LIII Simpósio Brasileiro de Pesquisa Operacional
João Pessoa - PB, 3 a 5 de novembro de 2021

Abordagem exata para duas variantes do problema de roteamento de veículos com coleta e entrega simultâneas

Rafael Praxedes, Teobaldo Bulhões, Anand Subramanian
Universidade Federal da Paraíba
Centro de Informática, Rua dos Escoteiros, s/n, Mangabeira, 58055-000, João Pessoa, PB, Brasil
rmrpraxedes@gmail.com, tbulhoes@ci.ufpb.br, anand@ci.ufpb.br

Eduardo Uchoa
Universidade Federal Fluminense
Departamento de Engenharia de Produção, Rua Passo da Pátria, 156, Campus Praia Vermelha,
Bloco D - sala 306, São Domingos, 24210-240, Niterói, RJ, Brasil
eduardo.uchoa@gmail.com

RESUMO

O Problema de Roteamento de Veículos (PRV) consiste em determinar um conjunto de rotas de menor custo, iniciando e finalizando no depósito, de forma que as demandas de cada cliente sejam satisfeitas em uma única visita. O PRV com Coleta e Entrega Simultâneas (PRVCES) generaliza o PRV pela inclusão de uma demanda de coleta (entrega) adicional. Este trabalho visa propor uma abordagem exata baseada em geração de cortes e colunas para resolver duas variantes do PRVCES: uma envolvendo frota heterogênea e outra com múltiplos depósitos. Tal abordagem está relacionada a utilização do *VRPSolver*, que consiste em um resolvedor de *branch-cut-and-price* aplicado a problemas de roteamento e afins. Os resultados obtidos mostram que a abordagem proposta é capaz de encontrar soluções ótimas e melhorar os limites para muitas instâncias da literatura em aberto.

PALAVRAS CHAVE. Roteamento de veículos. Coleta e entrega simultâneas. Abordagem exata.

L&T – Logística e Transportes. OC – Otimização Combinatória.

ABSTRACT

The Vehicle Routing Problem (VRP) consists in determining a set of least-cost vehicle routes, starting and ending at the depot, such that the demand of each customer is satisfied in a single visit. The VRP with Simultaneous Pickup and Delivery (VRPSPD) generalizes the VRP by including an additional pickup (delivery) demand. This work aims at proposing an exact approach based on cut and column generation to solve two VRPSPD variants: one involving heterogeneous fleet and another one with multiple depots. This approach uses the VRPSolver, which is a branch-cut-and-price solver for routing and similar problems. The results obtained show that the proposed approach is capable of finding optimal solutions and improved bounds for many open benchmark instances.

KEYWORDS. Vehicle routing. Simultaneous pickup and delivery. Exact approach.

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

1. Introdução

O *Problema de Roteamento de Veículos* (PRV) é um problema de otimização combinatória clássico, sendo bastante estudado desde quando proposto por Dantzig e Ramser [1959]. Em conformidade com Irnich et al. [2014], o PRV pode ser definido como um problema cujo objetivo consite em, dados um conjunto de clientes com suas respectivas demandas e uma frota de veículos, determinar quais veículos atenderão quais clientes, bem como a sequência de atendimento, de modo que todos eles sejam atendidos uma única vez e que o custo total de transporte seja mínimo.

Desde a publicação dos primeiros trabalhos, diversas variantes para o PRV foram propostas [Braekers et al., 2016]. Em particular, destacam-se os Problemas de Coleta e Entrega (PCE) que, de acordo com Berbeglia et al. [2007], são uma importante classe do PRV, a qual lida com o transporte de pessoas e objetos, seja coletando-os e/ou distribuindo-os. Nesse sentido, o PRV com Coleta e Entrega Simultâneas (PRVCES) é uma vertente dos PCE, proposta por Min [1989], na qual as tarefas de coleta e entrega são realizadas simultaneamente. Além disso, está incluso nos PCE, o PRV com Coleta e Entrega Mistas (PRVCEM), que se diferencia do primeiro pelo fato dos clientes possuírem apenas demandas de entrega ou de coleta, contudo, ambos os tipos de cliente estão inclusos em uma mesma rota. Apesar dessa diferença, Salhi e Nagy [1999] afirmam que o PRVCES e o PRVCEM podem ser modelados em um mesmo *framework* e, por essa razão, serão analisados conjuntamente neste artigo. Mais precisamente, serão consideradas duas variantes desses problemas, as quais incluem Frota Heterogênea (PRVCESFH) e Múltiplos Depósitos (PRVCESMD ou PRVCEMMD).

Em relação ao PRVCESFH, Qu e Bard [2013] lidaram com uma variante do PCE denominada, em Inglês, *Heterogeneous Pickup and Delivery Problem (HPDP) with multiple request types and configurable vehicle capacity*. Esta engloba o uso de uma frota heterogênea com capacidade dos veículos configurável, aplicada ao planejamento das rotas diárias do *Program of All-Inclusive Care for the Elderly* (PACE), o qual consiste em um programa estabelecido pela Medicare para atendimento médico de idosos com mais de 55 anos de idade, sendo uma alternativa econômica para essas pessoas que necessitam de cuidados especiais em seus lares. Para esse problema, foram propostos um modelo de programação inteira mista, bem como um algoritmo *Multi-Start Adaptive Large Neighborhood Search* (MSALNS) na busca por soluções. Os experimentos foram realizados com as instâncias do Dial-a-Ride Problem (DARP) utilizadas por Parragh [2011], como também com instâncias reais providas pela PACE, em Wichita, Kansas. Além disso, foram propostos dois novos conjuntos de instâncias geradas randomicamente para o problema. Por fim, como resultados, uma redução nos custos de até 40% pôde ser alcançada. Por sua vez, Qu e Bard [2015] lidaram com um problema bastante similar ao HPDP com capacidade dos veículos configurável, contudo com o uso de janelas de tempo mais restritivas. Motivados também pelas atividades do PACE, propuseram um modelo de programação inteira mista para o problema e um método *Branch-Cut-and-Price* (BCP) para resolvê-lo. Além disso, importante ressaltar que foram utilizadas *subset row inequalities* [Jepsen et al., 2008], para o fortalecimento dos limites inferiores (LIs) encontrados pela geração de colunas e condições de dominância, a fim de acelerar os tempos de execução. Durante os experimentos computacionais, foram utilizadas instâncias reais e outras geradas randomicamente com até 50 clientes, sendo obtidas soluções ótimas para a maioria delas dentre os resultados.

