**Pesquisa Operacional para o Desenvolvimento**
Vol. 19, e19006, p. 1-25, 2025

**ISSN 1984-3534**

# MÉTODOS DE SOLUÇÃO APLICADOS AO PROBLEMA DE ROTAS DE COBERTURA MULTIVEÍCULO[1]

**Cristina Teruko Ota, Diego Jacinto Fiorotto, Washington Alves de Oliveira** *
Faculdade de Ciências Aplicadas
Universidade Estadual de Campinas (UNICAMP)
R. Pedro Zaccaria, 1300, Limeira, 13484-350, SP, Brasil

Recebido 17/01/2025, aceito 10/07/2025

## RESUMO

Este artigo explora métodos de solução para o problema de rotas de cobertura multiveículo, modelado com base em três conjuntos de vértices: os que podem ser visitados, os que devem ser visitados e os que precisam ser cobertos sem visita direta, mas próximos a vértices das rotas. O balanceamento entre rotas é uma restrição, garantindo a distribuição uniforme dos vértices visitados. O objetivo é minimizar a distância total percorrida. As contribuições incluem a proposta de um algoritmo genético, o estudo de um método *branch-and-cut* e a análise de um método que combina essas duas técnicas. Experimentos computacionais em instâncias adaptadas da TSPLIB foram avaliadas com estatística descritiva, curvas de perfil de desempenho e do tipo *time-to-target plot*. Os resultados mostraram que o método *branch-and-cut* enfrentou dificuldades em encontrar soluções viáveis em parte dos casos, enquanto o algoritmo genético obteve soluções para todas as instâncias em tempo satisfatório. A combinação dessas técnicas melhorou o desempenho geral da abordagem.

**Palavras-chave: Problema de rotas de cobertura multiveículo, algoritmo genético, método *branch-and-cut*.**

## ABSTRACT

This article explores solution methods for the multi-vehicle covering tour problem, modeled using three sets of vertices: those that can be visited, those that must be visited, and those that need to be covered without direct visits but should remain close to vertices included in the routes. Route balancing is imposed as a constraint, ensuring a uniform distribution of visited vertices. The objective is to minimize the total distance traveled. The contributions include the proposal of a genetic algorithm, the study of a branch-and- cut method, and the analysis of a hybrid method. Computational experiments on instances adapted from TSPLIB were assessed using descriptive statistics, performance profiles, and time-to-target plots. The results indicated that the branch-and-cut method faced difficulties in finding feasible solutions for some instances, while the genetic algorithm successfully solved all cases within a satisfactory time. The hybrid method significantly enhanced the overall performance of the approaches.

**Keywords: Multi-vehicle covering tour problem, genetic algorithm, branch-and-cut method.**

*Autor para correspondência. E-mail: waoliv@unicamp.br
DOI: https://doi.org/10.4322/PODes.2025.006

[1]*Todos os autores assumem a responsabilidade pelo conteúdo do artigo.*

## 1. Introdução

Os interesses industriais recentes em otimizar seus processos justificam a grande variedade de trabalhos na literatura com enfoque na modelagem matemática e métodos de solução para problemas complexos próprios do processo de produção. Em particular, modelos de programação linear inteira mista (PLIM), que buscam otimizar funções lineares, considerando restrições também lineares e variáveis contínuas e inteiras ou binárias, são amplamente usados na modelagem de desafios em diversos setores industriais. Por exemplo, no planejamento de escala de mão-de-obra, designação de tarefas, implementação de centros de distribuição, planejamento da produção e da logística de entrega de mercadorias, entre outros. A enumeração, mesmo que implícita, do espaço de soluções para identificar as soluções ótimas de um PLIM é computacionalmente intratável, mesmo em problemas dimensionalmente moderados. Essa limitação reforça a necessidade de desenvolver métodos de solução que priorizem o equilíbrio entre a qualidade das soluções obtidas e a eficiência computacional, destacando algoritmos exatos que combinam ramificação com planos de corte e métodos metaheurísticos.

Os problemas de roteamento de veículos são extensivamente adotados na literatura para modelar diversas aplicações que aparecem em diferentes contextos, especialmente relacionados com a cadeia logística. A necessidade de melhorar o planejamento das entregas para reduzir os custos e garantir o atendimento eficiente ao cliente exemplificam a importância dos estudos sobre roteamento de veículos. Este artigo segue essa direção, ao estudar métodos de solução para o problema de rotas de cobertura multiveículo (*m*-PRC), o qual foi introduzido por Hachicha et al. (2000) da seguinte forma: uma frota de veículos idênticos está em um depósito e cada veículo usado percorre uma rota que inicia e termina no depósito, visitando um conjunto de localidades; algumas localidades relevantes são de visita obrigatória, algumas localidades podem ser visitadas e outras localidades estratégicas devem ser cobertas porém não visitadas, ou seja, devem estar próximas o suficiente de uma localidade visitada por alguma rota; o objetivo é minimizar a distância total percorrida pelos veículos usados tal que a capacidade de cada veículo não seja excedida.

As aplicações surgem em vários contextos em que problemas práticos têm sido resolvidos por meio de variantes do *m*-PRC, incluindo a localização de caixas de correios cobrindo localidades próximas o suficiente (Labbé e Laporte, 1986), o planejamento de rotas para equipes médicas atuarem em locais afetados por desastre porém próximos de outros vilarejos (Doerner e Hartl, 2008), na construção de rotas de vigilância cobrindo locais sensíveis por meio de patrulhas (Oliveira et al., 2015), no transporte que realiza tarefas perigosas (Margolis et al., 2022), na vigilância utilizando veículos não tripulados (Fröhlich et al., 2023), no monitoramento de grandes áreas marítimas (Torabi et al., 2023), entre outros.

A principal contribuição deste artigo consiste do estudo e aplicação de métodos de solução para o *m*-PRC com equilíbrio nas visitas, tal que, para alcançar o balanceamento exigido, a quantidade de visitas por rota é adequadamente inserida no modelo matemático como uma variável de decisão. Assim, este trabalho desenvolve uma metaheurística baseada em algoritmo genético, aplica o método *branch-and-cut* proposto por Ota et al. (2024) e explora esses dois métodos agregados para resolver o *m*-PRC. As três abordagens são usadas para resolver uma série de instâncias adaptadas da biblioteca TSPLIB. Os resultados dos experimentos computacionais são detalhados por meio de estatística descritiva, onde uma proposta de perfil de desempenho baseada no índice de qualidade é apresentada para comparar os métodos. Além disso, o comportamento aleatório do algoritmo genético é brevemente analisado por meio de um gráfico do tipo *time-to-target plot* obtido com simulações.

O restante deste artigo está organizado da seguinte maneira: A Seção 2 resume a literatura existente sobre o tema; a Seção 3 descreve o problema estudado e notações preliminares; a Seção 4 apresenta a formulação matemática para o problema; as três abordagens de solução são detalhados na Seção 5; uma série de experimentos e resultados são descritos e analisados na Seção 6; e a Seção 7 apresenta as conclusões do trabalho e propostas de pesquisa futura.

## 2. Revisão da Literatura

O primeiro estudo sobre o problema de rotas de cobertura pode ser atribuído a Current (1981), que posteriormente foi formalmente definido por Gendreau et al. (1997) como o problema de encontrar um circuito hamiltoniano de comprimento mínimo que visita um subconjunto de vértices, de modo que todos os vértices a serem cobertos estejam dentro de uma distância predefinida de algum vértice pertencente ao circuito. A extensão do problema de rotas de cobertura para múltiplos veículos proposta por Hachicha et al. (2000) introduziu capacidades para os diferentes veículos, limitando a carga total transportada e a distância total percorrida. Hachicha et al. (2000) foram os primeiros a propor um modelo matemático para o *m*-PRC. Observa-se que tal modelo pode ser reduzido a um problema de roteamento de veículos (PRV) ao considerar o conjunto de pontos de cobertura vazio e todos os outros pontos como visita obrigatória. Assim, o *m*-PRC é identificado como um problema NP-difícil, o que justifica o uso dos métodos de solução desenvolvidos neste artigo.

