

UM ALGORITMO HÍBRIDO PARA O *HETEROGENEOUS DIAL-A-RIDE PROBLEM*

Igor Malheiros, Rodrigo Ramalho, Bruno Passeti, Teobaldo Bulhões, Anand Subramanian

Universidade Federal da Paraíba

Centro de Informática, Rua dos Escoteiros s/n, Mangabeira, 58055-000, João Pessoa, Brasil

igormalheiros92@gmail.com, rodrigoramalho.dev@gmail.com,

brunopasseti@eng.ci.ufpb.br, tbulhoes@ci.ufpb.br,

anand@ci.ufpb.br

RESUMO

Os problemas de roteamento de veículos emergem em inúmeras situações práticas em logística de transporte. Dentre elas, pode-se destacar o *dial-a-ride problem* (DARP), que consiste em construir rotas de custo mínimo que atendam requisições de coleta e entrega, obedecendo as restrições de capacidade, janelas de tempo e tempo máximo de viagem. Este trabalho propõe um algoritmo híbrido para resolver uma variante do DARP com demanda e frota heterogêneas. O método combina a meta-heurística *iterated local search* (ILS) com um procedimento exato para resolver o problema de particionamento de conjuntos. Além disso, diversos procedimentos foram implementados para acelerar as buscas locais. Experimentos computacionais foram realizados em instâncias da literatura e os resultados obtidos sugerem que o algoritmo proposto é capaz de produzir soluções de alta qualidade, inclusive encontrando uma nova melhor solução, em tempos computacionais competitivos.

PALAVRAS CHAVE. Roteamento de veículos. *Dial-a-ride*. *Iterated local search*.

MH – Meta-heurísticas. L&T – Logística e Transportes. OC – Otimização Combinatória

ABSTRACT

Vehicle routing problems arise in many practical situations regarding transportation logistics. Among them, one can highlight the dial-a-ride problem (DARP), which consists of designing least-cost routes to serve pickup-and-delivery requests, while meeting capacity, time windows and maximum ride time constraints. This work proposes a hybrid algorithm to solve a DARP variant where both the demands and vehicle fleet are heterogeneous. The method combines the iterated local search (ILS) metaheuristic with an exact procedure based on a set partitioning approach. In addition, several procedures were also implemented to speed-up the local search phase. Computational experiments were conducted on benchmark instances, and the results obtained suggest that the proposed algorithm is capable of producing highly quality solutions, even improving one best known solution, in a very competitive runtime.

KEYWORDS. Vehicle routing. *Dial-a-ride*. *Iterated local search*.

MH – Metaheuristics. L&T – Logistics and Transportation. CO – Combinatorial Optimization

1. Introdução

Muitos problemas práticos que aparecem em logística de transporte, como por exemplo distribuição de mercadorias, coleta de lixo e entrega de jornais, podem ser modelados por uma classe de problemas conhecida como *problema de roteamento de veículos* (PRV). Dentre as variantes nesta classe de problemas, uma aplicação aparece para serviços de transporte particular de pessoas. Os usuários requisitam que veículos façam coleta e entrega de pessoas em determinados lugares, estabelecendo restrições de tempo de saída e chegada e duração máxima da viagem. Esse tipo de aplicação está se tornando cada vez mais popular nos últimos anos [Henao, 2017].

Designar quais veículos devem atender quais requisições, em qual ordem, por quais caminhos e obedecendo restrições de tempo, duração e capacidade se torna uma tarefa muito difícil quando as demandas crescem. Além disso, tal designação deve minimizar os custos de transporte. O problema descrito é conhecido como *dial-a-ride problem* (DARP) [Cordeau e Laporte, 2003a].

Uma das mais conhecidas aplicações do DARP é para o transporte de pessoas idosas ou deficientes, no qual as transportadoras devem buscar essas pessoas em casa e deixá-las em seus destinos. Uma segunda aplicação conhecida é usada nos sistemas de saúde, em que clientes que precisam se deslocar entre centros médicos, ou de suas casas para clínicas, ou ainda de locais de acidentes para hospitais. Nessa variação, os clientes possuem uma demanda variável, eles podem necessitar serem transportados, por assentos comuns, por macas ou por cadeiras de rodas. Essa variação é apresentada como (*heterogeneous DARP* – HDARP) [Parragh, 2011].

Incorporando restrições de janela de tempo e de tempo máximo de viagem, o DARP pode ser visto como uma generalização do *problema de coleta e entrega*. Encontrar uma solução viável para o DARP é $\mathcal{NP}$ -Difícil, uma vez que ele também é uma generalização do *problema do caixeiro viajante com janela de tempo* [Savelsbergh, 1985; Cordeau, 2006].

Uma abordagem por algoritmos exatos para resolver o HDARP pode necessitar de tempos computacionais longos em instâncias grandes, sendo intratáveis para fins práticos. Nenhum dos métodos da literatura resolve de forma consistente instâncias acima de 80 requisições [Braekers et al., 2014]. As meta-heurísticas surgem como uma alternativa viável para obter soluções próximas ou iguais às ótimas em tempos computacionais aceitáveis. Dessa forma, esses métodos se apresentam como mais adequados para lidar com instâncias de médio e grande porte. Uma abordagem promissora seria combinar meta-heurística com métodos exatos, criando um algoritmo híbrido. Esses métodos vem se popularizando cada vez mais principalmente pelos bons resultados obtidos para resolver PRVs e algumas de suas variações [Subramanian et al., 2013].

Este trabalho apresenta uma abordagem híbrida para resolver instâncias do HDARP disponibilizadas na literatura. O algoritmo se baseia no *iterated local search* (ILS) com descida em vizinhança variável aleatória (*random variable neighborhood descent* – RVND) junto a procedimentos exatos. A ideia é armazenar um conjunto de rotas geradas durante a heurística e, em seguida, resolver o problema de *set partitioning* (SP) de forma exata para extrair a melhor combinação de rotas. Os resultados dos testes sobre as instâncias mostraram que o algoritmo é capaz de produzir soluções de alta qualidade em tempos computacionais competitivos.