O PRVCESFH também foi abordado por Avci e Topaloglu [2016], os quais propuseram uma formulação matemática para o problema (v. Seção 2), bem como um algoritmo de busca local híbrido (em inglês, *Hybrid Local Search* - HLS) que combina um mecanismo de ajuste de limiar adaptativo (*Threshold adjusting mechanism*) com a meta-heurística Busca Tabu (*Tabu Search*), considerando uma frota ilimitada de veículos. Importante ressaltar que foi utilizada uma estratégia de

modo que o algoritmo seja auto-ajustável. Além disso, validaram o método proposto com dois conjuntos de instâncias geradas randomicamente, as quais podem chegar até 550 clientes. Dentre os resultados obtidos, o HLS se mostrou bastante eficiente na resolução das instâncias testadas, quando comparado a outros métodos anteriormente propostos, encontrando soluções de boa qualidade em curtos períodos computacionais. Vale ressaltar que essa diferença entre os métodos se tornava mais evidente a medida em que crescia o tamanho do problema. Por sua vez, Nepomuceno et al. [2019] propuseram um algoritmo randomizado com estratégia de busca pelo vizinho mais próximo (em inglês, *Nearest-Neighbor-Based Randomized Algorithm*) para o PRVCESFH. Para a validação deste, utilizaram os mesmos conjuntos de instâncias de Avci e Topaloglu [2016] e obtiveram resultados competitivos quando comparado ao HLS, em tempos computacionais curtos, especialmente para aquelas consideradas maiores, ou seja, com mais de 100 clientes.

O PRVCEMMD foi introduzido por Salhi e Nagy [1999], os quais conceituaram a diferença entre PRVCES e PRVCEM. Como método proposto, desenvolveram uma heurística de inserção em *cluster*, resolvendo ambos os problemas, incluindo também adaptações para o caso com múltiplos depósitos (definindo assim o PRVCEMMD ou PRVCESMD). Propuseram também um conjunto de 14 instâncias, contendo 1 depósito e variando a quantidade de clientes entre 50 e 199, bem como outro conjunto de 11 instâncias, as quais variam entre 2 e 5 depósitos e 50 e 249 clientes. Apesar da heurística desenvolvida não ter apresentado resultados substancialmente melhores, foram obtidas pequenas melhorias a um custo computacional baixo. Anos mais tarde, Nagy e Salhi [2005] propuseram a criação de uma abordagem heurística composta para os mesmos problemas, incluindo as variantes com mais de um depósito, a qual encontrava soluções viáveis para o PRV e, então, aplicava técnicas de melhoramento, a fim de torná-las viáveis para o PRVCES (ou PRVCEM). Durante os experimentos computacionais, foram utilizadas as mesmas instâncias de Salhi e Nagy [1999], com as heurísticas desenvolvidas sendo capazes de encontrar soluções de boa qualidade, superando, no geral, os resultados já conhecidos na época.

Por sua vez, Ropke e Pisinger [2006] desenvolveram um *framework* para que pudessem ser resolvidas diversas variantes do PRV com *Backhauls*, incluindo o PRVCEMMD. Tais variantes são transformadas em um caso do *Rich Pickup and Delivery Problem with Time Windows* (*Rich* PDPTW), uma vez que, segundo os autores, todas as variantes consideradas por eles podem ser vistas como extensões do PDPTW. Após a transformação, são resolvidas por meio de uma heurística *Large Neighborhood Search* (LNS) e, então, adaptadas ao problema original. Durante os experimentos computacionais, foram realizados testes com nove conjuntos de instâncias presentes na literatura, sendo obtidas soluções melhores para 67% dos problemas testados.

Continuando a análise, Li et al. [2015] desenvolveram uma heurística baseada no *Iterated Local Search* com um mecanismo de seleção de vizinhança adaptativo, em inglês *Adaptive Neighborhood Selection*, (ILS_ANS) para lidar com o PRVCESMD. Além disso, propuseram novos métodos de perturbação de vizinhança. Para a validação do método desenvolvido, utilizaram alguns dos conjuntos de instâncias propostos por Salhi e Nagy [1999], obtendo resultados bem competitivos. Os autores ressaltam ainda que o método proposto apresentou um melhor desempenho, se comparado a outras abordagens utilizadas para o problema. Já Koulaeian et al. [2015] construíram um modelo matemático para o PRVCESMD incluindo frota heterogênea. Tinham como objetivo a minimização do custo de roteamento, aplicando penalidades em caso de descumprimento dos tempos limite associados às rotas e considerando os custos fixos associados aos motoristas. Para resolver o problema, foram desenvolvidas duas meta-heurísticas baseadas no *Imperialist Competitive Algorithm* (ICA) e no *Genetic Algorithm* (GA). Foram geradas instâncias próprias e, como resultado, percebeu-se que o ICA apresentou um melhor desempenho quando comparado ao GA,

especialmente para aquelas de maior tamanho.

Por fim, Subramanian [2012] fez uma análise completa do PRV e algumas de sua variantes, dentre as quais está presente o PRVCEMMD. Tal análise compreendia abordagem exatas, heurísticas e híbridas para os problemas. Considerando especificamente a variante em questão, foi proposto um algoritmo de *Branch-and-Cut* (B&C) baseado em uma formulação matemática definida apenas sobre variáveis de arestas, na qual as restrições de verificação da carga do veículo exceder sua capacidade, como também aqueles que garantem início e término de rotas no mesmo depósito, são inseridas de forma *lazy* (*lazy constraints*). Além da abordagem exata, um método baseado na meta-heurística *Iterated Local Search* (ILS) com o *Random Variable Neighborhood Descent* (RVND) e um algoritmo híbrido que combina o ILS-RVND com uma técnica baseada em Particionamento de Conjuntos foram aplicados ao PRVCEMMD. Todas essas abordagens implicaram a obtenção de resultados bastante competitivos.