Diversas variantes do *m*-PRC foram propostas para lidar com diferentes cenários práticos. Doerner e Hartl (2008) resolveram um problema de ajuda humanitária ao planejar o fornecimento de provisões às populações afetadas por desastres, onde as organizações de saúde visitam apenas um subconjunto de paradas estratégicas de tal maneira que todas as comunidades vizinhas possam chegar a uma dessas paradas dentro de um tempo razoável. Na configuração de rondas de policiamento, Oliveira et al. (2015) estudaram a construção de rotas de patrulhamento urbano preventivo que estimula presença territorial de agentes, ou seja, uma coleção de trajetos que cobre a maior área possível, garantindo maior visibilidade da viatura e contato com a população, prevenindo atos ilegais e, consequentemente, dando maior sensação de segurança à comunidade. Tan e Hill (2016) utilizaram o *m*-PRC para planejar rotas de tamanho mínimo para cobrir um conjunto de pontos de controle críticos, que exigem vigilância obrigatória, por veículos terrestres não tripulados, enquanto alguns pontos de controlo podem ser visitados enquanto cobrem zonas dentro de uma distância fixa. Kammoun et al. (2017) estudaram e resolveram um problema envolvendo campanhas de vacinação, onde a equipe médica precisa fazer viagens de tempo curto, usando um procedimento de busca em vizinhança variável baseado em métodos de descida. Flores-Garza et al. (2017) introduziram um *m*-PRC com rotas acumulativas para logística humanitária. Torabi et al. (2023) apresentaram soluções para o monitoramento de ambientes marinhos, de forma que poluentes podem ser medidos em locais distantes da sua fonte devido às correntes oceânicas. Adicionalmente, Fröhlich et al. (2023) destacaram o uso do *m*-PRC para obter soluções em aplicações militares, onde áreas específicas de controle podem ser examinados por veículos não tripulados sem entrar em zona de detecção.

As técnicas de solução para o *m*-PRC desenvolvidas na literatura combinam formulações matemáticas ajustadas e métodos de otimização. De modo expressivo, modelos de fluxo em redes são propostos e resolvidos com métodos exatos de enumeração implícita que combinam ramificação com planos de corte. Em consequência, a dimensão das instâncias resolvidas cresce ao aplicar métodos baseados em heurísticas e metaheurísticas. Adicionalmente, esses métodos também são combinados como propostas de abordagens híbridas. Combinações de heurísticas construtivas ficam evidentes, enquanto algoritmos evolutivos aparecem como metaheurísticas promissoras na resolução do *m*-PRC. Métodos exatos para variantes do PRV são adaptados para gerar versões do algoritmo *branch-and-cut* para o *m*-PRC (Hà et al., 2013; Pham et al., 2017; Ota et al., 2024), caracterizados por diferentes caminhos para detectar violação de restrições, técnicas de separação e regras de ramificação. As heurísticas seguem caminhos semelhantes ao serem adaptadas de técnicas para o PRV.

No estudo de Gendreau et al. (1997) para o problema com uma única rota, uma formulação de fluxo em rede com dois índices foi desenvolvida e resolvida exatamente por meio de um algoritmo *branch-and-cut*. Além disso, os autores consideraram uma abordagem que combina heurísticas para o problema de cobertura de conjuntos e o problema do caixeiro viajante. Hachicha et al. (2000) apresentaram uma formulação matemática para o *m*-PRC e desenvolveram três heurísticas

construtivas baseadas em métodos correspondentes para o PRV que permitem encontrar boas soluções para instâncias realistas em um tempo computacional aceitável. Baldacci et al. (2005) desenvolveram formulações de fluxo em arcos para a versão com uma única rota do problema. Então, soluções exatas foram obtidas com um algoritmo *branch-and-cut*, enquanto instâncias relativamente grandes foram resolvidas por meio de três algoritmos heurísticos baseados em busca por dispersão. Jozefowiez et al. (2007) estudaram um modelo de otimização biobjetivo para o *m*-PRC, para o qual foi avaliado o conflito entre o tamanho total das rotas e a distância limite de cobertura. Tricoire et al. (2012) agregaram demanda estocástica ao modelo biobjetivo para avaliar a probabilidade de ocorrer demandas não cobertas. Hà et al. (2013) desenvolveram um algoritmo *branch-and-cut* e uma metaheurística evolutiva de busca local. Lopes et al. (2013) aplicaram um algoritmo *branch-and-price* para o *m*-PRC, combinando em uma rotina de geração de colunas e os problemas de cobertura de conjuntos e caminho mais curto. Karaoğlan et al. (2018) desenvolveram um modelo de programação não linear inteira para um *m*-PRC com incertezas, incorporando probabilidade de cobertura para maximizar o valor esperado de cobertura dos clientes. Kammoun et al. (2019) estudaram um *m*-PRC considerando múltiplas coberturas, onde um vértice deve ser coberto várias vezes por diferentes rotas. Glize et al. (2020) desenvolveram um método exato para o *m*-PRC biobjetivo baseado no método $\varepsilon$-restrito. Ota et al. (2024) desenvolveram um algoritmo *branch-and-cut* para resolver o *m*-PRC com balanceamento de rotas. Recentemente, Torabi et al. (2025) inovaram com um método de aprendizado de máquina guiado por uma hiper-heurística de reforço profundo para resolver uma extensão do problema em que o raio de cobertura depende da quantidade de tempo gasto pelo veículo em cada vértice.

Entre os métodos de solução que utilizam metaheurísticas, os algoritmos evolutivos baseados em algoritmos genéticos são amplamente usados para problemas de roteamento. Dentre algumas aplicações de algoritmos genéticos, Jozefowiez et al. (2007) desenvolveram um método de solução em duas fases para um modelo biobjetivo do *m*-PRC que combina um algoritmo evolutivo multiobjetivo com um algoritmo *branch-and-cut*, em que os operadores genéticos foram desenhados para capturar os aspectos essenciais do problema para alcançar soluções bem distribuídas na curva de Pareto. Pham et al. (2017) introduziram uma variante do *m*-PRC que incorpora multi-coberturas de vértices. Adicionalmente, um modelo de programação linear inteira foi proposto e resolvido com um algoritmo *branch-and-cut* e um algoritmo genético. O último forneceu soluções de alta qualidade para o problema, superando outras metaheurísticas comparadas. Jiang et al. (2022) desenvolveram um algoritmo evolutivo de convergência rápida que utiliza uma matriz de probabilidade para guiar o operador *crossover* e acelerar a convergência do método, além de aplicar uma estratégia de preservação de diversidade. Dutta et al. (2022) estudaram uma modelagem multiobjetivo para o problema de roteamento de veículos verdes que considera sustentabilidade e tem o objetivo de minimizar custos operacionais e emissões de carbono. O método utilizado agrupa os clientes usando uma modificação do *k-means* e resolve o problema com algoritmos evolutivos multiobjetivo do tipo SPEA2 e NSGA-II. Além disso, trabalhos na literatura mostram a eficiência do algoritmo genético aplicados a diferentes variantes do PRV.

Problemas de roteamento que exigem algum tipo de balanceamento (fairness) entre rotas também têm ganhado atenção na literatura, incluindo os casos de balanceamento na quantidade de vértices visitados (Oliveira et al., 2015; Bektaş et al., 2019; Ota et al., 2024), no tamanho/custo das rotas (Bektaş e Letchford, 2020; Mara et al., 2021; Feng e Wei, 2023), e na quantidade de carga/força de trabalho (Matl et al., 2019; Lehuédé et al., 2020; Mancini et al., 2021; Fröhlich et al., 2023). O impacto do balanceamento tem sido avaliado por meio de elaboradas funções de equidade (Matl et al., 2019; Bektaş e Letchford, 2020; Feng e Wei, 2023). Em particular, Glize et al. (2020) indicaram o estudo de Oliveira et al. (2015) como sendo o primeiro a propor uma variante do *m*-PRC que considera o balanceamento entre rotas, onde as rotas são usadas para o serviço de patrulhamento preventivo urbano realizado por viaturas policiais, para as quais o balanceamento nas visitas melhora a equidade do contato preventivo dos agentes policiais com a população. Abordagens que consideram que cada rota pode visitar uma quantidade máxima e

mínima de clientes para conseguir o balanceamento desejado também foi estudado (Bektaş et al., 2019). No entanto, diferentemente desses trabalhos anteriores, cujo valores de máximo e mínimo de visitas são parâmetros, Ota et al. (2024) consideraram esses valores como variáveis. O resultado foi um modelo não linear que precisou ser adequadamente linearizado. Mais especificamente, Ota et al. (2024) combinaram os modelos de fluxo de dupla direção das cargas de Baldacci et al. (2004) e Hà et al. (2013) para desenvolver o modelo *m*-PRC estudado neste trabalho.

O algoritmo *branch-and-cut* desenvolvido neste artigo considera as desigualdades válidas descritas por Ota et al. (2017, 2024) ao incluir o balanceamento entre rotas. Além disso, um algoritmo genético é desenvolvido ao agregar as heurísticas construtivas de Hachicha et al. (2000); Hà et al. (2013); Oliveira et al. (2015) e Ota et al. (2024) com as técnicas bioinspiradas de Ombuki-Berman e Hanshar (2009). Então, este último é usado de forma conjunta com o algoritmo *branch-and-cut* proposto na tentativa de aprimorar as soluções do problema.