2. Trabalhos Relacionados

A definição mais trabalhada do DARP na literatura foi apresentada por Cordeau e Laporte [2003a]. As rotas devem respeitar restrições de janela de tempo, precedência entre coleta e entrega, capacidade dos veículos, máxima duração de viagem dos clientes e máxima duração da rota.

O trabalho de Cordeau e Laporte [2003b] introduziu uma abordagem via meta-heurística busca tabu. Os autores reportaram dificuldades nas avaliações das restrições de máxima duração de viagem em uma rota. Por isso, foi proposto um algoritmo capaz de verificar se a rota satisfaz

tais restrições. Esse algoritmo tem complexidade de tempo $\mathcal{O}(n^2)$, que, dentro de uma avaliação de movimento, pode se tornar uma operação de alto custo computacional.

Uma abordagem para o DARP via programação inteira mista (do inglês, *mixed-integer programming* – MIP) e um algoritmo *branch-and-cut* (B&C) foi realizada por Cordeau [2006]. Foi apresentada uma formulação matemática para o problema, junto a desigualdades válidas herdadas de problemas relacionados e novas desigualdades válidas específicas para o DARP. Como resultado, o autor identificou que o algoritmo B&C foi capaz de resolver instâncias com até 24 requisições.

Uma variação do DARP foi introduzida por Parragh [2011], em que os clientes podem demandar diferentes tipos de recursos em um veículo. O problema apresenta demanda e frota heterogênea, HDARP. Essa variação foi motivada pela *Austrian Red Cross* (organização nacional da Cruz Vermelha na Áustria). Os pacientes podem ser transportados de três formas: sentados, em cadeiras de rodas ou em uma maca. O trabalho apresentou duas formulações matemática para o HDARP, algoritmos de B&C e uma meta-heurística de *variable neighborhood search*. Os testes foram realizados em adaptações das instâncias de [Cordeau e Laporte, 2003a]. Os métodos exatos foram capazes de encontrar a solução ótima em instâncias com até 40 requisições, enquanto a meta-heurística forneceu resultados iguais ou próximos aos ótimos.

Modificando o HDARP para que os veículos saiam de diferentes depósitos, Braekers et al. [2014] introduziram o problema denominado por *Multi-depot HDARP* (MDHDARP). Foi criada uma nova modelagem matemática compacta junto à uma adaptação do algoritmo de B&C para HDARP publicado em [Parragh, 2011]. Os resultados mostraram que o novo algoritmo conseguiu resolver, de forma ótima e em segundos, todas as instâncias do HDARP da literatura. Os autores também implementaram a meta-heurística de *deterministic annealing* (DA). O algoritmo foi testado para resolver instâncias do DARP, HDARP e MDHDARP. Os resultados superaram outras abordagens meta-heurísticas do estado da arte daquele momento tanto em tempo computacional, como em qualidade da solução. Por último, foram apresentadas novas instâncias para o HDARP e para o MDHDARP com até 8 veículos e 96 requisições.

Uma aplicação do DARP para veículos elétricos em que trocas de baterias podem ser realizadas durante o seu percurso foi proposta por Masmoudi et al. [2018]. Desenvolveu-se um algoritmo que combina o VNS com abordagens evolutivas. Além de experimentos em adaptações de instâncias para o problema, os autores realizaram testes nas instâncias do HDARP [Braekers et al., 2014]. Para o HDARP, os foram resultados competitivos em qualidade de solução, porém obtiveram desempenho inferior em tempos computacionais.

O trabalho de Gschwind e Drexel [2019] contribuiu com uma checagem da viabilidade de rota para as restrições de tempo máximo de viagem em complexidade de tempo de $\mathcal{O}(1)$ amortizado. Com pré-processamento de estruturas de dados auxiliares, a verificação tem custo de tempo constante independente da quantidade de requisições na rota. Essa rotina de verificação foi implementada na meta-heurística *adaptive large neighborhood search* (ALNS). Os resultados apresentaram novas melhores soluções para o DARP.

3. Definição do problema

Sejam n o número de requisições e v o número de veículos. O conjunto $P = \{1, \dots, n\}$ representa os vértices de coleta, enquanto que o conjunto $D = \{n + 1, \dots, 2n\}$ representa os vértices de entrega. Além disso, sejam $M_o = 2n + 1$ e $M_e = 2n + 2$ os vértices relativos ao depósito de origem e de fim de rota, respectivamente. Para cada requisição, existe um vértice de coleta $i \in P$ e um vértice de entrega $i + n \in D$. Define-se o conjunto $N = P \cup D$, isto é, vértices de coleta e entrega, e o conjunto de todos os vértices $V = N \cup M_o \cup M_e$.

Os arcos entre os vértices são distribuídos nos seguintes conjuntos: A^{mp} contém todos arcos que conectam o depósito de M_o a um vértice de coleta $i \in P$; A^{pd} contém todos arcos que conectam um vértice de coleta ou entrega $i \in N$ a outro vértice de coleta ou entrega $j \in N$ tal que $i \neq j$ e $j + n \neq i$, ou seja, não existem arcos de um vértice para ele mesmo, nem de um vértice de entrega para sua coleta; A^{dm} possui todos os arcos que conectam um vértice de entrega $j \in D$ ao depósito M_e ; e A^{mm} é o conjunto com o arco do depósito de origem M_o para o depósito final M_e , para o caso de rotas vazias. Por fim, define-se $A = A^{mp} \cup A^{pd} \cup A^{dm} \cup A^{mm}$ como o conjunto de arcos que representa a união de todos os quatro conjuntos definidos anteriormente.

Com as definições dos conjuntos V e A , define-se $G = (V, A)$ como um grafo orientado com conjunto de vértices V e de arcos A . Para cada arco $(i, j) \in A$, existem valores t_{ij} e c_{ij} que indicam o tempo e o custo para percorrê-lo, respectivamente. Assume-se que t_{ij} e c_{ij} satisfazem a desigualdade triangular.