Em face do exposto, o objetivo do presente artigo é propor uma abordagem exata para lidar com o PRVCESFH e o PRVCEMMD, a fim de que possam ser encontradas soluções ótimas para instâncias em aberto na literatura ou melhorados os LIs já conhecidos, utilizando o *VRPSolver* para tal finalidade. Este, por sua vez, foi proposto por Pessoa et al. [2020], e consiste em um *solver* de BCP para modelos genéricos, com ênfase no PRV e suas variantes. Tal *solver* se destaca por empregar os melhores elementos presentes nos algoritmos de PRV que compõem a literatura, a saber: relaxação *ng-path*, corte *rank-1* com memória limitada, enumeração de caminho e cortes de capacidade arredondada. Todos esses elementos estão implementados no que os autores definem como *packing sets*. Importante ressaltar que, no algoritmo de BCP utilizado pelo *solver*, os subproblemas de *pricing* são modelados como um problema de caminho mais curto com restrição de recursos. Desse modo, além dos modelos definidos pelo usuário para o *solver* — os quais serão apresentados, na Seção 3, para as variantes do PRV em questão — são definidos também grafos, com seus respectivos recursos e elementos associados. As soluções para o modelo implementado são representadas por caminhos nesses grafos, uma vez que suas variáveis (ou parte delas) são mapeadas para os seus arcos, sendo o modelo como um todo para o problema resolvido pelo algoritmo de BCP.

O restante deste artigo está estruturado da seguinte forma: na Seção 2, são definidos o PRVCESFH e o PRVCEMMD; na Seção 3 são abordados os modelos e demais elementos inerentes à abordagem com o *VRPSolver* para as variantes do PRV em questão; na Seção 4 são descritos os experimentos computacionais realizados, com seus respectivos resultados. Por fim, na Seção 5, são feitas algumas considerações finais acerca do trabalho desenvolvido.

2. Definição dos problemas

Nesta seção serão apresentadas as duas variantes do problema de roteamento de veículos com coleta e entrega simultâneas que são abordadas neste artigo. O PRVCESFH pode ser definido da seguinte forma: sejam $V = \{v_1, ..., v_n\}$ o conjunto de clientes; $V_0 = V \cup \{v_0\}$ o conjunto de clientes mais o depósito e $E = \{\{v_i, v_j\} : v_i \in V_0, v_j \in V_0, v_i < v_j\}$ o conjunto de arestas entre os clientes e entre eles e o depósito. Além disso, c_{ij} representa a distância entre os pontos v_i e v_j, $\forall v_i, v_j \in V_0$ e p_j e d_j as demandas de coleta e entrega do cliente v_j, $\forall v_j \in V$, respectivamente. Quanto aos veículos, estes estão representados pelo conjunto $K = \{1, ..., m\}$, os quais, por serem heterogêneos, possuem capacidades q_k e custos fixos fc_k e variáveis vc_k, $\forall k \in K$ associados a cada um deles. Por fim, seja $A = \{(v_i, v_j), (v_j, v_i) : \{v_i, v_j\} \in E\}$ o conjunto de arcos correspondentes a cada uma das arestas definidas por E.

Com este problema, objetiva-se a construção de um conjunto com no máximo m rotas, minimizando-se o custo total (distância total percorrida, bem como os custos fixos e variáveis associados a cada um dos veículos que compõem a frota), de modo que cada uma delas seja iniciada e

finalizada no depósito, que todas as demandas de coleta e entrega dos clientes sejam satisfeitas, que as capacidades dos veículos não sejam excedidas em qualquer ponto da rota (considerando a carga coletada e a que ainda será entregue, simultaneamente) e, por fim, que cada um dos clientes seja visitado apenas uma vez. Na formulação do PRVCESFH, as seguintes variáveis são utilizadas:

- $x_{ij}^k = \begin{cases} 1, & \text{se o arco } (v_i, v_j) \in A, \text{ percorrido pelo veículo } k \in K \text{ pertence à solução} \\ 0, & \text{c.c.} \end{cases}$
- P_{ij}^k, D_{ij}^k: Variáveis de fluxo que indicam a carga de coleta e entrega, respectivamente, contidas no veículo $k \in K$ ao trafegar pelo arco $(v_i, v_j) \in$ A;

Adiante será abordada a formulação matemática para o PRVCESFH apresentada por Avci e Topaloglu [2016] e inspirada no modelo proposto por Montané e Galvão [2006] para o PRVCES clássico.

$$\min \sum_{k \in K} \sum_{v_j \in V} fc_k x_{0j}^k + \sum_{k \in K} \sum_{v_i \in V_0} \sum_{v_j \in V_0 \setminus \{v_i\}} vc_k c_{ij} x_{ij}^k \tag{1}$$

$$\text{s.a.} \sum_{k \in K} \sum_{v_j \in V_0 \setminus \{v_i\}} x_{ij}^k = 1, \qquad \forall v_i \in V \tag{2}$$

$$\sum_{v_i \in V_0 \setminus \{v_p\}} x_{ip}^k - \sum_{v_j \in V_0 \setminus \{v_p\}} x_{pj}^k = 0, \qquad \forall v_p \in V_0, \forall k \in K \tag{3}$$

$$\sum_{v_j \in V} x_{0j}^k \leq 1 \qquad \forall k \in K \tag{4}$$

$$P_{0j}^k = 0 \qquad \forall v_j \in V, \forall k \in K \tag{5}$$

$$D_{i0}^k = 0 \qquad \forall v_i \in V, \forall k \in K \tag{6}$$

$$\sum_{k \in K} \sum_{v_j \in V_0 \setminus \{v_i\}} D_{ji}^k - \sum_{k \in K} \sum_{v_j \in V_0 \setminus \{v_i\}} D_{ij}^k = d_i, \qquad \forall v_i \in V \tag{7}$$