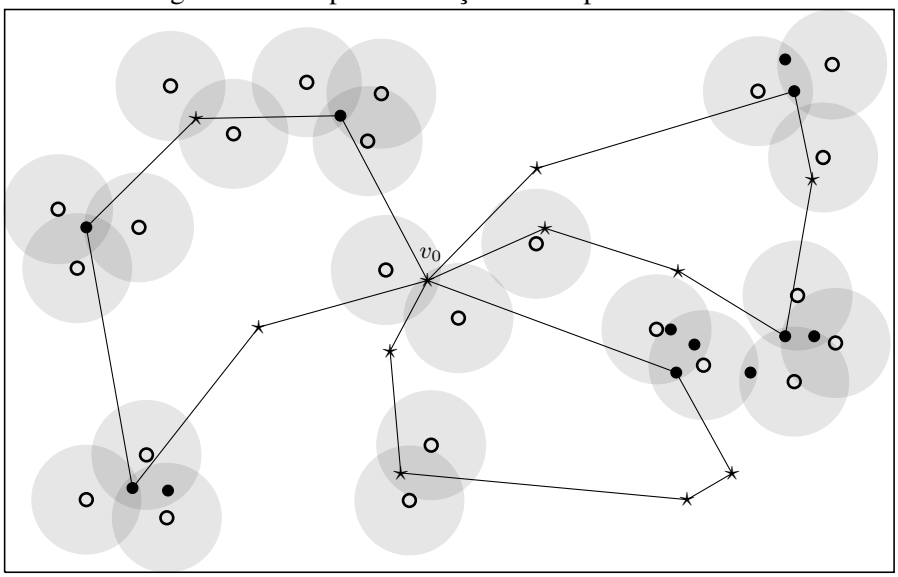
## 3. Descrição do Problema

Esta seção apresenta uma descrição preliminar do *m*-PRC para então apresentar a formulação matemática estudada neste artigo. A modelagem de Hachicha et al. (2000) considera um grafo não direcionado $G = (V \cup W, E)$, em que $V \cup W$ é o conjunto de vértices e $E$ é o conjunto de arestas. O conjunto dos $n$ vértices que podem ser visitados é representado por $V = \{v_0, \ldots, v_{n-1}\}$, incluindo um subconjunto de visita obrigatória, denotado por $T = \{v_0, \ldots, v_{t-1}\} \subset V$. O vértice $v_0$ representa o depósito, local onde uma frota homogênea de $m$ veículos fica disponível. Os vértices que devem ser cobertos, porém não visitados, são inseridos no conjunto denotado por $W = \{w_1, \ldots, w_L\}$. $E = \{(v_i, v_j) : v_i \in V \cup W, v_j \in V \cup W, i < j\}$ denota o conjunto de arestas, onde cada aresta $\{v_i, v_j\}$ está associada a um distância de deslocamento $c_{ij}$. O objetivo consiste em determinar um conjunto de $m$ rotas de comprimento total mínimo, considerando as seguintes condições: i) no máximo $m$ rotas são usadas, iniciando e terminando no depósito; ii) cada vértice de $V$ é visitado no máximo uma vez, enquanto que cada vértice de $T$, com exceção do depósito, pertence a exatamente uma rota; iii) cada vértice de $W$ deve ser coberto por uma rota, ou seja, estes vértices devem estar dentro do raio de alcance $c$ dos vértices em $V$ que pertencem a uma rota, assumindo que $v_0$ não cobre todos os vértices de $W$; iv) o número de vértices visitados por rota, excluindo-se o depósito, é limitado; e v) o comprimento de cada rota não pode exceder uma capacidade predeterminada.

A Figura 1 ilustra uma típica solução viável para o problema, considerando $m = 3$, $|V| = 23$, $|T| = 11$ e $|W| = 24$, onde $|\cdot|$ representa a cardinalidade do conjunto. Observe que as três rotas contêm todos os vértices em $T$ (os vértices em formato de estrelas), no entanto alguns vértices em $V \setminus T$ (os vértices em formato de pontos) não foram escolhidos. Os discos cinzas ilustram as vizinhanças dos vértices em $W$ (vértices em formato de círculos). Note que a solução é viável no que diz respeito à cobertura, cada disco cinza contém pelo menos um vértice visitado, e todas as rotas iniciam e terminam no depósito $v_0$.

A cobertura de conjuntos é um fator importante nessa modelagem e na configuração das instâncias do problema. Seja $S_l = \{v_h \in V : c_{hl} \leq c\}$ o conjunto dos vértices em $V$ que cobrem cada vértice $w_l \in W$, ou seja, em $S_l$ estão todos os vértices $v_h \in V$ tal que a distância, considerada aqui na norma Euclidiana, até $w_l$ seja menor que $c$ (parâmetro que fornece o raio limite de cobertura). A Figura 2 ilustra um exemplo de cobertura dos vértices em $W$, tal que, o vértice $w_1$ é coberto pelos vértices $v_1$ e $v_2$ e o vértice $w_2$ é coberto por $v_3$, $v_4$ e $v_5$, ou seja, $S_1 = \{v_1, v_2\}$ e $S_2 = \{v_3, v_4, v_5\}$. Para o conjunto $S_l$, exigimos $|S_l| \geq 2$, para cada $w_l \in W$, pois se existir algum $S_l$ com apenas um vértice (ou seja, $S_l := \{v_i\}$), então $v_i$ pode ser "virtualmente" incluído como um elemento de $T$ e $w_l$ pode ser "virtualmente" eliminado de $W$. Além disso, podemos assumir que $S_l \cap T = \emptyset$, uma vez que não precisamos dar atenção aos vértices de $W$ que estão próximos o suficiente de vértices em $T$ (Oliveira et al., 2015), ou seja, $S_l = \{v_h \in V \setminus T : c_{hl} \leq c\}$. Assim, note que "virtualmente" um vértice de $V \setminus T$ pode ser incluído em $T$ e alguns vértices de $W$

Figura 1: Exemplo de solução viável para o *m*-PRC.

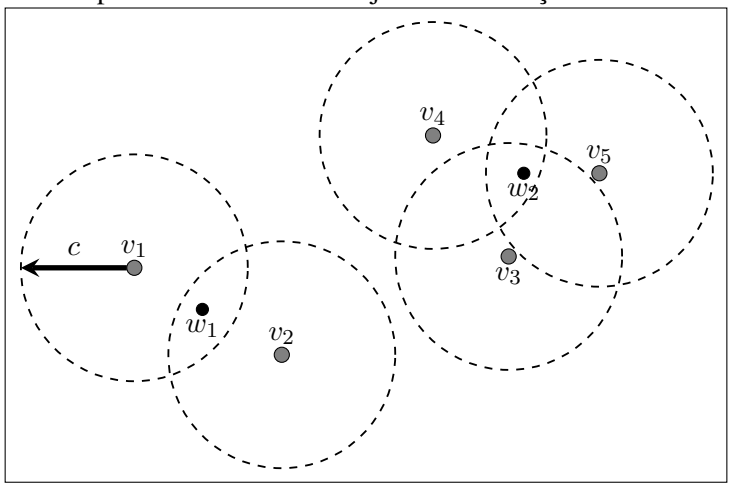



Fonte: Oliveira et al. (2015).

podem ser deletados da Figura 1. Por outro lado, vértices de $W$ com múltiplas coberturas, similar a cobertura de $w_2$ em $S_2$ (também observado na Figura 1), aumenta as combinações de escolha de quem não precisa ser visitado em $V \setminus T$.

Soluções exatas para o *m*-PRC foram determinadas por Hà et al. (2013) considerando a resolução de uma formulação matemática baseada em um modelo de fluxo em redes. Oliveira et al. (2015) foram os primeiros a examinar o *m*-PRC com balanceamento entre as rotas, resolvendo o problema com uma série de métodos heurísticos. A formulação matemática resolvida neste artigo incorpora as duas ideias anteriores em um modelo de fluxo em redes com balanceamento entre rotas para viabilizar a sua resolução por meio de métodos exatos. Então, neste artigo, o modelo de fluxo em redes para o *m*-PRC com balanceamento nas visitas entre rotas é denotado por *m*-PRCB. Note que a solução ilustrada na Figura 1 também é viável com respeito ao balanceamento, onde cada rota percorre um trajeto que tem aproximadamente a mesma quantidade de vértices, variando em no máximo um vértice.

Figura 2: Exemplo de cobertura de conjuntos em relação a $W$ na modelagem.




Fonte: Autores.