Outros conjuntos usados no problema são $K = \{1, \dots, v\}$, correspondente aos veículos, e $R = \{0, \dots, 3\}$, que indica os recursos que podem ser solicitados por uma requisição. Atribui-se 0 para assentos de acompanhantes, 1 para assentos de pacientes, 2 para macas e 3 para cadeiras de rodas. Uma demanda de assento para acompanhante pode consumir os recursos 0, 1 ou 2; uma demanda de assento para paciente pode consumir os recursos 1 ou 2; por fim, demandas de viagem por maca ou cadeira de rodas só podem consumir seus respectivos recursos. Os veículos com mesma capacidade e mesmo tempo máximo de viagem são agrupados em um conjunto de tipos de veículos $\mathcal{K}$, sendo a quantidade de cada tipo de veículo existente na frota indicada por U_k para $k \in \mathcal{K}$.

Para cada vértice $i \in V$, existe um tempo de serviço d_i , uma janela de tempo $[e_i, l_i]$ e uma demanda q_i^r , $r \in R$. Os vértices de coleta e entrega possuem a relação de que $q_{i+n}^r = -q_i^r$ e os vértices de depósito possuem a relação de que $q_{2n+1}^r = q_{2n+2}^r = 0$. Para cada $i \in P \cup M_o$ existe um tempo máximo de viagem L_i . Para cada tipo de veículo $k \in K$ e recurso $r \in R$, existe uma capacidade C^{rk} . Rotas devem ser criadas para atender todas as requisições, tendo como objetivo minimizar os custos. Para isso, as rotas devem satisfazer as restrições apresentadas a seguir.

- Uma requisição i deve ser atendida por exatamente um veículo. Esse veículo deve visitar o vértice i , para fazer a coleta das demanda, antes de visitar o vértice $i + n$, onde ele fará a entrega da demanda.
- Cada veículo deve iniciar sua rota no depósito de M_o e terminar no depósito final de M_e .
- As capacidades de um veículo C^{rk} , $r \in R$, $k \in K$ não pode ser excedidas em nenhum trecho da rota.
- Cada requisição deve ter a sua coleta e a sua entrega atendidas dentro de suas janelas de tempo, $[e_i, l_i]$ e $[e_{i+n}, l_{i+n}]$, para todo $i \in P$.
- O tempo total entre a coleta e a entrega da requisição i não deve exceder seu tempo máximo de viagem L_i , $i \in P$. Isto é, a diferença entre os tempos de começo de serviço dos vértice de coleta e a chegada ao vértice de entrega deve ser menor que o tempo máximo de viagem. Essa mesma restrição é usada entre o tempo de saída e de retorno ao depósito, que não deve exceder o tempo máximo de duração da rota, L_{m_o} , onde $m_o \in M_0$.

4. Método Proposto

O algoritmo proposto, chamado de ILS-SP_{HDARP}, combina a meta-heurística ILS com abordagem exata para resolver o problema SP. Durante o ILS, a busca local é realizada via RVND [Subramanian et al., 2010]. O Algoritmo 1 descreve o funcionamento do método.

A primeira operação do algoritmo é um pré-processamento sobre os dados (linha 2). Baseado nos trabalhos de Cordeau [2006] e Parragh [2011], são feitos estreitamentos das janelas de tempo e uma análise sobre pares de requisições para proibir arcos que violam janela de tempo ou tempo máximo de viagem. Além disso, modificações sobre as representações das demandas das requisições e das capacidades dos veículos são realizada para que cada recurso possa ser processado e avaliado de forma independente durante a execução do método.

O laço de iteração principal do ILS (linhas 4–20) começa com uma solução inicial gerada pelo método construtivo (linha 5). Por meio dos procedimentos RVND (linha 9 e 11) e mecanismos de perturbação (linha 16), procura-se melhorar a qualidade da solução. Após cada busca local, a solução corrente é armazenada em uma estrutura (*pool* de rotas) para ser usada posteriormente no SP (linha 12). O parâmetro I_{ILS} indica o número máximo de perturbações consecutivas permitidas sem que ocorra melhora da solução. Após o ILS atingir seu número máximo de iterações, o procedimento para resolver o SP é realizado (linhas 21–26). Esse procedimento procura em um conjunto de rotas armazenados no *pool* uma solução que forneça o menor custo. Sempre que uma solução inteira e com custo menor que a melhor solução corrente é encontrada pelo SP, executa-se o ILS com apenas um iteração e cuja a solução inicial seja a solução inteira encontrada pelo SP, conforme descrito em Subramanian et al. [2013].

Algoritmo 1 ILS-SP_{HDARP}

```
1: Procedimento ILS-SPHDARP ( $I_R, I_{ILS}, N_{pert}$ )
2: préprocessamento()
3:  $f^* \leftarrow \infty$ 
4: for  $i = 1$  to  $I_R$  do
5:    $s \leftarrow \text{geraSolucaoInicial}()$ 
6:    $s' \leftarrow s$ 
7:    $iter \leftarrow 0$ 
8:   while  $iter \leq I_{ILS}$  do
9:      $s \leftarrow \text{RVND}_{INTER}(s)$ 
10:    if  $iter == I_{ILS}$  then
11:       $s \leftarrow \text{RVND}_{INTRA}(s)$ 
12:       $\text{atualizaPool}(\text{pool}, s)$ 
13:      if  $f(s) < f(s')$  then
14:         $s' \leftarrow s$ 
15:         $iter \leftarrow 0$ 
16:       $s \leftarrow \text{perturba}(s', N_{pert})$ 
17:       $iter \leftarrow iter + 1$ 
18:    if  $f(s') < f^*$  then
19:       $s^* \leftarrow s'$ 
20:       $f^* \leftarrow f(s')$ 
21:  while true do
22:     $s^* \leftarrow \text{SP}(\text{pool}, s^*, I_{ILS}, N_{pert})$ 
23:    if  $f(s^*) < f^*$  then
24:       $f^* \leftarrow f(s^*)$ 
25:    else
26:      break
27: return  $s^*$ 
```

4.1. Procedimento construtivo

O procedimento guloso aleatório usado para criação da solução inicial armazena e mantém uma lista de candidatos de vértices (*CL*) a serem inseridos em alguma rota. Cada elemento dessa lista corresponde a uma requisição, ou seja, um par de vértices coleta/entrega. Para cada veículo, inicia-se uma rota com somente dois vértices, o depósitos inicial e final da rota. Em seguida, para cada rota, um elemento da *CL* é escolhido aleatoriamente para ser inserido entre os depósitos, garantindo diversificação na solução inicial. Depois, o algoritmo utiliza uma abordagem gulosa:

para cada elemento da CL , avalia-se em todas as rotas e em todas as posições de uma rota qual inserção tem menor custo e não viola restrições. Deve-se encontrar a melhor posição para inserir um vértice de coleta e um vértice de entrega, onde a entrega sempre vem após a coleta. Encontrada a melhor posição de inserção, o par é inserido na rota e removido da CL . O procedimento se repete até que a lista de candidatos esteja vazia. Caso não seja possível inserir algum elemento da CL sem violação das restrições, uma nova construção é iniciada.