$$\sum_{k \in K} \sum_{v_j \in V_0 \setminus \{v_i\}} P_{ij}^k - \sum_{k \in K} \sum_{v_j \in V_0 \setminus \{v_i\}} P_{ji}^k = p_i, \qquad \forall v_i \in V \tag{8}$$

$$\sum_{k \in K} \sum_{v_i \in V} P_{i0}^k = \sum_{v_i \in V} p_i \tag{9}$$

$$\sum_{k \in K} \sum_{v_j \in V} D_{0j}^k = \sum_{v_j \in V} d_j \tag{10}$$

$$D_{ij}^k + P_{ij}^k \leq q_k x_{ij}^k \qquad \forall (v_i, v_j) \in A, k \in K \tag{11}$$

$$D_{ij}^k \geq 0, \qquad \forall (v_i, v_j) \in A, k \in K \tag{12}$$

$$P_{ij}^k \geq 0, \qquad \forall (v_i, v_j) \in A, k \in K \tag{13}$$

$$x_{ij}^k \in \{0, 1\}, \qquad \forall (v_i, v_j) \in A, k \in K \tag{14}$$

A função objetivo (FO) (1) busca minimizar o custo total, o qual engloba a distância total percorrida, bem como os custos fixo e variável de cada um dos veículos. Os conjuntos de restrições (2) e (3) garantem que cada cliente será visitado apenas um vez, sendo análogo afirmar que será

visitado por um único veículo k. As restrições (4) permitem a utilização de um determinado veículo k. As restrições (5) e (6) indicam que a cargas de coleta do veículo k ao sair do depósito e a carga de entrega do mesmo veículo ao chegar no depósito devem ser nulas, respectivamente. As restrições (7) e (8) representam as equações de fluxo para as demandas de entrega e coleta dos clientes, respectivamente, implicando que ambas devem ser satisfeitas para todos os clientes. As restrições (9) e (10) estabelecem que a carga total coletada pelos veículos ao chegarem no depósito e a carga total a ser entregue pelos mesmos veículos ao saírem do depósito devem ser iguais ao somatório das demandas de coleta e entrega de todos os clientes, respectivamente. As restrições (11) garantem que as demandas de coleta e entrega deverão ser transportadas por meio de arcos e veículos que compõem a solução e que as mesmas não devem exceder a capacidade q_k, referente ao veículo k responsável por transportá-las. Por fim, as restrições (12)-(14) definem a natureza das variáveis do problema.

Por sua vez, o PRVCEMMD pode ser definido de forma semelhante ao PRVCESFH, exceto pelo fato de que, agora, passa a ser considerada a existência de múltiplos depósitos, a partir dos quais as rotas podem ser iniciadas e finalizadas, bem como uma frota homogênea. Apesar disso, o depósito do qual uma rota é iniciada deve ser visitado ao final da mesma. Outra importante peculiaridade está relacionada às demandas dos clientes. Nessa variante, tais clientes possuem apenas demandas de entrega ou coleta, contudo, ambos os tipos de cliente coexistem. Isto é, se $p_i > 0$, para algum cliente v_i, então d_i = 0. Analogamente, sendo $d_i > 0$, então $p_i = 0$.

Segundo Baldacci e Mingozzi [2009], qualquer Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD) pode ser transformado em uma instância do Problema de Roteamento de Veículos com Frota Heterogênea (PRVFH). Desse modo, a mesma formulação do PRVCESFH pode ser utilizada para a variante em questão, desde que se considere as seguintes adaptações: seja K = {1, ..., p} o conjunto de todos os depósitos. Logo, como K não representa mais a frota de veículos, os quais deixam de ser heterogêneos e passam a ser homogêneos, tem-se que $q_k = Q$. Tal homogeneidade também influencia nos custos associados aos veículos, ou seja, $fc_k = 0$ e $vc_k = 1$, pois os depósitos não possuem custos específicos, sendo relevante apenas a minimização da distância total percorrida. Em relação aos custos do problema, estes estão definidos por (15), sendo c_{n+kj} e c_{in+k} o custo entre o depósito $k \in K$ e o cliente v_j e entre o cliente v_i e o depósito $k \in K$, respectivamente, em que n representa a quantidade de clientes. Uma vez definido c_{ij}^k, estabelece-se a FO do PRVCEMMD, a qual está representada em (16).

$$c_{ij}^k = \begin{cases} c_{ij}, & \text{se } v_i \neq 0 \text{ e } v_j \neq 0 \\ c_{n+kj}, & \text{se } v_i = 0 \text{ e } v_j \neq 0 \\ c_{in+k}, & \text{se } v_i \neq 0 \text{ e } v_j = 0 \end{cases} \tag{15}$$

$$\min \sum_{k \in K} \sum_{v_i \in V_0} \sum_{v_j \in V_0} c_{ij}^k x_{ij}^k \tag{16}$$

Portanto, afirma-se que uma formulação para o PRVCEMMD pode ser composta pela FO expressa em (16), cujos custos estão descritos em (15), acrescida das restrições apresentadas em (2)-(14), levando-se em consideração as adaptações explicadas.

3. Abordagem exata utilizando o *VRPSolver*

Nesta seção, serão apresentadas as modelagens do PRVCESFH e do PRVCEMMD utilizando o *VRPSolver* [Pessoa et al., 2020]. Nas descrições que seguem, serão utilizados conceitos e terminologias definidas na linguagem de modelagem do *VRPSolver*. Dadas as limitações de espaço

e questões de escopo, esses elementos não serão detalhados neste trabalho, devendo o leitor consultar o trabalho de Pessoa et al. [2020] para um completo entendimento dos modelos.