## 4. Formulação Matemática

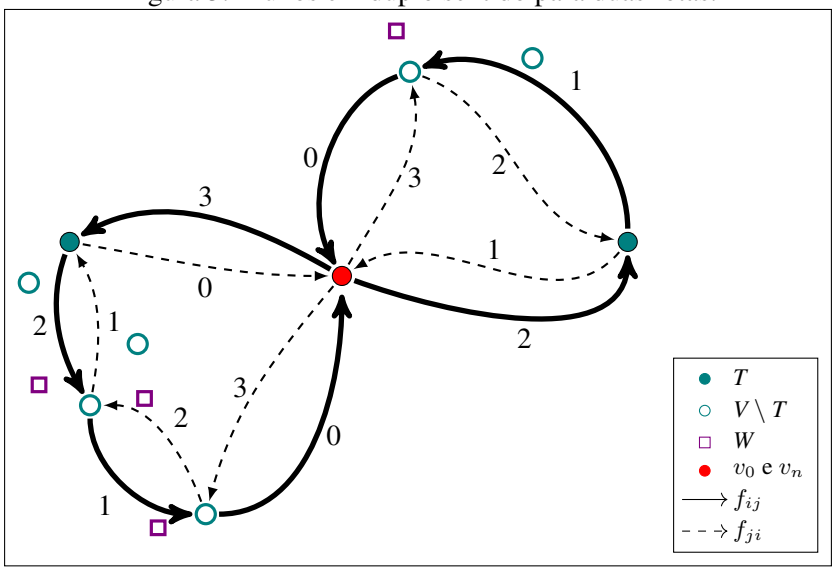
Esta seção apresenta a formulação matemática para o *m*-PRCB que foi estudada e resolvida neste artigo. A formulação pela abordagem de fluxo em redes de dois índices, que simula o movimento de mercadoria ao longo das arestas, elimina a informação de qual vértice visitado pertence a qual rota e o trajeto percorrido por cada rota. No entanto, como vantagem, ela permite incluir as restrições de eliminação de subrotas em quantidade polinomial. A formulação usada foi desenvolvida por Ota et al. (2024) considerando modificações do modelo de Hà et al. (2013) para inserir em um modelo de fluxo em redes o balanceamento de visitas entre rotas. O resultado das modificações é um modelo de otimização não linear que precisa ser linearizado.

O *m*-PRCB é formulado sobre um grafo não direcionado $G = (V \cup W,\ E_1 \cup E_2)$, para o qual o conjunto de arestas $E$ da definição anterior é substituído pelo conjunto de arestas $E_1 \cup E_2$, onde cada aresta em $E_1 = \{(v_i, v_j) : v_i, v_j \in V,\ i < j\}$ tem associada a ela uma distância $c_{ij}$ e cada aresta em $E_2 = \{(v_i, w_j) : v_i \in V \setminus T,\ w_j \in W\}$ tem associada a ela uma distância $d_{ij}$. Um vértice artificial $v_n$, representando uma cópia do depósito $v_0$, é inserido no conjunto de vértices para criar uma extensão do grafo original $G$ para $\widehat{G} = (\widehat{V} \cup W,\ \widehat{E_1} \cup E_2)$, onde $\widehat{V} := V \cup \{v_n\}$, $V^{'} := V \setminus \{v_0\}$, $T^{'} := T \setminus \{v_0\}$, $\widehat{E_1} := E_1 \cup \{(v_i, v_n), v_i \in V^{'}\}$ e, além disso, considera-se $c_{in} = c_{0i}$ para todo $v_i \in V^{'}$. Além disso, seja $S_l = \{v_h \in V : c_{hl} \leq c\}$ o conjunto de vértices em $V$ que estão situados dentro da distância permitida $c$ de cada vértice $w_l \in W$. Observe que se apenas um vértice $v_h \in V$ está próximo o suficiente de algum vértice $w_l$, podemos incluir $v_h$ em $T$ e eliminar $w_l$ de $W$. Como não precisamos nos preocupar em cobrir os vértices que estão próximos o suficiente de algum vértice em $T$, podemos assumir que $S_l \cap T = \emptyset$. Portanto, usamos $S_l = \{v_h \in V \setminus T : c_{hl} \leq c\}$, com $|S_l| \geq 2$ para todo $w_l \in W$, para modelar a cobertura realizada por $m$ rotas na solução do problema, onde $v_i \in V^{'}$ é visitado no máximo uma vez, $v_i \in T^{'}$ é visitado exatamente uma vez, $w_l \in W$ deve estar a no máximo $c$ de distância de um vértice visitado. Neste trabalho, as rotas são comparadas por meio da quantidade de vértices que cada rota visita. A tolerância $r$ é usada para verificar a discrepância no balanceamento de visitas entre rotas, ou seja, a diferença entre o total de vértices visitados em duas rotas deve ser menor ou igual a $r$. Sejam $\tilde{\rho}$ e $\rho$ as quantidades máxima e mínima de visitas permitidas por rota. Então, cada rota é um circuito em $G$ contendo o vértice $v_0$ e, para qualquer duas rotas, $r$ é a diferença máxima na quantidade de vértices ente elas, isto é, se $\tilde{\rho} - \rho \leq r$ para todas as rotas, implica em rotas balanceadas. As seguintes variáveis de decisão são usadas na descrição do modelo matemático.

- $f_{ij}$ e $f_{ji}$: variáveis contínuas não negativas que representam o fluxo de "carga" nas arestas $\{v_i, v_j\} \in \widehat{E_1}$ de uma solução viável. Partindo do depósito, cada veículo pode carregar até $\tilde{\rho}$ unidades de carga, mas, neste modelo, cada visita consome uma unidade de carga. No trajeto de $v_i$ para $v_j$, o fluxo $f_{ij}$ representa a quantidade de carga carregada no veículo, ou seja, o número de vértices que ainda pode ser visitado na rota e $f_{ji}$ representa o espaço vazio no veículo, que também pode ser associado com visitas. A relação de $f_{ij}$ e $f_{ji}$ é dada por $f_{ji} = \tilde{\rho} - f_{ij}$.
- $\tilde{\rho}$ e $\rho$: variáveis inteiras que representam as quantidades máxima e mínima de visitas permitidas por rota.
- $x_{ij}, i < j$: variável binária igual a 1 se um veículo usa a aresta $\{v_i, v_j\} \in \widehat{E_1}$ na rota; igual a 0 caso contrário.
- $y_i$: variável binária igual a 1 se o vértice $v_i \in V^{'}$ é visitado por uma rota; igual a 0 caso contrário.

Baldacci et al. (2004) descreveram as variáveis de fluxo $f_{ij}$ que definem dois caminhos para o fluxo, um caminho iniciando do depósito $v_0$ para a cópia do depósito $v_n$ representa a carga do veículo e um segundo caminho iniciando da cópia do depósito $v_n$ para o depósito $v_0$ representa o espaço vazio no veículo. A Figura 3 ilustra uma representação dos fluxos em duplo sentido de um exemplo de solução viável típica para o *m*-PRCB, considerando $m = 2$ e $\tilde{\rho} = 3$. O exemplo de solução também é viável com respeito ao balanceamento.

Figura 3: Fluxos em duplo sentido para duas rotas.




Fonte: Adaptado de Ota et al. (2024).

**Definição da função objetivo:** o objetivo é minimizar o comprimento total das rotas na solução.

$$\text{Minimizar} \sum_{\{v_i,v_j\}\in\widehat{E_1}} c_{ij}x_{ij} \tag{1}$$

**Definição das restrições:** o modelo está sujeito ao seguinte conjunto de restrições.

(a) restrições que estabelecem o padrão do fluxo, partindo do depósito e sua cópia, percorrendo os vértices de $V^{'}$. Elas ajudam a garantir o balanceamento entre rotas e também eliminam subrotas.