4.2. Busca Local

A busca local é realizada via RVND [Subramanian et al., 2010] e tem como objetivo melhorar a qualidade de uma solução corrente. Para isso, o procedimento utiliza uma estrutura de vizinhança enumerada, $\mathcal{N} = \{\mathcal{N}^{(1)}, \mathcal{N}^{(2)}, \dots, \mathcal{N}^{(n)}\}$, em que cada elemento representa um tipo de movimento. O algoritmo escolhe uma das vizinhanças de forma aleatória e aplica o movimento sobre a solução corrente. Caso não se obtenha melhora na função objetivo, o movimento escolhido é removido da lista; caso contrário, a lista é reiniciada. Neste trabalho, duas versões do RVND foram implementadas, uma apenas para movimentos inter-rota (RVND_{INTER} com $\mathcal{N}_{inter}$) e outra apenas para movimentos intra-rota (RVND_{INTRA} com $\mathcal{N}_{intra}$). Todos os movimentos utilizam o método de *best improvement*. Eles realizam a exploração apenas sobre soluções viáveis. A seguir são detalhados os movimentos utilizados na busca local.

- **Shift** – $\mathcal{N}_{inter}^{(1)}$: Os vértices de coleta e entrega de uma mesma requisição são movidos de uma rota π_1 para outra rota π_2 .
- **Swap** – $\mathcal{N}_{inter}^{(2)}$: Duas requisições, sendo um pertencente a uma rota π_1 e a outra a uma segunda rota π_2 , são permutadas. Ou seja, acontece a permuta entre os vértices de coleta e entre os de entrega.
- **Cross** – $\mathcal{N}_{inter}^{(3)}$: Dois arcos uv e $u'v'$ pertencentes a rotas distintas são removidos. Em seguida, criam-se dois novos arcos uv' e $u'v$.
- **SwapVehicle** – $\mathcal{N}_{inter}^{(4)}$: Permutam-se os veículos entre um par de rotas π_1 e π_2 .
- **Reinsertion** – $\mathcal{N}_{intra}^{(1)}$: As posições dos vértices de coleta e entrega de uma mesma requisição são redefinidas dentro da mesma rota.
- **Exchange** – $\mathcal{N}_{intra}^{(2)}$: As posições de dois vértices de uma mesma rota são permutadas, podendo eles serem de requisições ou tipos (coleta ou entrega) diferentes.

As vizinhanças do conjunto $\mathcal{N}_{inter}$ são aplicadas para cada par de rotas, por isso, a vizinhança $\mathcal{N}_{inter}^{(1)}$ possui o tamanho $\mathcal{O}(n^3)$, $\mathcal{N}_{inter}^{(2)}$ e $\mathcal{N}_{inter}^{(3)}$ possuem tamanho $\mathcal{O}(n^2)$, e $\mathcal{N}_{inter}^{(4)}$ tem tamanho $\mathcal{O}(v^3)$. Para o conjunto de vizinhanças intra-rota $\mathcal{N}_{intra}$ a exploração é realizada em cada rota, assim, $\mathcal{N}_{intra}^{(1)}$ possui o tamanho $\mathcal{O}(n^3)$ e $\mathcal{N}_{intra}^{(2)}$ tamanho $\mathcal{O}(n^2)$.

4.3. Avaliação de movimento

Como uma grande quantidade de comparações é realizada durante os movimentos de vizinhança, é importante que essas avaliações sejam realizadas da forma mais eficiente possível, idealmente com complexidade de tempo constante. As avaliações devem checar não só os novos custos de remoção/inserção de uma requisição, como também a viabilidade das rotas envolvidas.

As restrições de paridade e precedência das requisições são naturalmente satisfeitas dado que a construção inicial e os movimentos de busca sempre inserem o par coleta e entrega juntos,

onde a entrega sempre vem após a coleta. Para as restrições de carga e janela de tempo, o algoritmo utiliza estruturas de dados auxiliares para cada *subsequência* de rotas, inspiradas nos trabalhos de Vidal et al. [2013] e Bulhões et al. [2018]. Essas estruturas são pré-processadas em tempo $\mathcal{O}(n^2)$ para cada rota e permitem avaliações em tempo constante amortizado durante as buscas.

Seja $\sigma = (\sigma^0, \dots, \sigma^{|\sigma|-1})$ uma subsequência e σ^{ij} a subsequência de σ que começa na i -ésima posição e termina na j -ésima posição, i.e., $\sigma^{ij} = (\sigma^i, \dots, \sigma^j)$. Para cada possível σ de uma rota, as estruturas armazenam e atualizam os seguintes valores:

- $q_{sum}^r(\sigma)$: soma das cargas de coleta/entrega (carga acumulada) para cada recurso $r \in R$;
- $q_{max}^r(\sigma)$: máxima carga acumulada para cada recurso $r \in R$;
- $D(\sigma)$: mínima duração para percorrer σ ;
- $TW(\sigma)$: mínimo *time-warp*, onde *time-warp* é a quantidade de tempo excedida do final da janela de tempo;
- $E(\sigma)$: horário mais cedo de visita ao primeiro vértice tal que a duração e o *time-warp* sejam mínimos;
- $L(\sigma)$: horário mais tarde de visita ao primeiro vértice tal que a duração e o *time-warp* sejam mínimos;