Quanto à primeira variante, define-se: sejam V o conjunto de clientes; $V_0 = V \cup \{v_0\}$, o conjunto de clientes mais o depósito; n o número total de clientes, $n = |V|$; c_{ij} a distância entre os clientes v_i e v_j, $\forall v_i, v_j \in V_0$; p_j e d_j as demandas de coleta e entrega do cliente v_j, $\forall v_j \in V$, respectivamente; K o conjunto de todos os tipos de veículo e, por fim, q_k, fc_k e vc_k a capacidade, custos fixo e variável associados aos veículos do tipo k, respectivamente. É válido ressaltar que tais definições muito se assemelham àquelas associadas ao modelo de Programação Inteira Mista (*Mixed Integer Programming*, MIP) correspondente ao PRVCESFH (v. Seção 2). No entanto, a principal diferença consiste na redefinição do conjunto K, o qual deixa de representar o conjunto de todos os veículos disponíveis e passa a ser o conjunto dos tipos de veículos que constituem a frota. Essa alteração é fundamental para viabilizar a maneira como os grafos serão definidos, conforme será visto ainda nesta seção.

Dando continuidade às definições, sejam $G^k = (V_0^k, A^k), \forall k \in K$ os grafos associados ao modelo para o *VRPSolver*, nos quais $V_0^k = \{v_0^k, v_1^k, ..., v_n^k\}$ representa o conjunto de todos os vértices; $V^k = \{v_1^k, ..., v_n^k\}$ o conjunto de todos os clientes e, por fim, $A^k = \{(v_i^k, v_j^k)$: $v_i^k \in V_0^k$, $v_j^k \in V_0^k$, $v_i^k \neq v_j^k$ $\}$ o conjunto dos arcos que compõem esse grafo. Aos vértices especiais de origem e destino associados a cada um desses grafos, tem-se as seguintes atribuições: $V_{source}^k = V_{sink}^k = v_0^k$. Diante disso, percebe-se a existência de $|K|$ grafos, cada um deles associado a um tipo de veículo k. Essa é a justificativa para a redefinição do conjunto K. Outro importante ponto a ser observado é que, apesar de terem sido definidos $|K|$ conjuntos de vértices, referentes a cada grafo, estes consistem em cópias do único conjunto V_0 definido no início desta seção.

Quanto aos recursos, sejam $R = R_M = \{1, 2\}$ recursos principais e *disposables* associados às demandas de coleta e entrega dos clientes, respectivamente. Assim, o consumo de tais recursos pode ser estabelecido como $q_{a^k,1} = p_j$ e $q_{a^k,2} = d_j$, $\forall a^k = (v_i^k, v_j^k) \in A^k$, $\forall k \in K$, sendo $q_{a^k,1}$ e $q_{a^k,2}$ as representações do consumo de tais recursos no arco a^k, respectivamente, e p_j e d_j as demandas de coleta e entrega, respectivamente, referentes ao vértice v_j^k. Além disso, o consumo acumulado de tais recursos deve respeitar os seguintes limites: sejam $l_{v_i^k,1}$ e $l_{v_i^k,2}$ as quantidades mínimas dos recursos 1 e 2, respectivamente, que devem ter sido consumidas no momento em que o vértice v_j^k é visitado. O mesmo ocorre para as quantidades máximas de consumo, as quais são representadas por $u_{v_i^k,1}$ e $u_{v_i^k,2}$. Desse modo, tem-se $[l_{v_i^k,1}, u_{v_i^k,1}] = [l_{v_i^k,2}, u_{v_i^k,2}] = [0, q^k]$, na qual q^k representa a capacidade dos veículos do tipo k.

Por fim, são necessárias à formulação as variáveis inteiras $x_{a^k}, \forall a^k \in A^k, \forall k \in K$. Com isso, pode-se renomear os custos c_{ij} como $c_{a^k}, \forall a^k = (v_i^k, v_j^k) \in A^k, \forall k \in K$, a fim de combinar com a notação utilizada para as variáveis do modelo. Em face das definições apresentadas, tem-se a seguinte formulação para o PRVCESFH utilizando o *VRPSolver*:

$$\min \sum_{k \in K} \sum_{a^k \in A^k,\ v_i^k = v_0^k} fc_k x_{a^k} + \sum_{k \in K} \sum_{a^k \in A^k} vc_k c_{a^k} x_{a^k} \tag{17}$$

$$\text{s.a.} \sum_{k \in K} \sum_{a^k \in \delta^-(v_i^k)} x_{a^k} = 1, \qquad \forall v_i^k \in V^k \tag{18}$$

$[L^k, U^k] = [0, n], \forall k \in K;\ \ M(x_{a^k}) = \{a^k\}, \forall a^k \in A^k, \forall k \in K;\ \ \mathcal{B}^{\mathcal{V}} = \cup_{v_i^k \in V^k} \{\{v_i^k : k \in K\}\}.$

Novamente, tem-se como FO (17) a minimização da distância total percorrida, levando-se em consideração os custos fixos e variáveis associados a cada tipo de veículo. As restrições (18) são úteis na determinação de que cada um dos clientes pode ser visitado apenas uma vez, isto é, por um único veículo. Quanto aos limites inferior L^k e superior U^k do número de rotas associadas a cada um dos $|K|$ grafos, atribui-se 0 e n, uma vez que não é obrigatório o uso de todos os tipos de veículo disponíveis e a frota não é limitada, respectivamente. Por essa razão, pode-se limitar o número máximo de rotas ao total de clientes, pois é de conhecimento prévio que a rota mais custosa possível é aquela na qual cada cliente é atendido por um veículo próprio. Quanto aos mapeamentos, estabelece-se que cada variável x_{a^k} está associada ao arco $a^k, \forall a^k \in A^k, \forall k \in K$. Em relação aos *packing sets*, cada um desses conjuntos é composto por todos os vértices que representam um único cliente v_i, isto é, considerando as representações do mesmo em cada um dos grafos G^k. Por fim todas as demais restrições do modelo MIP associado ao PRVCESFH são implicitamente satisfeitas com as definições dos grafos, recursos, limites, mapeamentos e *packing sets*.