- O fluxo total de saída no depósito é igual ao total de vértices que são visitados na solução.

$$\sum_{v_j\in V'} f_{0j} = \sum_{v_i\in V'} y_i \tag{2}$$

- O fluxo total de saída da cópia do depósito $v_n$ é igual a capacidade total da frota de veículos.

$$\sum_{v_j\in V'} f_{nj} = m\tilde{\rho} \tag{3}$$

- Se o vértice $v_i$ é visitado, então o fluxo total de entrada menos o fluxo total de saída neste vértice é igual 2, 0 caso contrário. Assim, a conservação do fluxo é garantida em cada vértice visitado.

$$\sum_{v_j\in\widehat{V}} (f_{ji} - f_{ij}) = 2y_i, \quad \forall v_i \in V^{'} \tag{4}$$

- O fluxo total de entrada no depósito corresponde a capacidade residual da frota de veículos.

$$\sum_{v_j\in V'} f_{j0} = m\tilde{\rho} - \sum_{v_i\in V'} y_i \tag{5}$$

• Se uma aresta está ativa, então a definição das variáveis de fluxo é atendida, garantindo que a soma dos fluxos nos dois sentidos seja igual a $\tilde{\rho}$, caso uma das arestas $\{v_i, v_j\}$ ou $\{v_j, v_i\}$, $i < j$, seja percorrido por uma rota na solução.

$$f_{ij} + f_{ji} = \tilde{\rho}(x_{ij} + x_{ji}), \quad \{v_i, v_j\} \in \widehat{E_1},\ i < j \tag{6}$$

• Se um veículo parte do depósito para um vértice $v_j$, então a quantidade mínima $\rho$ de vértices visitados nesta rota é garantida.

$$f_{0j} \geq x_{0j}\rho, \quad \forall v_j \in V^{'} \tag{7}$$

• Não negatividade das variáveis de fluxo.

$$f_{ij} \geq 0,\ f_{ji} \geq 0, \quad \{v_i, v_j\} \in \widehat{E_1},\ i < j \tag{8}$$

• Integralidade das variáveis inteiras.

$$\tilde{\rho},\ \rho \geq 0 \text{ e inteiros} \tag{9}$$

(b) restrições de designação e ativação das variáveis binárias. Elas determinam o caminho das rotas por meio das arestas ativas e quais vértices são visitados.

• Cada vértice $w_l \in W$ deve ser coberto por pelo menos um vértice de $V \setminus T$ , considerando $S_l = \{v_h \in V \setminus T : c_{hl} \leq c\}$.

$$\sum_{v_i \in S_l} y_i \geq 1, \quad \forall w_l \in W \tag{10}$$

• Cada vértice em $T$, exceto o depósito, deve ser visitado exatamente uma vez.

$$y_i = 1, \quad \forall v_i \in T^{'} \tag{11}$$

• Se a aresta $\{v_i, v_j\}$ é usada, então ela pode ser atravessada em apenas uma direção.

$$x_{ij} + x_{ji} \leq 1, \quad v_i, v_j \in V^{'},\ i < j \tag{12}$$

• Se o vértice $v_k \in V^{'}$ é visitado, então ele é visitado apenas uma vez, contendo apenas uma aresta ativa com fluxo de chegada e apenas uma aresta ativa com fluxo de partida.

$$\sum_{v_i \in \widehat{V},\ i \neq k} x_{ik} = y_k, \quad \forall v_k \in V^{'}, \tag{13}$$

$$\sum_{v_j \in \widehat{V},\ j \neq k} x_{kj} = y_k, \quad \forall v_k \in V^{'} \tag{14}$$

• Em uma solução, não existem rotas chegando no depósito $v_0$ ou partindo da cópia do depósito $v_n$.

$$x_{i0} = 0, \quad \forall v_i \in V^{'}, \tag{15}$$

$$x_{nj} = 0, \quad \forall v_j \in V^{'} \tag{16}$$

• Integralidade das variáveis binárias.

$$x_{ij} \in \{0, 1\}, \quad \{v_i, v_j\} \in \widehat{E_1},\ i < j, \tag{17}$$

$$y_i \in \{0, 1\}, \quad \forall v_i \in V^{'} \tag{18}$$

(c) extensão do modelo com a linearização das restrições não lineares.

No modelo (1)–(18), o tamanho da frota $m$ é um parâmetro, enquanto que a quantidade máxima $\tilde{\rho}$ e a quantidade mínima $\rho$ de visitas permitidas por rota são variáveis inteiras. Assim, as restrições (3) e (5) permanecem lineares, enquanto que as restrições (6) e (7) tornam-se não lineares. No entanto, considerando $\tilde{\rho}$ e $\rho$ variáveis de decisão, a solução do problema determina as quantidades ideais de vértices visitados em cada rota, além de facilitar a inclusão de restrições de balanceamento. Note que a quantidade de vértices em cada rota é finita, assim é possível escolher os parâmetros $\tilde{\beta}$ e $\beta$ tal que $\tilde{\rho} \leq \tilde{\beta}$ e $\rho \leq \beta$ (estimativas para $\tilde{\beta}$ e $\beta$ aparecem na Seção 6.1). Então, o método de envelopes de McCormick pode ser adequadamente usado para linearizar os produtos de variáveis nas restrições (6) e (7).

Em (6), as variáveis $x_{ij}$ e $x_{ji}$ são binárias e $\tilde{\rho}$ é uma variável limitada por $\tilde{\beta}$, com $0 \leq \tilde{\rho} \leq \tilde{\beta}$, então definimos uma nova variável $a_{ij} = \tilde{\rho}(x_{ij} + x_{ji})$ para substituir (6) e incluir no modelo as seguintes novas restrições lineares.

$$f_{ij} + f_{ji} = a_{ij}, \quad \{v_i, v_j\} \in \widehat{E_1} \tag{19}$$

$$a_{ij} \leq \tilde{\beta}(x_{ij} + x_{ji}), \quad \{v_i, v_j\} \in \widehat{E_1} \tag{20}$$

$$a_{ij} \leq \tilde{\rho}, \quad \{v_i, v_j\} \in \widehat{E_1} \tag{21}$$

$$a_{ij} \geq \tilde{\rho} - \tilde{\beta}(1 - (x_{ij} + x_{ji})), \quad \{v_i, v_j\} \in \widehat{E_1} \tag{22}$$

$$a_{ij} \geq 0, \quad \{v_i, v_j\} \in \widehat{E_1} \tag{23}$$

$$\tilde{\rho} \geq 0 \text{ e inteiro} \tag{24}$$

Em (7), a variável $x_{0j}$ é binária e $\rho$ é uma variável limitada por $\beta$, com $0 \leq \rho \leq \beta$, então, similarmente, definimos uma nova variável $s_{0j} = x_{0j}\rho$ para substituir (7) e incluir no modelo as seguintes novas restrições lineares.

$$f_{0j} \geq s_{0j}, \quad v_j \in V^{'} \tag{25}$$

$$s_{0j} \leq \beta x_{0j}, \quad v_j \in V^{'} \tag{26}$$

$$s_{0j} \leq \rho, \quad v_j \in V^{'} \tag{27}$$

$$s_{0j} \geq \rho - \beta(1 - x_{0j}), \quad v_j \in V^{'} \tag{28}$$

$$s_{0j} \geq 0, \quad v_j \in V^{'} \tag{29}$$

$$\rho \geq 0 \text{ e inteiro} \tag{30}$$

Finalmente, para garantir o balanceamento entre rotas no modelo, se duas rotas são selecionadas, a diferença na quantidade de vértices visitados entre elas não ultrapassa a tolerância de equilíbrio.

$$\tilde{\rho} - \rho \leq r \tag{31}$$

## 5. Métodos de Solução

Esta seção descreve os três métodos de solução estudados para resolver o *m*-PRCB. Um algoritmo genético foi configurado para obter soluções satisfatórias com baixo tempo computacional. A motivação para utilizar o algoritmo genético se dá pela facilidade em encontrar soluções viáveis para o problema, além de ser um método reconhecido na literatura por geralmente fornecer resultados promissores. Para obter soluções exatas, um algoritmo *branch-and-cut* foi aplicado. Adicionalmente, a melhor solução do algoritmo genético foi usada como solução inicial e diversas desigualdades válidas foram incluídas para acelerar a convergência do algoritmo *branch-and-cut*.

### 5.1. Algoritmo Genético

Algoritmos genéticos surgem na resolução de problemas de engenharia inspirados na ideia da evolução populacional, em que uma população de soluções candidatas para um dado problema

evolui ao longo do tempo por meio de operadores guiados pela variação natural da genética e na seleção natural. De maneira geral, a rotina computacional de um algoritmo genético aproveita as características de um problema específico para representar uma solução por meio de uma estrutura semelhante a um cromossomo. Sobre essa estrutura podem ser aplicados operadores de seleção, mutação e *crossover* de forma a preservar as características relativas às soluções do problema.

O algoritmo genético desenvolvido é uma contribuição deste artigo que agrega de maneira prática três rotinas difundidas na literatura e que foram usadas com sucesso na resolução de problemas de roteamento. Hà et al. (2013) e Ota et al. (2024) usaram uma heurística construtiva para a resolução do subproblema de cobertura (SPC). Hachicha et al. (2000) e Oliveira et al. (2015) desenvolveram heurísticas do tipo *route-first/cluster-second* (RF/CS) para determinar soluções para o *m*-PRC. Ombuki-Berman e Hanshar (2009) apresentaram técnicas bioinspiradas para resolver variantes do *m*-PRV. Entre elas, foi desenvolvido o operador Crossover de Rota de Melhor Custo (CRMC). Assim, o algoritmo genético proposto é dividido em três fases, conforme descrito na Tabela 1. As duas primeiras fases são de construção, que determinam os indivíduos da população inicial do algoritmo genético. A rotina completa é denominada AG-CRMC, pois as rotas obtidas como soluções iniciais das Fases 1 e 2 são então melhoradas com o uso da técnica CRMC e operações de mutação.

