}, \qquad \forall\, h \in H_{ij} \cap H_{i'j'},\ (i,j),(i',j') \in A^m. \tag{34}$$

As restrições (13)-(14) asseguram que toda vez que um arco é atravessado por um veículo, um limite inferior no tempo de servir seu nó de destino é determinado pelos tempos do nó de origem. As restrições (15) garantem que um nó de coleta de um pedido é visitado antes do nó de entrega. As restrições (16) implicam que uma máquina é usada para um arco se e somente se ele for atravessado por um veículo. As restrições (17)-(18) definem que toda vez que uma máquina for usada para atravessar um arco, um limite inferior no seu tempo de início na máquina é determinado pelos tempos correspondentes ao nó de origem. As restrições (19) determinam que toda vez que uma máquina é usada para atravessar um arco, os tempos das máquinas definem um limite inferior no tempo de início do seu nó de destino. As restrições (20)-(23) definem que existe uma ordem entre a travessia de dois arcos distintos se e somente se eles estão escalonados na mesma máquina. As restrições (24) definem um limite inferior no tempo para atravessar um arco toda vez que ele é precedido por outro arco. As restrições (25) definem um limite inferior no tempo para atravessar um arco (toda vez que ele é atravessado) baseado no tempo que a máquina leva entre sua estação

inicial e a estação do início de viagem. As restrições (26)-(27) determinam limites inferiores nos tempos em que o veículo chega no depósito. As restrições (28) estabelecem limites inferiores nos tempos de conclusão dos veículos. As restrições (29)-(34) limitam os domínios das variáveis $\boldsymbol{t}$, $\boldsymbol{C}$, $\boldsymbol{\alpha}$, $\boldsymbol{\phi}$, e $\boldsymbol{\gamma}$. Os limites usados na formulação devem ser grandes o suficiente. Os cálculos são omitidos neste artigo dado o limite de páginas.

O PDPTW-SE pode, portanto, ser formulado como:

$$\min \quad \sum_{k \in K} C_k \tag{35}$$
$$(1) - (34).$$

A função objetivo (35) minimiza o tempo de conclusão total.

4. Heurística construtiva

Nesta seção, nós apresentamos uma heurística construtiva gulosa para o PDPTW-SE. As ideias principais do seu funcionamento são como se segue. Inicialmente, criamos duas listas, uma com $|K|$ listas contendo as rotas iniciais dos veículos, ou seja, com início e fim no depósito, e outra com $|H|$ listas contendo as rotas iniciais das máquinas, ou seja, com início na respectiva estação inicial. Enquanto existirem pedidos não alocados nas rotas, repetimos os seguintes passos. Primeiro, selecionamos o pedido não atendido que possui a menor janela de tempo de coleta. Depois, considerando todos os veículos, encontramos duas posições na rota de um veículo, uma para coleta e outra para entrega, cuja inserção do pedido completo seja viável e seu custo, seja mínimo. O custo é definido como o tempo acrescido no tempo de conclusão do veículo escolhido. Por fim, atualizamos a rota do veículo escolhido inserindo a coleta e a entrega nas posições escolhidas e atualizamos as viagens efetuadas pelas máquinas, caso necessário. Após atender todos os pedidos, a solução é formada pelas rotas dos veículos e das máquinas construídas, cujo valor de solução será a soma dos tempos de chegada no depósito ao final da rota de cada veículo.

5. Experimentos computacionais

Experimentos computacionais preliminares foram realizados para avaliar os desempenhos das abordagens propostas. Os testes foram executados em uma máquina com processador Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz, 16 GB de memória RAM, e sistema operacional Ubuntu 22.04. Todas as implementações foram feitas em Julia. O resolvedor Gurobi v10.0.1, com configurações padrões e tempo limite de 3600 s, foi utilizado para a resolução das formulações.