Um importante ressalva deve ser feita apenas em relação ao fato das cargas de coleta e entrega contidas simultaneamente em um veículo não poderem exceder a capacidade do mesmo, o que não está garantido pelo modelo apresentado. Para tanto, utiliza-se o artifício das *lazy constraints*. Esse artifício consiste no acréscimo de restrições ao modelo que impeçam a existência de rotas inviáveis em relação a um aspecto. No caso em questão, o aspecto consiste na existência concomitante das cargas de coleta e entrega em qualquer ponto da rota. Cada uma dessas restrições será adicionada à formulação original no momento em que uma inviabilidade for encontrada e, assim, a rota que a contém não será permitida. As expressões matemáticas referentes a essas restrições são indicadas em (19), na qual $\mathcal{R}$ representa o conjunto de todas as rotas que são viáveis em relação às demandas de coleta e entrega isoladamente, mas não considerando as mesmas simultaneamente. A fim de que as restrições possam ser adicionadas, funções *callbacks* são utilizadas, as quais são responsáveis por, a cada solução inteira encontrada, verificar a viabilidade das rotas, adicionando tal restrição (*lazy constraint*) se necessário.

$$\sum_{(i,j)\in R} x_{ij} \leq |R| - 1, \qquad \forall R \in \mathcal{R} \tag{19}$$

Por sua vez, a implementação do PRVCEMMD utilizando o *VRPSolver* difere em alguns aspectos, se comparada à formulação apresentada para o PRVCESFH. Continuam sendo válidas as definições do conjunto V e das constantes n, p_j e d_j, $\forall v_i, v_j \in V$. A primeira diferença está relacionada à definição do conjunto V_0, o qual passa a englobar todos os clientes, bem como todos os depósitos disponíveis, ou seja, $V_0 = V \cup \{v_{n+k} : k \in K\}$. Definido o V_0, as constantes c_{ij} $\forall v_i, v_j \in V_0$, da forma como foram definidas anteriormente, continuam sendo válidas. A segunda modificação está relacionada ao conjunto $K = \{1, ..., p\}$, o qual passa a englobar todos os depósitos disponíveis. Sejam Q e m a capacidade e a quantidade total de veículos homogêneos, respectivamente.

Além disso, sejam $G^k = (V_0^k, A^k)$, $\forall k \in K$, o conjunto de grafos do PRVCEMMD, um para cada depósito, no qual $V_0^k = \{v_1^k, ..., v_n^k, v_{n+k}\}$ corresponde ao conjunto de todos os vértices pertences ao grafo G^k; $V^k = \{v_1^k, ..., v_n^k\}$ ao conjunto de todos os clientes e $A^k = \{(v_i^k, v_j^k)$: $v_i^k \in V_0^k$, $v_j^k \in V_0^k$, $v_i^k \neq v_j^k$ $\}$ ao conjunto dos arcos que compõem esse grafo. Em relação aos vértices de origem e destino para cada um dos grafos, são válidas as seguintes atribuições: $V_{source}^k = V_{sink}^k = v_{n+k}$. Note que o conjunto V de clientes contém cópias de si mesmo em cada um dos $|K|$ grafos, contudo, em relação aos depósitos, neles estão contidos apenas o vértice v_{n+k} correspondente, como representado na definição do conjunto $V_0^k, \forall k \in K$.

Em relação aos recursos, também são válidos os mesmos definidos para o PRVCESFH (R

= R_M = $\{1, 2\}$), como também o consumo em cada um dos arcos ($q_{a^k,1} = p_j$ e $q_{a^k,2} = d_j, \forall a^k = (v_i^k, v_j^k) \in A^k, \forall k \in K$). É feita apenas um modificação no intervalo de consumo acumulado desses recursos, $[l_{v_i^k,1}, u_{v_i^k,1}]$ e $[l_{v_i^k,2}, u_{v_i^k,2}]$, os quais passam a ser os mesmos para todos os grafos, uma vez que a frota é homogênea. Por essa razão, tem-se que $[l_{v_i^k,1}, u_{v_i^k,1}] = [l_{v_i^k,2}, u_{v_i^k,2}] = [0, Q]$, $\forall v_i^k \in V_0^k$.

Por fim, para a formulação, são definidas novamente as variáveis inteiras x_{a^k}, $\forall a^k \in A^k, \forall k \in K$ e renomeados os custos $c_{ij} = c_{a^k}, \forall a^k = (v_i^k, v_j^k) \in A^k, \forall k \in K$. Desse modo, tem-se o seguinte modelo matemático para o PRVCEMMD adaptado ao *VRPSolver*.

$$\min \sum_{k \in K} \sum_{a^k \in A^k} c_{a^k} x_{a^k} \tag{20}$$

$$\text{s.a.} \sum_{k \in K} \sum_{a^k \in \delta^-(v_i^k)} x_{a^k} = 1, \qquad \forall v_i^k \in V^k \tag{21}$$

$$[L^k, U^k] = [0, \mu^k] \, \forall k \in K; \;\; M(x_{a^k}) = \{a^k\}, \forall a^k \in A^k, \forall k \in K; \;\; \mathcal{B}^{\mathcal{V}} = \cup_{v_i^k \in V^k} \{\{v_i^k : k \in K\}\}.$$

Da análise do modelo, tem-se que a FO (20) visa a minimização da distância total percorrida, enquanto as restrições (21) estabelecem que cada cliente deve ser visitado uma única vez. As definições dos mapeamentos e dos *packing sets* são idênticas àquelas feitas para o PRVCESFH. Uma única diferença, se comparado ao modelo dessa variante, é perceptível na definição dos limites L^k e U^k, aos quais são atribuídos 0, tendo em vista a não obrigatoriedade de utilização de todos os depósitos, e μ^k, respectivamente. Como toda a frota de veículos homogêneos (m) está disponível para todos os depósitos, atribui-se parte dela (μ^k) para cada um deles. Importante ressaltar que também são utilizadas as *lazy constraints*, a fim de garantir a simultaneidade entre as cargas de coleta e entrega contidas nos veículos, de modo que elas não excedam a capacidade máxima Q dos mesmos.