Tabela 1: Rotina AG-CRMC. Fases 1 e 2: construtiva. Fase 3: melhoria.

| **Fase** | **Método** | **Descrição** |
|---|---|---|
| 1 | Solução do SPC | Seleção dos vértices de $V \setminus T$ que cobrem todos os vértices de $W$ (Hà et al., 2013; Ota et al., 2024). |
| 2 | RF/CS | Construção de rotas iniciais, indivíduos da população inicial (Hachicha et al., 2000; Oliveira et al., 2015). |
| 3 | CRMC | Melhoria das rotas (Ombuki-Berman e Hanshar, 2009). |

Fonte: Autores.

A abordagem *route-first/cluster-second* é usada para obter cada solução viável a ser adicionada na população inicial. Ou seja, uma primeira rota (*route-first*) é determinada combinando a solução de um subproblema de cobertura e uma abordagem da heurística de varredura e, na sequência, uma regra aleatória é aplicada no segundo passo (*cluster-second*) para dividir a primeira rota em múltiplas rotas viáveis do problema, verificando o balanceamento entre elas para *m*-PRCB. Uma visão geral da rotina em três fases usada neste artigo é descrita a seguir.

### 5.1.1. Fase 1

A escolha dos vértices em $V \setminus T$ pertencentes as rotas a serem percorridas na solução pode ser realizada por meio da resolução do modelo de cobertura de conjunto (32)–(34), cuja solução garante que cada elemento de $W$ deve ser coberto por pelo menos um elemento de $V$ visitado por uma rota.

$$\text{Minimizar} \quad \sum_{v_i \in V \setminus T} b_i y_i \tag{32}$$

$$\text{sujeito a} \quad \sum_{v_i \in S_l} y_i \geq 1, \quad \forall w_l \in W \tag{33}$$

$$y_i \in \{0, 1\}, \quad \forall v_i \in V \setminus T. \tag{34}$$

Em (32)–(34), a variável binária $y_i$ é igual a 1 se o vértice $v_i \in V \setminus T$ é selecionado. Então, o objetivo é minimizar a quantidade de elementos em cada $S_l \subset V \setminus T$ necessários para cobrir todo $W$, onde $S_l$ é um subconjunto de vértices em $V$ que cobrem cada vértice $w_l$. Note que o

parâmetro $b_i$ é um custo associado à variável $y_i$ e, assim, ele pondera a escolha do vértice $v_i$ na solução. Portanto, diferentes ponderações têm potencial de fornecer elementos alternativos de $V \setminus T$ na solução.

Esta primeira fase do algoritmo consiste em gerar aleatoriamente $\theta_1$ subconjuntos de $V \setminus T$ que cobrem todos os vértices de $W$, na qual cada subconjunto é obtido por meio da solução de uma versão do modelo de programação inteira (32)–(34). Escolhemos aleatoriamente $b_i$ em $\{1, 2, 3\}$ com o objetivo de diversificar a quantidade de subconjuntos de $V \setminus T$ que cobrem $W$. No final de cada iteração $k \in \{1, \ldots, \theta_1\}$, o conjunto $N_k \subset V$ contém todos os elementos que são solução de (32)–(34) unidos aos vértices de $T^{'}$. Desta maneira, $N_k$ cobre todos os vértices de $W$. Cada $N_k$ é usado na segunda fase do algoritmo para construir uma solução do problema (um indivíduo da população). Para evitar soluções redundantes, a rotina usa um critério para descartar $N_k$ que já foi gerado previamente.

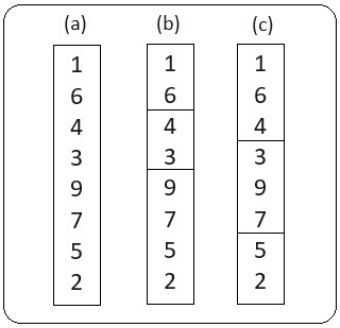
### 5.1.2. Fase 2

Esta fase é baseada nas heurísticas da varredura e *route-first/cluster-second* de Hachicha et al. (2000), que constrói uma solução viável para cada conjunto $N := N_k \subset V$, $k \in \{1, \ldots, \theta_1\}$, obtido na Fase 1. Considere $z = |N|$, primeiramente (*route-first*) é obtida uma única rota grande $R = (h_0, \ldots, h_z)$ contendo vértices $h_i \in N$ e cobrindo $W$. A sequência dada em $R$ é uma lista circular construída por meio de uma modificação da heurística da varredura. Então, $R$ é dividido (*cluster-second*) aleatoriamente em $m$ rotas que atendem uma regra de equilíbrio entre as rotas.

Seja $C_i = \{v_j \in W \mid v_i \in S_h\}$ representando o subconjunto de vértices em *W* que são cobertos por $v_i \in V$, onde $S_h = \{v_l \in V \setminus T \mid c_{lh} \leq c\}$. Então, aplicamos para $T \cup W$ o processo de varredura, gerando várias soluções. O processo consiste em construir rotas particionadas ordenadas de acordo com o menor ângulo das coordenadas polares entre os vértices, no sentido anti-horário. Assim, os vértices são selecionados para pertencer a uma determinada rota, selecionando o vértice mais próximo do vértice selecionado anteriormente em uma varredura similar a abertura de um leque. A rota começa com $h_0 = v_0$, $R = (h_0)$ e $L = T^{'} \cup W$. Seja $\bar{h}$ um vértice aleatório de $L$ e considere uma linha começando em $h_0$ e passando por $\bar{h}$. O conjunto *L* é esvaziado gradualmente usando o critério que varre os vértices $h \in L$ em ordem crescente dos ângulos $\delta_h = \widehat{\bar{h}h_0h}$ utilizando o deslocamento anti-horário dos vértices. O processo de varredura seleciona um vértice de $L$, fazendo a seguinte verificação: a) se $h \in T$, então ele é adicionado a *R*, e $\{h\} \cup C_h$ é removido de *L*; b) se $h \in W$, então é selecionado de $S_h$ o vértice que cobre o maior número de vértices restantes em $W$ ainda não cobertos. Se este vértice escolhido for $v_l$, então ele é adicionado a *R* e $C_l$ é removido de *L*.

Uma vez que *R* é construído, para cada $k$, no máximo $\theta_2^k$ soluções distintas para o *m*-PRCB são produzidas aleatoriamente a partir de *R*. A Figura 4(a) ilustra um exemplo de uma rota grande $R$ composta por 8 vértices gerados pelo processo de varredura (*route-first*), em que $m = 3$ e

Figura 4: Exemplo do processo de divisão de rotas (*cluster-second*) na Fase 2.




Fonte: Autores.

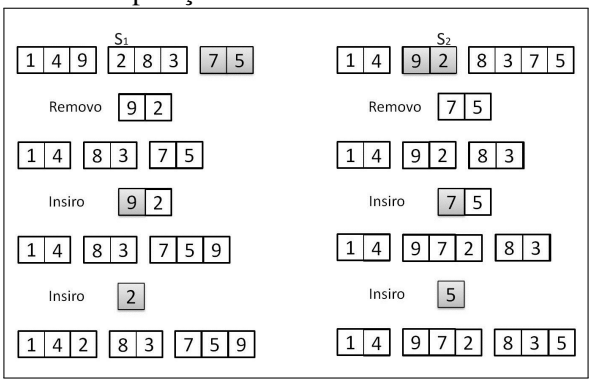
$r = 2$, e os Itens (b) e (c) representam duas soluções possíveis ($\theta_2^k = 2$) geradas aleatoriamente (*cluster-second*), tendo cada uma delas 3 rotas com $\tilde{\rho} - \rho \leq r$. Além disso, a ordem dos vértices na rota grande inicial $R$ é preservada e cada parte (uma rota de veículo) deve ser considerada conectada ao depósito. No final da Fase 2, para cada $k \in \{1, \ldots, \theta_1\}$, no máximo $\theta_3^k$ soluções, as melhores em relação ao custo total das rotas, são selecionadas para formar a população inicial do algoritmo genético. Aqui, cada solução viável torna-se um indivíduo, ou seja, um cromossomo da população. Seja $P$ a população inicial, então $P$ tem tamanho igual a $p = \sum_{k=1}^{\theta_1} \theta_3^k$.