Se a subsequência σ' é composta apenas pelo vértice i , então $q_{sum}^r(\sigma') = q_i^r$, $q_{max}^r(\sigma') = \max(0, q_i^r)$, $D(\sigma') = d_i$, $TW(\sigma') = 0$, $E(\sigma') = e_i$, $L(\sigma') = l_i$. Tomando o operador $\oplus$ como sinal de concatenação entre duas subsequências distintas. A seguir é apresentado que qualquer subsequência σ , tal que $|\sigma| > 1$, pode ser derivada de duas outras subsequências σ^1 e σ^2 , a operação de concatenação é caracterizada pela seguinte atualização de dados:

$$q_{sum}^r(\sigma^1 \oplus \sigma^2) = q_{sum}^r(\sigma^1) + q_{sum}^r(\sigma^2), \forall r \in R \quad (1)$$

$$q_{max}^r(\sigma^1 \oplus \sigma^2) = \max\{q_{max}^r(\sigma^1), q_{sum}^r(\sigma^1) + q_{max}^r(\sigma^2)\}, \forall r \in R \quad (2)$$

$$D(\sigma^1 \oplus \sigma^2) = D(\sigma^1) + D(\sigma^2) + t_{\sigma^1(|\sigma^1|)\sigma^2(1)} + \Delta_{WT} \quad (3)$$

$$E(\sigma^1 \oplus \sigma^2) = \max\{E(\sigma^2) - \Delta, E(\sigma^1)\} - \Delta_{WT} \quad (4)$$

$$L(\sigma^1 \oplus \sigma^2) = \min\{L(\sigma^2) - \Delta, L(\sigma^1)\} + \Delta_{TW} \quad (5)$$

$$TW(\sigma^1 \oplus \sigma^2) = TW(\sigma^1) + TW(\sigma^2) + \Delta_{TW}, \quad (6)$$

onde:

$$\Delta = D(\sigma^1) - TW(\sigma^1) + t_{\sigma^1(|\sigma^1|)\sigma^2(1)} \quad (7)$$

$$\Delta_{WT} = \max\{E(\sigma^2) - \Delta - L(\sigma^1), 0\} \quad (8)$$

$$\Delta_{TW} = \max\{E(\sigma^1) + \Delta - L(\sigma^2), 0\}. \quad (9)$$

Uma rota é viável para carga e janela de tempo se sua subsequência $\sigma = (\sigma^0, \dots, \sigma^{|\sigma|-1})$ possui $TW = 0$ e $q_{max}^r \leq C^{rk}$ para todo $r \in R$, onde k é o veículo associado à rota.

Quando a rota é viável para as restrições anteriores e melhoram a qualidade da solução, avaliações são feitas sobre as restrições de máximo tempo de viagem e máxima duração da rota. Para isso, um procedimento que executa o algoritmo de 8 passos apresentado por Cordeau e Laporte [2003b] é realizado. Esse procedimento tem complexidade tempo de $\mathcal{O}(n^2)$ que, dentro da avaliação do movimento de vizinhança, representa um alto custo computacional.

4.4. Perturbação

Dois tipos de perturbações foram usadas no ILS-SP_{HDARP}, ambas escolhidas de forma aleatória e com mesma probabilidade. No primeiro método, escolhe-se aleatoriamente uma rota π e remove-se um par coleta/entrega também escolhido de forma aleatória. Esse par é inserido em posições aleatórias em uma rota extra π_e com custo penalizado. O procedimento é repetido aleatoriamente entre 1 e N_{pert} vezes. No segundo método, escolhe-se aleatoriamente uma rota π e remove-se um bloco aleatório onde os vértices no começo e no final do bloco possuem cargas acumuladas zeradas. Esse bloco é inserido em uma posição aleatória na rota extra π_e com custo penalizado. O procedimento é repetido aleatoriamente entre 1 e $N_{pert}/2$ vezes. A rota extra π_e , que pode conter inviabilidades, é inserida na solução corrente. Portanto, as buscas locais são realizadas também em π_e que naturalmente ficará vazia por causa do seu custo penalizado.

4.5. Abordagem baseada em *set partitioning*

Rotas de soluções ótimas locais geradas após a execução de cada busca local do ILS-SP_{HDARP} são armazenadas em uma estrutura chamada de *pool* para serem usadas como entrada para o SP. O método procura qual a combinação de rotas fornecidas tem menor custo. A solução para o modelo SP é encontrada de forma exata por um resolvidor de MIP.

Seja $\mathcal{R}$ o conjunto de todas as rotas armazenadas no *pool*, $\mathcal{R}_i \subseteq \mathcal{R}$ é definido como o subconjunto de rotas que possui a requisição $i \in P$. Define-se $\mathcal{R}_k \subseteq \mathcal{R}$ como o subconjunto de rotas que aparecem o veículo do tipo $k \in \mathcal{K}$. A variável y_j assume valor 1 caso a rota $j \in \mathcal{R}$ faça parte do particionamento, ou 0 caso contrário. A constante c_j representa o custo da rota $j \in \mathcal{R}$.

$$\min \sum_{j \in \mathcal{R}} c_j y_j \quad (10)$$

$$\text{s.a.: } \sum_{j \in \mathcal{R}_i} y_j = 1 \quad i \in P \quad (11)$$

$$\sum_{j \in \mathcal{R}_k} y_j \leq U_k \quad k \in \mathcal{K} \quad (12)$$

$$y_j \in \{0, 1\} \quad j \in \mathcal{R}. \quad (13)$$

A função objetivo (10) minimiza a soma dos custos das rotas. As restrições (11) garantem que para cada vértice $i \in P$ existe uma única rota $j \in \mathcal{R}_i$. As restrições (12) indicam que a quantidade de veículos de um certo tipo não deve ultrapassar o limite da frota. As restrições (13) referem-se ao domínio das variáveis de decisão y_j .

4.6. Mecanismos de aceleração

Os seguintes mecanismos de aceleração foram propostos.