4. Resultados computacionais

Nesta seção serão fornecidas as informações inerentes aos experimentos computacionais realizados, bem como os resultados obtidos para as instâncias testadas do PRVCESFH e PRVCEMMD. Quanto aos experimentos, todos eles foram executados em uma máquina Intel® Core™ i7-3770 3.40GHz, com 16 GB de memória RAM e sistema operacional Ubuntu 16.04.6 LTS de 64 bits. Todos os modelos foram programados por meio da linguagem Julia, versão 1.3.1, sendo utilizada a versão 0.4.1 do *VRPSolver* com *BaPCod* versão 0.52e. Importante ressaltar que, para o PRVCESFH, foram utilizadas 14 dentre 28 instâncias do conjunto proposto por Avci e Topaloglu [2016], considerando o critério de até 100 clientes. Por sua vez, para o PRVCEMMD, o mesmo critério foi utilizado, sendo testadas, portanto, 21 das 33 instâncias propostas por Salhi e Nagy [1999] para a variante do PCE com múltiplos depósitos, compreendendo um total de 35 testes realizados.

Em relação aos resultados propriamente ditos, é importante lembrar que o principal interesse com a abordagem exata proposta consiste em encontrar soluções ótimas para instâncias da literatura que ainda estejam em aberto ou, caso isso não seja possível, melhorar os LIs conhecidos. Desse modo, para a variante com frota heterogênea, são apresentados os resultados obtidos sem nenhum comparativo direto, uma vez que, nos outros trabalhos que lidam com as instâncias consideradas, foram aplicados apenas métodos heurísticos. Quanto à variante com múltiplos depósitos,

por sua vez, a comparação foi feita com o B&C com *lazy constraints* de Subramanian [2012]. Nesse sentido, nas Tabelas 1 e 2 são expostos os resultados propriamente ditos. Importante ressaltar que as colunas NC, NVD e NDD, nessas tabelas, indicam o número de clientes, veículos e depósitos disponíveis, respectivamente, e que os valores das soluções ótimas estão destacados em negrito.

Analisando a Tabela 1, percebe-se que foram encontradas as soluções ótimas para 9 dentre as 14 instâncias testadas, sendo conhecidos previamente uma solução ótima para apenas 3 delas (PR_1_1 - PR_3_1). Isso significa uma contribuição de comprovação de otimalidade para 42,85% do primeiro conjunto de instâncias proposto por Avci e Topaloglu [2016]. É importante ressaltar que todos esses testes foram realizados com a consideração de um tempo limite de execução de 14400 segundos. Em relação aos resultados contidos na Tabela 2, estes são ainda mais promissores. Foram encontradas as soluções ótimas para 17 dentre as 21 instâncias de Salhi e Nagy [1999] testadas (80,95% do conjunto de instâncias considerado), enquanto haviam sido resolvidas apenas 4 delas previamente. Além disso, para as 4 instâncias não resolvidas, 3 delas tiveram seus LIs melhorados, com *gaps* entre eles e os valores de soluções inteiras já conhecidos, isto é, os limites superiores (LSs), inferiores a 3%.

Tabela 1: Resultados computacionais das instâncias do PRVCESFH.

Instância	NC	NVD	BKS	Root LB	Root time (s)	Tempo total (s)	LB	UB	GAP (%)
PR_1_1	10	2	**620,2**	620,2	1,68	1,99	620,2	**620,2**	0,00
PR_2_1	10	2	**588,5**	588,5	3,33	3,78	588,5	**588,5**	0,00
PR_3_1	15	3	**445,1**	445,1	3,37	3,98	445,1	**445,1**	0,00
PR_4_1	15	4	437,1	437,1	2,79	3,51	437,1	**437,1**	0,00
PR_5_1	20	3	494,0	490,7	4,51	21,85	494,0	**494,0**	0,00
PR_6_1	20	4	542,7	541,9	44,92	129,95	542,7	**542,7**	0,00
PR_7_1	35	3	1097,3	1086,3	28,20	797,27	1092,2	**1092,2**	0,00
PR_8_1	35	3	1586,5*	1593,1	91,76	560,47	1598,5	**1598,5**	0,00
PR_9_1	50	3	937,0	930,3	247,65	611,06	931,7	**931,7**	0,00
PR_10_1	50	2	1169,2	1112,9	199,66	14400,00	1115,3	1169,2	4,61
PR_11_1	75	3	1531,9	1495,4	463,52	14400,00	1497,4	1531,9	2,25
PR_12_1	75	2	925,6	886,2	243,10	14400,00	890,1	925,6	3,84
PR_13_1	100	2	1299,5	1172,7	3742,70	14400,00	1173,2	1299,5	9,72
PR_14_1	100	2	1539,4	1407,9	5813,61	14400,00	1409,5	1539,4	8,44

*Em contato com os autores [Avci e Topaloglu, 2016], confirmou-se que o valor em questão foi reportado erroneamente.

5. Considerações finais

Em face do exposto, pode-se afirmar que a abordagem exata com o *VRPSolver* conseguiu cumprir os objetivos pretendidos, encontrando soluções ótimas para partes consideráveis dos conjuntos de instâncias testados ou propondo LIs interessantes, responsáveis por *gaps* inferiores a 5% para a maioria dos casos. No entanto, pretende-se a busca pelo aperfeiçoamento dos resultados já obtidos, fazendo-se experimentações com diferentes valores de parâmetros do *solver* ou mesmo melhorando a formulação, se possível. Além disso, o principal ponto para a continuidade da pesquisa em questão está relacionado à generalização da abordagem utilizada para outras variantes do PRVCES. Os resultados obtidos se mostraram bastante promissores e, por essa razão, acredita-se ser possível a obtenção do mesmo desempenho para outras variantes do problema. Vale ressaltar que tal generalização já está em processo de execução, cujo ápice seria a construção de um único modelo matemático implementado para o *VRPSolver* capaz de resolver um conjunto de variantes do PRVCES, levando-se em consideração as particularidades de cada uma delas. Por fim, planeja-se a criação e disponibilização de um conjunto de novas instâncias para o PRVCES, com seus respectivos LSs, almejando-se a perpetuação das pesquisas e estudos acerca do tema.