5.1. Geração de instâncias

As instâncias para o PDPTW-SE foram geradas a partir das instâncias aleatórias do PDPTW de Li e Lim [2003]. Os pedidos selecionados foram separados em regiões utilizando a técnica de agrupamento k-means. As capacidades dos veículos variam em $100\% - k\Delta c\%$ da capacidade original, para $k \in \{1, ..., t\}$, sendo t a quantidade de tipos de veículos diferentes e Δc a quantidade de variações. Dadas as regiões que a máquina atende, seus pontos foram definidos ao redor do centroide das mesmas. A garantia da viabilidade das instâncias foi realizada com uma adaptação da heurística descrita na Seção 4, modificando as janelas de tempo dos pedidos quando necessário. As instâncias dos Tipos 1 e 2 possuem, respectivamente, curtos e longos horizontes de planejamento. Selecionamos as 3 primeiras instâncias aleatórias de cada tipo e variamos o número de pedidos (6, 8 e 10), veículos (mesmo número de pedidos), máquinas (3 e 4) e regiões (2 e 4). Além disso, há 3 tipos de veículos diferentes com variação de capacidade de 5%, ou seja, 100%, 95% e 90% da capacidade original. Para cada combinação, adotou-se a seguinte convenção de nome: $n_r R_n_v V_n_i I_n_m M$, onde n_r é o número de pedidos, n_v é o número de veículos, n_i é o número de ilhas (regiões), n_m é o número de máquinas. Ao todo, 72 instâncias foram geradas.

5.2. Resultados computacionais

Cada instância foi resolvida uma vez com a heurística gulosa e com a formulação propostas. A Tabela 1 apresenta os resultados médios de cada tipo de instância para cada grupo. A primeira coluna indica o tipo das instâncias. A coluna 'Grupo' possui a identificação do grupo de instâncias. Para cada método utilizado (Heurística ou Formulação) temos a coluna 'Melhor solução', a qual é a média dos valores de solução obtidos das instâncias do grupo. A coluna 'Tempo' indica o tempo médio gasto em segundos na execução da formulação para as instâncias do grupo. O tempo para a execução da heurística não foi descrito, pois para todas as instâncias o tempo máximo de execução foi de 0,5 segundos. A coluna 'Gap', definida como $100\frac{UB-LB}{UB}$, indica a média da diferença percentual entre os limites inferior (LB) e superior (UB) de cada instância do grupo. A coluna 'Soluções ótimas' indica quantas instâncias do grupo a formulação provou a otimalidade da solução. Por fim, a coluna 'Δ', definida como $100\frac{H-F}{H}$, indica a diferença percentual entre a solução da heurística (H) e da formulação (F).

Tabela 1: Resultados para as instâncias dos Tipos 1 e 2

		Heurística	Formulação				Comparação
	Grupo	Melhor solução	Melhor solução	Tempo (s)	Gap (%)	Soluções ótimas	Δ (%)
Tipo 1	6R_6V_2I_3M	876,0	826,3	15,4	0,0	3	6,1
	6R_6V_2I_4M	841,6	698,2	177,2	0,0	3	17,5
	6R_6V_4I_3M	903,3	727,3	21,7	0,0	3	18,1
	6R_6V_4I_4M	963,0	637,9	67,6	0,0	3	31,0
	8R_8V_2I_3M	1152,2	1046,3	1201,8	12,8	2	8,4
	8R_8V_2I_4M	1200,0	935,1	1201,4	13,6	2	22,7
	8R_8V_4I_3M	1269,3	973,9	180,3	0,0	3	22,8
	8R_8V_4I_4M	1447,9	1163,9	1235,8	11,0	2	20,7
	10R_10V_2I_3M	1338,0	1160,7	1625,6	17,5	2	14,0
	10R_10V_2I_4M	1318,4	1151,7	1307,3	30,0	2	12,8
	10R_10V_4I_3M[a]	2087,0	1380,7	1683,0	0,0	2	28,1
	10R_10V_4I_4M[a]	1490,7	1475,0	1328,1	0,0	2	1,0
	Média			837,1	7,1		16,9
	Total					29/36 (80,5%)	
Tipo 2	6R_6V_2I_3M	1135,7	693,4	503,6	0,0	3	37,3
	6R_6V_2I_4M	969,3	647,6	319,6	0,0	3	32,7
	6R_6V_4I_3M	1407,3	773,5	281,8	0,0	3	43,8
	6R_6V_4I_4M	1155,5	703,3	1355,4	33,3	2	38,1
	8R_8V_2I_3M	1343,4	880,3	2480,6	66,7	1	33,0
	8R_8V_2I_4M	1327,2	647,6	2592,2	66,7	1	51,0
	8R_8V_4I_3M	1659,7	900,9	2512,6	66,7	1	45,8
	8R_8V_4I_4M	1335,2	870,6	2895,3	66,7	1	34,6
	10R_10V_2I_3M	1415,1	848,2	3600,0	100,0	0	40,4
	10R_10V_2I_4M	1588,6	959,2	3600,0	90,9	0	39,1
	10R_10V_4I_3M[b]	2020,2	1399,8	3600,0	100,0	0	31,5
	10R_10V_4I_4M	1833,1	1305,2	3600,0	100,0	0	28,6
	Média			2278,4	57,6		38,0
	Total					15/36 (41,7%)	

[a] O resolvedor encontrou solução viável para as instâncias $lr101$ e $lr102$. A média da Melhor solução para a heurística foi aplicada apenas nessas instâncias.