- *Memoization Move Descriptor* (MMD) — O primeiro mecanismo é uma extensão daquele proposto por Zachariadis e Kiranoudis [2010], denominado *Static Move Descriptor*, que foi posteriormente aprimorado por Beek et al. [2018]. A estrutura guarda, para cada movimento de busca local e para cada rota (ou par de rotas), as informações de quais vértices e para quais posições devem ir tal que forneçam o melhor movimento viável. Em outras palavras, uma vez que uma rota (ou par de rotas) é analisada por um movimento, armazena-se os vértices e posições que geram uma solução viável de menor custo. Caso o movimento não gere nenhuma melhoria para aquela rota (ou pares de rotas), a informação também é salva. Dessa forma, antes da avaliação de um movimento, verifica-se se aquela rota já foi analisada em

algum momento do algoritmo. Se sim, a MMD retorna a informação do melhor movimento previamente checado (ou que não existe melhoria possível). Caso contrário, a avaliação do movimento é feita normalmente e, ao final, armazena-se a informação do movimento para a rota (ou pares de rotas) que foi avaliado.

- *Memoization Solution Descriptor (MSD)* — A mesma ideia do mecanismo MMD pode ser estendida para salvar soluções já encontradas pelo ILS-SP_{HDARP}. A estrutura permite salvar soluções ótimas locais que já foram visitadas pelo algoritmo em algum momento de sua execução, junto com a solução, salva-se também a iteração i corrente. Caso o ILS-SP_{HDARP} encontre novamente uma solução já armazenada na MSD e cuja a iteração salva seja diferente da iteração i corrente, a iteração i é abortada. Isso permite que a heurística não fique realizando buscas a partir de soluções ótimas locais previamente visitadas e que, ou alguma perturbação já levou até outro ótimo local, ou I_{ILS} perturbações não foram suficientes para escapar do ótimo local.
- *Feasible Search Range (FSR)* — O último mecanismo aproveita-se das propriedades das janelas de tempo dos vértices. Quando dois vértices possuem janelas de tempo disjuntas entre si, pode-se afirmar que um dos vértices deve vir antes (ou depois) do outro caso pertençam à mesma rota. Essas informações entre todos os pares vértices podem ser armazenadas durante o pré-processamento de dados. Utilizando essas informações, é possível identificar para cada rota qual é a primeira e a última posição viável para as restrições de janela de tempo de todos os vértices de entrada do problema. Assim, a FSR armazena para cada rota e para cada vértices qual a primeira e a última posição viável em janela de tempo que esse vértice poderá aparecer. A FSR é utilizada durante as avaliações de movimentos com o objetivo de fornecer um intervalo mais restrito em que um novo vértice poderá aparecer na rota.

5. Resultados

O algoritmo proposto foi executado em um computador com processador Intel(R) Core(TM) i5-9600KF CPU 3.70 GHz e 8 GB de memória RAM. A máquina utiliza o sistema operacional Linux Ubuntu 18.04. Os métodos exatos foram implementados com o auxílio do resolvidor CPLEX 12.7. No mesma máquina, foram feitos testes utilizando a meta-heurística proposta por Braekers et al. [2014], cujo código foi fornecido pelos autores. Dessa forma, pôde ser feita uma comparação mais realista entre os tempos de execução dos dois algoritmos.

Foram realizados experimentos sobre um conjunto de instâncias da literatura [Parragh, 2011; Braekers et al., 2014]. As instâncias são agrupadas em três conjuntos: U , que utiliza apenas um único recurso de demanda/capacidade; E , que possui os quatro tipos de recursos e frota homogênea; e, por último, I , que, além de utilizar os quatro tipos de recurso, possui frota heterogênea. As instâncias variam de 2 a 8 veículos e de 16 a 96 requisições. Os pontos de coleta/entrega são localizados em um plano com limites $[-10, 10]^2$. O depósito é localizado no centro do quadrado.

Para o ILS-SP_{HDARP}, foram utilizados os parâmetros: $I_R = 20$, $I_{ILS} = \max(150, 2 \times n)$ e $N_{pert} = 6$. Foram realizadas 10 execuções para cada instância, sendo os resultados reportados na Tabela 1. Para cada instância, são apresentados os valores das melhores soluções (do inglês, *best known solution* – BKS) e os melhores *lower bounds* (LBs) disponíveis na literatura. Em seguida, são apresentados os resultados referentes à heurística proposta por Braekers et al. [2014] e indicam respectivamente: melhor valor encontrado pelo DA, média dos valores encontrados, o *gap* das médias em relação aos LBs, o tempo médio de execução reportado pelos autores e o tempo médio que o algoritmo utilizou na mesma máquina dos experimentos deste trabalho em 10 execuções. É

importante destacar que em Braekers et al. [2014] são apresentados resultados apenas em instâncias de até 4 veículos, os demais resultados foram disponibilizados depois pelos autores. Por último, os demais valores se referem ao algoritmo proposto neste artigo e representam: melhor valor encontrado, média das execuções, *gap* das médias em relação aos LBs e tempo médio de execução.

O método proposto obteve resultados competitivos com os da heurística proposta por Braekers et al. [2014] para os valores médios das execuções, com discreta melhora apresentada na média dos resultados. Apenas uma nova melhor solução foi encontrada. Para todas as outras instâncias, as duas abordagens se igualaram na melhor solução alcançada. Por fim, o algoritmo proposto obteve melhores tempos médios em todas as instâncias. As diferenças de tempo entre os métodos são mais evidentes nas instâncias de pequeno porte. Porém, as instâncias de médio e grande porte também foram resolvidas mais rapidamente. Os baixos tempos são atribuídos aos mecanismos de aceleração propostos neste trabalho.