Tabela 2: Resultados computacionais das instâncias do PRVCEMMD.

			B&C com *lazy constraints*			**Abordagem com o *VRPSolver***					
Instância	NC	NDD	LB	UB*	GAP (%)	Root LB	Root time (s)	Tempo total (s)	LB	UB	GAP (%)
GJ01H	50	4	480,43	499,12	3,74	490,06	20,90	12942,27	499,12	**499,12**	0,00
GJ02H	50	4	440,00	**440,00**	0,00	438,50	33,75	70,21	440,00	**440,00**	0,00
GJ03H	75	5	572,75	579,45	1,16	572,97	48,29	35096,31	579,45	**579,45**	0,00
GJ04H	100	2	754,33	789,19	4,42	767,26	441,23	250635,83	768,68	789,19	2,60
GJ05H	100	2	668,72	676,81	1,20	663,31	1408,01	306074,40	664,32	676,81	1,85
GJ06H	100	3	709,71	742,18	4,38	731,95	801,39	172051,38	734,77	742,18	1,00
GJ07H	100	4	706,86	732,73	3,53	726,21	588,63	336190,39	732,73	**732,73**	0,00
GJ01Q	50	4	515,09	528,30	2,50	526,95	30,61	103,77	528,30	**528,30**	0,00
GJ02Q	50	4	449,72	**449,72**	0,00	449,72	19,94	19,95	449,72	**449,72**	0,00
GJ03Q	75	5	591,35	605,25	2,30	605,03	114,40	170,30	605,25	**605,25**	0,00
GJ04Q	100	2	841,74	874,78	3,78	866,56	286,90	464498,17	867,94	874,78	0,78
GJ05Q	100	2	700,15	**700,15**	0,00	692,36	687,33	65799,52	700,15	**700,15**	0,00
GJ06Q	100	3	765,70	793,85	3,55	790,92	362,98	1751,04	793,85	**793,85**	0,00
GJ07Q	100	4	768,21	802,20	4,24	797,74	845,24	71044,17	801,91	**801,91**	0,00
GJ01T	50	4	558,12	569,43	1,99	569,43	17,69	18,20	569,43	**569,43**	0,00
GJ02T	50	4	464,13	**464,13**	0,00	464,13	26,70	26,71	464,13	**464,13**	0,00
GJ03T	75	5	606,68	624,44	2,84	623,40	65,00	123,73	624,44	**624,44**	0,00
GJ04T	100	2	923,90	962,25	3,99	962,25	325,66	331,33	962,25	**962,25**	0,00
GJ05T	100	2	723,35	733,17	1,34	725,15	783,92	304974,74	733,17	**733,17**	0,00
GJ06T	100	3	815,61	850,82	4,14	849,77	326,26	1208,57	850,82	**850,82**	0,00
GJ07T	100	4	816,31	853,54	4,36	853,54	276,39	284,13	853,54	**853,54**	0,00

*Obtidos utilizando o método heurístico em [Subramanian, 2012].

Referências

Avci, M. e Topaloglu, S. (2016). A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, 53:160–171. ISSN 09574174.

Baldacci, R. e Mingozzi, A. (2009). An unified exact method for solving different classes of vehicle routing problems. *Math. Program.*, 120:347–380.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., e Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *TOP*, 15:1–31. ISSN 1134-5764.

Braekers, K., Ramaekers, K., e Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313. ISSN 0360-8352. URL https://www.sciencedirect.com/science/article/pii/S0360835215004775.

Dantzig, G. B. e Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1): 80–91. URL https://doi.org/10.1287/mnsc.6.1.80.

Irnich, S., Toth, P., e Vigo, D. (2014). *Chapter 1: The Family of Vehicle Routing Problems*, chapter 1, p. 1–33. Monographs on Discrete Mathematics and Applications. SIAM. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611973594.ch1.

Jepsen, M., Petersen, B., Spoorendonk, S., e Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2):497–511. URL https://doi.org/10.1287/opre.1070.0449.

Koulaeian, M., Seidgar, H., Kiani, M., e Fazlollahtabar, H. (2015). A multi depot simultaneous pickup and delivery problem with balanced allocation of routes to drivers. *The International Journal of Industrial Engineering: Theory, Applications and Practice*, 22:294–313.

Li, J., Pardalos, P. M., Sun, H., Pei, J., e Zhang, Y. (2015). Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. *Expert Systems with Applications*, 42:3551–3561. ISSN 09574174.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5):377–386. ISSN 0191-2607. URL https://www.sciencedirect.com/science/article/pii/019126078990085X.

Montané, F. A. T. e Galvão, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, 33:595–619.

Nagy, G. e Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, 162 (1):126–141. ISSN 0377-2217. URL https://www.sciencedirect.com/science/article/pii/S0377221703008361. Logistics: From Theory to Application.

Nepomuceno, N., Saboia, R. B., e Pinheiro, P. R. (2019). A fast randomized algorithm for the heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Algorithms*, 12. ISSN 19994893.

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*, 19:912–930. ISSN 0968090X.

Pessoa, A., Sadykov, R., Uchoa, E., e Vanderbeck, F. (2020). A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183:483–523. ISSN 14364646.

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 0968090X.

Qu, Y. e Bard, J. F. (2015). A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. *Transportation Science*, 49:254–270. ISSN 15265447.

Ropke, S. e Pisinger, D. (2006). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171:750–775. ISSN 03772217.

Salhi, S. e Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *The Journal of the Operational Research Society*, 50(10):1034–1042. ISSN 01605682, 14769360. URL http://www.jstor.org/stable/3009928.

Subramanian, A. (2012). *Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems*. PhD thesis, Universidade Federal Fluminense, Niterói, RJ.