[b] O resolvedor encontrou solução viável para as instâncias $lr201$ e $lr202$. A média da Melhor solução para a heurística foi aplicada apenas nessas instâncias.

A Tabela 1 mostra que a formulação conseguiu provar otimalidade das soluções obtidas para todas as instâncias do Tipo 1 com 6 pedidos, como também para 17/24 instâncias com 8 ou mais pedidos, totalizando 29/36 (80,5%) soluções ótimas. Embora nem todas as soluções tenham sido ótimas, o gap médio entre os limites superior e inferior para cada instância foi de 7,1%. Esses resultados são em média 16,9% melhores que os resultados obtidos pela heurística. Para as instâncias do Tipo 2, a tabela evidencia que a formulação conseguiu provar a otimalidade das soluções obtidas para 15/36 (41,7%), sendo que 11/15 soluções ótimas são para as instâncias com 6

pedidos e o restante para instâncias com 8 pedidos. Todas as instâncias com 10 pedidos alcançaram o limite de tempo. Esses resultados são em média 38% melhores que os resultados obtidos pela heurística.

Pode-se notar que a formulação levou em média 2,7 vezes mais tempo para encontrar soluções viáveis para as instâncias do Tipo 2 em comparação com instâncias do Tipo 1. Ademais, a média de *gap* entre os limites superior e inferior para as instâncias do Tipo 2 foi 8,1 vezes maior que a média para as instâncias do Tipo 1. Por fim, o número de soluções ótimas encontradas nas instâncias do Tipo 1 foi aproximadamente 2 vezes maior em comparação com o do Tipo 2. Portanto, a formulação demonstrou ter mais dificuldade na resolução de instâncias com horizonte de planejamento mais longo. Ao todo, 44/72 (61,1%) das soluções foram ótimas.

6. Considerações finais

Neste trabalho, introduzimos e formalizamos o problema de coleta e entrega com janelas de tempo e escalonamento nas arestas (PDPTW-SE, do inglês *pickup and delivery problem with time windows and scheduling on the edges*). O PDPTW-SE é um problema NP-difícil que integra as tomadas de decisão associadas ao roteamento de veículos com aquelas vinculadas ao uso de máquinas para atendimento de pedidos situados em diferentes regiões inacessíveis apenas por veículos. Foram propostas uma formulação de programação inteira mista e uma heurística construtiva simples que visa fornecer um limite superior inicial de forma rápida para qualquer instância. Outra contribuição deste trabalho foi o desenvolvimento de um gerador de instâncias para o PDPTW-SE.

Resultados preliminares utilizando instâncias geradas no contexto deste artigo mostraram que o modelo consegue obter soluções ótimas no limite de uma hora para instâncias com até 10 pedidos e com curto horizonte de planejamento. Em contrapartida, o modelo encontra dificuldades para resolver de forma ótima dentro do mesmo limite de tempo instâncias a partir de 8 pedidos e com longo horizonte de planejamento.

Agradecimentos

O trabalho de Rafael A. Melo foi apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), projeto número 314662/2020-0.

Referências

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., e Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *TOP*, 15(1):1–31.

Bodin, L. D. e Berman, L. (1979). Routing and scheduling of school buses by computer. *Transportation Science*, 13(2):113–129.

Cullen, F. H., Jarvis, J. J., e Ratliff, H. D. (1981). Set partitioning based heuristics for interactive routing. *Networks*, 11(2):125–143.

Dahle, L., Andersson, H., Christiansen, M., e Speranza, M. G. (2019). The pickup and delivery problem with time windows and occasional drivers. *Computers & Operations Research*, 109:122–133.

Dumas, Y., Desrosiers, J., e Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54(1):7–22.

Fikar, C. e Hirsch, P. (2017). Home health care routing and scheduling: A review. *Computers & Operations Research*, 77:86–95.

Ghilas, V., Cordeau, J.-F., Demir, E., e Van Woensel, T. (2018). Branch-and-price for the pickup and delivery problem with time windows and scheduled lines. *Transportation Science*, 52(5):1191–1210.

Li, H. e Lim, A. (2003). A metaheuristic for the pickup and delivery problem with time windows. *International Journal on Artificial Intelligence Tools*, 12(02):173–186.

Parragh, S. N., Doerner, K. F., e Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, 58(1):21–51.

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. 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.

Savelsbergh, M. W. e Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, 29 (1):17–29.