### 5.1.3. Fase 3

A Fase 3 consiste em melhorar a população inicial do algoritmo genético formada por todas as soluções construídas na Fase 2. Para isso, o operador CRMC de Ombuki-Berman e Hanshar (2009) e operações de mutação são usadas. O operador CRMC realiza o *crossover* entre dois indivíduos selecionados aleatoriamente na população, buscando sempre por ordenamentos dos vértices que resultem em menor distância total das rotas.

Cada solução $s_i$ obtida na Fase 2 é adicionada a população inicial $P = \{s_1, \ldots, s_p\}$. O CRMC usa um operador *crossover* e um decodificador indireto baseado na permutação das posições dos vértices, utilizando as restrições do problema como guia. A Figura 5 ilustra um exemplo de *crossover*, em que são selecionados dois indivíduos da população ($s_1$ e $s_2$) e, para cada indivíduo, é selecionada uma das rotas aleatoriamente (a última rota de $s_1$ e a segunda rota de $s_2$). Cada uma dessas rotas é então retirada do outro indivíduo, ou seja, os vértices 7 e 5 são removidos do indivíduo $s_2$ e os vértices 9 e 2 são removidos de $s_1$. Após a retirada dos vértices, verifica-se em qual rota e posição incidem menor custo, onde cada um dos vértices removidos anteriormente será agora reinserido. Em $s_2$, por exemplo, e o vértice 7 é reinserido na Rota 2 entre os vértices 9 e 2 e o vértice 5 é reinserido na Rota 3 entre os vértices 3 e depósito, por serem as inserções que causariam menor impacto no custo. Além disso, a inclusão dos vértices é realizada mantendo-se o equilíbrio entre as rotas, ou seja, caso uma das rotas contenha menos vértices que as demais, esta tem preferência para receber o novo vértice. No final do *crossover* são obtidos indivíduos com balanceamento entre as rotas. Adicionalmente, cada iteração do procedimento de mutação envolve trocas de vértices entre duas rotas na solução que apresentam maior assimetria em termos de quantidade de vértices. Mais especificamente, um vértice da rota maior é removido e o par de arestas ativados desse vértice é transformado em uma única aresta ativa. Em seguida, o vértice removido é inserido na rota menor, na posição que produz redução de custos, alterando as arestas ativas. O vértice removido da rota maior é escolhido de acordo com o comprimento do par de arestas ativas, sendo que o par de arestas ativas mais caro é selecionado.

Figura 5: Exemplo de uma operação de *crossover* entre dois indivíduos da população.




Fonte: Adaptado de Ombuki-Berman e Hanshar (2009).

### 5.1.4. Configuração dos Parâmetros do Algoritmo Genético

Na codificação do algoritmo genético, após extensivos testes computacionais preliminares, os parâmetros foram configurados como segue. A quantidade máxima de vezes $\alpha$, que a rotina busca por rotas com quantidade viável de visitas, atendendo a exigência de balanceamento entre as rotas ($\tilde{\rho} - \rho \leq r$), é dado em função dos dados de entrada, $\alpha = mr$. Além do limite de tempo de processamento (3600 segundos), se a diferença entre a melhor solução da população atual e da população anterior não se altera por pelo menos $\gamma = \lceil |V \setminus T|/\theta_1 \rceil$ iterações, o processo de resolução da instância é interrompido. Os parâmetros $\theta_1$, $\theta_2^k$ e $\theta_3^k$ foram configurados de acordo com os dados de entrada de cada instância, $\theta_1 = \lceil |V \setminus T|/|T| \rceil$, $\theta_2^k = mr$ e $\theta_3^k = |V|/2$, para $k \in \{1, \ldots, \theta_1\}$. A população elite $P_E$ (mantida de uma geração para outra sem modificação) é formada com 30% da população $P$ de cada geração, $p_E = 0,3p$. A quantidade de indivíduos que sofrem mutação é $p_M = |T|$, selecionados entre aqueles indivíduos que não pertencem à população elite, quantidade relativamente pequena, pois poucos indivíduos sofrem mutação e também evita um aumento significativo no tempo computacional. Então, $p - (p_E + p_M)$ indivíduos restantes são resultados de *crossover* entre um indivíduo elite e um não elite.

## 5.2. Método *Branch-and-cut*

A literatura apresenta diversos métodos exatos para variantes do PRV (Bektaş et al., 2019) e alguns utilizam versões do algoritmo *branch-and-cut* para o *m*-PRC (Pham et al., 2017; Hà et al., 2013; Ota et al., 2024), caracterizados por diferentes caminhos para detectar violação de restrições, técnicas de separação e regras de ramificação. Ota et al. (2024) desenvolveram um algoritmo *branch-and-cut* para resolver o *m*-PRC com balanceamento de rotas. Este último é usado neste artigo de forma conjunta com o algoritmo genético na tentativa de determinar soluções exatas para o problema. Algumas desigualdades válidas são aplicadas a fim de reduzir o número de vértices na árvore de busca. Mais especificamente, este algoritmo utiliza cortes (desigualdades válidas) sobre a relaxação linear da formulação matemática (1)–(31) para melhorar os limites duais.

### 5.2.1. Desigualdades Válidas

Esta seção descreve uma série de conjuntos de desigualdades válidas apresentadas por Ota et al. (2024) que foram combinadas e inseridas no método. As desigualdades (35) derivam das relações entre arestas e vértices, ou seja, uma aresta saindo do vértice $v_i$ e uma aresta que chegando no vértice $v_j$ só podem estar ativas quando os vértices $v_i$ e $v_j$ são visitados, respectivamente.

$$x_{ij} \leq y_i \text{ e } x_{ij} \leq y_j, \text{ para } v_i \text{ ou } v_j \in V \setminus T. \tag{35}$$

A relação de dominância entre os vértices definem as desigualdades (36) e (37). Um vértice $v_i$ domina o vértice $v_j$ se $v_i$ pode cobrir todos os vértices de $W$ que $v_j$ pode cobrir. A partir desta definição, as desigualdades de dominância propostas por Gendreau et al. (1997) podem ser aplicadas ao *m*-PRCB: (36) derivam da dominância de cobertura, ou seja, se o vértice $v_i \in V \setminus T$ domina o vértice $v_j \in V \setminus T$ então apenas o vértice $v_i$ poderia pertencer à solução; e (37) estende essa última ideia, quando um subconjunto de dois vértices domina um terceiro vértice, então a dominância de cobertura é aplicável a três vértices.

$$y_i + y_j \leq 1, \text{ se } v_i \text{ domina } v_j \text{ ou vice-versa, } v_i, v_j \in V \setminus T \tag{36}$$

$$y_i + y_j + y_k \leq 2, \text{ se dois vértices entre } v_i, v_j, v_k \text{ dominam outro vértice, } v_i, v_j, v_k \in V \setminus T \tag{37}$$

Seja o seguinte conjunto politopo de cobertura $conv\{y : \sum b_a y_a \geq 1, y_a \in \{0,1\}\}$, onde $b_a$ é o coeficiente binário. Todas as desigualdades válidas deste conjunto são válidas para o *m*-PRCB. As facetas deste politopo com coeficientes em $\{0,1,2\}$ podems ser usadas para gerar o próximo

conjunto de desigualdades (38). Seja *S* um subconjunto não vazio de *W*, então o coeficiente $\alpha_k^S$ é definido para cada $k \in V$: $\alpha_k^S = 0$, se $v_k$ não cobre $w_l$, para todo $w_l \in S$; $\alpha_k^S = 2$, se $v_k$ cobre $w_l$, para todo $w_l \in S$; e $\alpha_k^S = 1$, caso contrário. Assim, (38) é válida para o problema estudado.

$$\sum_{v_k \in V} \alpha_k^S y_k \geq 2 \tag{38}$$

Baldacci et al. (2005) observaram que se uma aresta estiver ativa ($x_{ij} = 1$) ocorre fluxo nela ($f_{ij} > 1$ e $f_{ji} > 1$). Por outro lado, se não ocorrer fluxo, então a aresta não pode ser ativada ($x_{ij} = 0$), derivando assim as desigualdades (39).

$$f_{ij} \geq x_{ij},\ f_{ji} \geq x_{ij},\ \text{para } v_i, v_j \in V \setminus \{0, \bar{0}\} \tag{39}$$

Restrições de capacidade para o PRV, que também eliminam subrotas, podem ser usadas para gerar desigualdades ao adicionar elementos do conjunto *T* e a capacidade $\tilde{\rho}$ da frota de veículos do *m*-PRCB estudado. Cada rota pode visitar no máximo $\tilde{\rho}$ vértices, então o número de veículos necessários para visitar todos os pontos de um suconjunto $S$ de $T$ é determinado nas desigualdades (40).