Tabela 1: Resultados do HDARP

Instância	BKS	LB	DA					ILS-SP ^{HDARP}			
			Melhor	Média	Gap (%)	Tempo (s)	Tempo* (s)	Melhor	Média	Gap (%)	Tempo (s)
U											
a2-16	294,25	294,25	294,25	294,25	0,00	8,60	7,80	294,25	294,25	0,00	0,06
a2-20	344,83	344,83	344,83	344,83	0,00	20,20	19,80	344,83	344,83	0,00	0,08
a2-24	431,12	431,12	431,12	431,12	0,00	17,40	17,65	431,12	431,12	0,00	0,12
a3-18	300,48	300,48	300,48	300,48	0,00	9,40	7,30	300,48	300,48	0,00	0,11
a3-24	344,83	344,83	344,83	344,83	0,00	16,60	14,40	344,83	344,83	0,00	0,17
a3-30	494,85	494,85	494,85	494,85	0,00	18,80	18,10	494,85	494,85	0,00	0,23
a3-36	583,19	583,19	583,19	583,19	0,00	28,40	28,10	583,19	583,19	0,00	0,63
a4-16	282,68	282,68	282,68	282,68	0,00	9,80	6,90	282,68	282,68	0,00	0,09
a4-24	375,02	375,02	375,02	375,02	0,00	13,00	10,50	375,02	375,02	0,00	0,25
a4-32	485,50	485,50	485,50	485,50	0,00	25,60	21,70	485,50	485,50	0,00	0,38
a4-40	557,69	557,69	557,69	557,69	0,00	26,40	22,30	557,69	557,74	0,01	0,81
a4-48	668,82	668,82	668,82	668,82	0,00	35,40	33,50	668,82	668,82	0,00	2,80
a5-40	498,41	498,41	498,41	498,41	0,00	21,80	19,20	498,41	498,41	0,00	1,03
a5-50	686,62	686,62	686,62	686,62	0,00	30,60	29,60	686,63	686,63	0,00	3,95
a5-60	808,42	808,42	808,42	808,42	0,00	43,80	44,60	808,42	808,42	0,00	5,01
a6-48	604,12	604,12	604,12	604,12	0,00	30,60	27,00	604,12	604,12	0,00	2,23
a6-60	819,25	819,25	819,25	819,25	0,00	37,00	36,30	819,25	819,25	0,00	7,73
a6-72	916,05	916,05	916,05	916,05	0,00	53,60	54,10	916,05	916,56	0,06	13,22
a7-56	724,04	724,04	724,04	724,04	0,00	32,40	30,50	724,04	724,04	0,00	5,60
a7-70	889,12	889,12	889,12	889,30	0,02	43,60	42,90	889,12	889,21	0,01	11,57
a7-84	1033,37	1033,37	1033,37	1033,38	0,00	58,20	58,00	1033,37	1033,51	0,01	21,76
a8-64	747,46	747,46	747,46	747,46	0,00	37,80	36,20	747,46	747,46	0,00	11,95
a8-80	945,73	945,73	945,73	945,73	0,00	57,60	57,40	945,73	945,73	0,00	19,02
a8-96	1229,66	1229,66	1229,66	1232,03	0,19	66,40	65,60	1229,67	1230,44	0,06	35,75
Média	627,73	627,73	627,73	627,84	0,01	30,96	29,56	627,73	627,79	0,01	6,02
E											
a2-16	331,16	331,16	331,16	331,16	0,00	9,40	7,80	331,16	331,16	0,00	0,06
a2-20	347,03	347,03	347,03	347,03	0,00	19,60	19,20	347,03	347,03	0,00	0,08
a2-24	450,25	450,25	450,25	450,25	0,00	15,80	15,30	450,25	450,25	0,00	0,11
a3-18	300,63	300,63	300,63	300,63	0,00	9,60	7,20	300,63	300,63	0,00	0,10
a3-24	344,91	344,91	344,91	344,91	0,00	14,60	12,20	344,91	344,91	0,00	0,17
a3-30	500,58	500,58	500,58	500,58	0,00	17,00	16,00	500,58	500,58	0,00	0,24
a3-36	583,19	583,19	583,19	583,19	0,00	23,60	24,20	583,19	583,19	0,00	0,45
a4-16	285,99	285,99	285,99	285,99	0,00	8,20	5,90	285,99	285,99	0,00	0,09
a4-24	383,84	383,84	383,84	383,84	0,00	12,20	9,90	383,84	383,84	0,00	0,24
a4-32	500,24	500,24	500,24	500,24	0,00	22,80	20,70	500,25	500,25	0,00	0,37
a4-40	580,42	580,42	580,42	580,42	0,00	24,20	22,30	580,42	580,42	0,00	0,77
a4-48	670,52	670,52	670,52	670,52	0,00	33,60	32,70	670,52	670,52	0,00	2,13
a5-40	500,06	500,06	500,06	500,06	0,00	21,80	18,70	500,06	500,06	0,00	1,01
a5-50	693,77	693,77	693,77	693,77	0,00	28,00	26,30	693,77	693,77	0,00	3,00
a5-60	828,90	828,90	828,90	829,28	0,05	40,40	38,90	828,90	828,95	0,01	4,50
a6-48	614,36	614,36	614,36	614,36	0,00	29,40	25,80	614,36	614,36	0,00	2,49

Continua na página seguinte

Tabela 1: Resultados do HDARP (continuação)

Instância	BKS	LB	DA					ILS-SP ^{HDARP}			
			Melhor	Média	Gap (%)	Tempo (s)	Tempo* (s)	Melhor	Média	Gap (%)	Tempo (s)
E											
a6-60	847,58	847,58	847,58	848,48	0,11	41,60	38,60	847,58	847,94	0,04	9,02
a6-72	949,17	949,17	949,17	949,17	0,00	55,40	54,60	949,17	949,54	0,04	12,91
a7-56	740,63	740,63	740,63	740,63	0,00	32,80	29,70	740,63	740,63	0,00	4,52
a7-70	946,32	946,32	946,32	946,32	0,00	46,60	41,80	946,32	946,32	0,00	10,93
a7-84	1092,90	1092,90	1092,90	1092,93	0,00	57,20	56,20	1092,90	1093,43	0,05	23,18
a8-64	762,81	762,81	762,81	762,81	0,00	39,00	34,90	762,81	762,81	0,00	8,44
a8-80	982,71	982,71	982,71	982,71	0,00	57,40	57,30	982,71	982,71	0,00	13,56
a8-96	1265,36 ¹	1265,36	1265,54	1266,78	0,11	63,00	63,80	1265,36	1266,04	0,05	34,53
Média	645,98	645,97	645,98	646,09	0,01	30,13	28,33	645,97	646,05	0,01	5,54
I											
a2-16	294,25	294,25	294,25	294,25	0,00	7,20	6,60	294,25	294,25	0,00	0,07
a2-20	355,74	355,74	355,74	355,74	0,00	17,40	17,50	355,74	355,74	0,00	0,10
a2-24	431,12	431,12	431,12	431,12	0,00	12,60	12,90	431,12	431,12	0,00	0,15
a3-18	302,17	302,17	302,17	302,17	0,00	8,40	6,60	302,17	302,17	0,00	0,13
a3-24	344,83	344,83	344,83	344,83	0,00	13,40	12,00	344,83	344,83	0,00	0,24
a3-30	494,85	494,85	494,85	494,85	0,00	14,80	14,00	494,85	494,85	0,00	0,33
a3-36	618,15	618,15	618,15	618,59	0,07	22,60	23,40	618,15	618,15	0,00	3,97
a4-16	299,05	299,05	299,05	299,05	0,00	7,20	5,30	299,05	299,05	0,00	0,09
a4-24	375,02	375,02	375,02	375,02	0,00	12,00	10,30	375,02	375,02	0,00	0,29
a4-32	486,93	486,93	486,93	487,50	0,12	21,00	19,60	486,93	486,93	0,00	0,58
a4-40	557,69	557,69	557,69	557,69	0,00	23,80	22,50	557,69	557,84	0,03	1,40
a4-48	670,72	670,72	670,72	670,72	0,00	30,00	29,40	670,72	670,72	0,00	4,61
a5-40	507,18	507,18	507,18	507,18	0,00	19,20	17,70	507,18	507,18	0,00	1,92
a5-50	690,99	690,99	690,99	690,99	0,00	27,80	27,10	690,99	691,45	0,07	5,08
a5-60	816,15	816,15	816,15	816,81	0,08	39,00	39,50	816,15	816,15	0,00	7,74
a6-48	604,12	604,12	604,12	604,12	0,00	29,00	26,90	604,12	604,12	0,00	2,64
a6-60	829,23	829,23	829,23	829,26	0,00	34,20	35,00	829,23	829,23	0,00	9,22
a6-72	938,46	938,46	938,46	939,32	0,09	48,60	48,30	938,46	938,81	0,04	13,47
a7-56	727,20	727,20	727,2	727,20	0,00	29,20	28,60	727,20	727,28	0,01	7,75
a7-70	925,39	923,19 ²	925,39	927,60	0,48	39,20	38,60	925,38	925,77	0,28	13,11
a7-84	1037,04	1037,04 ²	1037,04	1037,88	0,08	54,00	54,70	1037,05	1037,24	0,02	21,78
a8-64	748,04	748,04	748,04	748,04	0,00	37,20	34,10	748,04	748,29	0,03	12,45
a8-80	969,87	968,84 ²	969,87	970,83	0,21	50,20	47,90	968,84	969,83	0,10	18,85
a8-96	1237,89	1233,86 ²	1237,89	1237,96	0,33	61,80	60,00	1237,89	1238,43	0,37	31,87
Média	635,92	635,62	635,92	636,20	0,06	27,49	26,60	635,88	636,02	0,06	6,58

¹ Resultado apresentado em Masmoudi et al. [2018]

² Novos LBs disponibilizados pelo algoritmo B&C proposto por Braekers et al. [2014]

6. Conclusão

Este trabalho propôs um método baseado na meta-heurística ILS combinada a procedimentos exatos para resolver o HDARP. Para avaliar a performance do algoritmo, foram utilizadas instâncias da literatura. Quanto as médias das qualidades das soluções, o algoritmo proposto alcançou resultados similares com uma pequena melhora geral. Uma nova melhor solução foi encontrada e todas as outras melhores soluções de outros métodos foram alcançadas. Por último, em virtude dos novos mecanismos de aceleração apresentados e checagens eficientes para as violações de restrições, o algoritmo foi capaz de resolver todas as instâncias em tempos menores do que os demais trabalhos relacionados ao problema. Em média, o algoritmo foi até 5 vezes mais rápido em relação ao método de referência quando executados na mesma máquina.

Para trabalhos futuros, pretende-se aplicar o algoritmo para o MDHDARP, além de aplicar o método proposto às instâncias tradicionais do DARP. Outra linha seria utilizar a checagem eficiente para violação da restrição de tempo máximo de viagem proposta por Gschwind e Drexl [2019].

Referências

- Beek, O., Raa, B., Dullaert, W., e Vigo, D. (2018). An efficient implementation of a static move descriptor-based local search heuristic. *Computers & Operations Research*, 94:1–10.
- Braekers, K., Caris, A., e Janssens, G. K. (2014). Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, 67:166–186.
- Bulhões, T., Subramanian, A., Erdoğan, G., e Laporte, G. (2018). The static bike relocation problem with multiple vehicles and visits. *European Journal of Operational Research*, 264(2):508–523.
- Cordeau, J.-F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54(3):573–586.
- Cordeau, J.-F. e Laporte, G. (2003a). The dial-a-ride problem (darp): Variants, modeling issues and algorithms. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(2):89–101.
- Cordeau, J.-F. e Laporte, G. (2003b). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579–594.
- Gschwind, T. e Drexl, M. (2019). Adaptive large neighborhood search with a constant-time feasibility test for the dial-a-ride problem. *Transportation Science*, 53(2):480–491.
- Henao, A. (2017). *Impacts of Ridesourcing-Lyft and Uber-on Transportation Including VMT, Mode Replacement, Parking, and Travel Behavior*. University of Colorado at Denver.
- Masmoudi, M. A., Hosny, M., Demir, E., Genikomsakis, K. N., e Cheikhrouhou, N. (2018). The dial-a-ride problem with electric vehicles and battery swapping stations. *Transportation research part E: logistics and transportation review*, 118:392–420.
- Parragh, S. N. (2011). Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part C: Emerging Technologies*, 19(5): 912–930.
- Savelsbergh, M. W. (1985). Local search in routing problems with time windows. *Annals of Operations research*, 4(1):285–305.
- Subramanian, A., Drummond, L. M. d. 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.
- Subramanian, A., Uchoa, E., e Ochi, L. S. (2013). A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, 40(10):2519–2531.
- Vidal, T., Crainic, T. G., Gendreau, M., e Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & operations research*, 40(1):475–489.
- Zachariadis, E. E. e Kiranoudis, C. T. (2010). A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. *Computers & operations research*, 37(12):2089–2105.