$$\sum_{v_i, v_j \in S} x_{ij} \leq |S| - \left\lceil \frac{|S|}{\tilde{\rho}} \right\rceil,\ \text{para } S \subseteq T, |S| \geq 2 \tag{40}$$

Finalmente, as desigualdades (41) e (42) são extensões das desigualdades de conectividade de Gendreau et al. (1997). Quando o vértice $v_t \in S$ é visitado, (41) força a conectividade por meio de pelo menos duas arestas ativas entre os conjuntos *S* e $V \setminus S$. Enquanto (42), que tem semelhança algébrica com (41), não permite subrotas em $S$.

$$\sum_{(v_i \in S, v_j \in V \setminus S) \text{ ou } (v_j \in S, v_i \in V \setminus S)} x_{ij} \geq 2y_t \quad \left(S \subset V, 2 \leq |S| \leq n-2, T \setminus S \neq \emptyset, v_t \in S\right) \tag{41}$$

$$\sum_{v_i, v_j \in S} x_{ij} \leq \sum_{v_k \in S \setminus \{v_t\}} y_k \quad (S \subset V,\ 2 \leq |S| \leq n-2,\ T \setminus S \neq \emptyset,\ v_t \in S) \tag{42}$$

### 5.2.2. Algoritmo *Branch-and-cut*

O algoritmo *branch-and-cut* estudado neste trabalho combina as desigualdades anteriores para fortalecer a relaxação linear da formulação (1)–(31). Depois de determinar a solução ótima da relaxação linear, uma busca por restrições violadas do tipo (35)–(42) é realizada, sendo que aquelas detectadas são adicionadas à formulação da relaxação linear atual. Com a adição dessas restrições, o problema de programação linear é então re-otimizado e esse processo continua até que todas as restrições sejam satisfeitas. Em seguida, se se surgir variáveis fracionárias, ramificação é realizada. Por outro lado, se todas as variáveis forem inteiras, outro vértice é explorado.

O separação das restrições (35)–(37) e (39) é mais simples e consideramos uma porcentagem de cada conjunto delas. Da mesma forma que Gendreau et al. (1997), nas desigualdades (38) apenas os conjuntos *S* contendo três elementos foram verificados. A geração das restrições de capacidade (40) foi alcançada por meio do algoritmo ganancioso descrito por Hà et al. (2013), um procedimento iterativo que é aplicado sobre um subconjunto predefinido $T' \subset T$, que repete o seguinte procedimento a cada iteração e para cada $S \in T'$. Seja $v_{i^*} \in T \setminus S$ um vértice tal que $\sum_{v_j \in S}(x_{i^*j} + x_{ji^*}) = \max_{v_i \in T \setminus S} \left[\sum_{v_j \in S}(x_{ij} + x_{ji})\right]$ (uma conexão entre $S$ e $T \setminus S$ por $v_{i^*}$). Se a solução atual $x$ viola (40) em relação ao subconjunto $S' = S \cup \{v_{i^*}\}$, adicione a desigualdade (40) ao modelo, atualize $S$ para $S'$ e repita o processo até que $S$ contenha todos os vértices de $T$. Um único vértice de $T \setminus \{0\}$ foi escolhido para o $T'$ inicial. A inclusão das restrições de

conectividade (41) e (42) foi realizada por meio do método descrito por Gendreau et al. (1997) que gera uma árvore de peso máximo sobre o grafo. Em resumo, o método constrói uma árvore de peso máximo e, ao detectar violações de conectividade, ajusta o conjunto de arestas removendo aquelas que causam desconexão, garantindo assim uma solução conectada e válida para o problema. Essa árvore é construída por um algoritmo ganancioso, de modo que o peso da aresta $\{v_i, v_j\}$ é considerada igual ao valor atual de $x_{ij}$. Em cada etapa da construção da árvore, seja $S$ o conjunto de vértices pertencentes ao componente conectado ao qual a aresta recém incluída pertence. Se $S$ resultar em uma violação da restrição de conectividade, esta restrição adicional é gerada. Quando a árvore estiver completa, realiza-se uma nova verificação de conexões violadas, removendo as arestas que violam as restrições até que toda a conectividade esteja reestabelecida.

### 5.3. Método *Branch-and-cut* com Solução Inicial

Na tentativa de acelerar a convergência do método *branch-and-cut*, a melhor solução obtida obtida com o algoritmo genético foi usada como solução inicial (*MIP starts*) no *Solver* utilizado para rodar o algoritmo *branch-and-cut*. Este procedimento de solução é uma contribuição deste artigo aplicado ao *m*-PRCB estudado, que é denotado por BC/AG-CRMC.

## 6. Experimentos Computacionais

Esta seção descreve uma série de experimentos realizados aplicando os métodos de solução para o *m*-PRCB desenvolvidos neste artigo. As siglas AG-CRMC (algoritmo genético) e BC (algoritmo *branch-and-cut*) são utilizadas para referenciar os resultados das implementações. Além disso, os resultados da abordagem envolvendo a solução final do algoritmo genético como uma solução inicial do algoritmo *branch-and-cut* é referenciada por meio da sigla BC/AG-CRMC.

As rotinas foram codificadas no software de otimização AIMMS® com o CPLEX como *solver*. O algoritmo *branch-and-cut* usou o procedimento *Callback* na verificação das desigualdades válidas violadas. Todos os cortes do CPLEX foram desativados e a regra de ramificação "*Down Branch First*" foi usada. Escolher seguir o *Ramo Inferior Primeiro* em cada vértice para ramificação mostrou ser mais promissor nos testes preliminares. Além disso, todos os outros parâmetros do CPLEX foram configurados como padrão. Um computador pessoal com CPU de 3.40GHz com 32GB de RAM, processador Intel i7 e sistema operacional Windows. O tempo limite de 3600 segundos foi fixado para cada método de solução em cada instância resolvida.

### 6.1. Configuração das Instâncias do Problema

Os experimentos computacionais foram realizados sobre instâncias adaptadas da biblioteca TSPLIB, combinados em cinco classes caracterizadas pelo número total de vértices em $|V \cup W|$ (100, 150, 200, 318 e 400). A Tabela 2 descreve alguns detalhes da configuração das instâncias. Cada classe é subdividida em três subclasses diferenciadas pela cardinalidades de $T$. $|T| = \lfloor |V|/8 \rfloor$, $|T| = \lfloor |V|/4 \rfloor$ e $|T| = \lfloor |V|/2 \rfloor$ definem as Subclasses 1, 2 e 3, respectivamente. Note na tabela que as instâncias testadas ainda são diferenciadas de acordo com o número de rotas $m$ e a diferença máxima $r$ permitida na quantidade de vértices em diferentes rotas. Portanto, a Classe 1 contém 33 instâncias, as Classes 2 e 3 contêm cada uma 18 instâncias e as Classes 4 e 5 contêm cada uma 18 instâncias, totalizando 99 instâncias para validar o método proposto.

Na biblioteca TSPLIB, cada instância pode ser vista como um conjunto de pares $\{(x_i, y_i) |\, i = 1, \ldots, |V| + |W|\}$, onde $x_i$ e $y_i$ são coordenados do plano Euclidiano. O vértice denominado depósito foi escolhido como sendo um dos pares mais centralizados no plano. Os primeiros pares deste conjunto formam os vértices de $T$, seguidos dos vértices de $V \setminus T$ e os demais são selecionados como os vértices de $W$. A sigla C-SC-TSPLIB-m fornece um nome para cada instância, onde C corresponde a cardinalidade $|V| + |W|$ da classe, SC corresponde a cardinalidade $|T|$ da subclasse, o número no lugar de TSPLIB corresponde à instância selecionada na biblioteca TSPLIB e $m$ é

Tabela 2: Descrição das classes de instâncias: 5 classes e 3 subclasses.

| Classe | Instância TSPLIB | $\|V\| + \|W\|$ | $\|V\|$ | $\|W\|$ | $m$ | $r$ |
|---|---|---|---|---|---|---|
| 1 | kro{A,B,C,D,E}100 | 100 | 50 | 50 | $\{2, 3, 4\}$ | 2 |
| 2 | kro{A,B}150 | 150 | 50 | 100 | $\{2, 3, 4\}$ | 2 |
| 3 | kro{A,B}200 | 200 | 100 | 100 | $\{3, 4, 5\}$ | 3 |
| 4 | lin318 | 318 | 100 | 218 | $\{3, 4, 5\}$ | 3 |
| 5 | rd400 | 400 | 200 | 200 | $\{4, 5, 6\}$ | 4 |

Fonte: Autores.

o parâmetro dado para a quantidade de rotas. Assim, ao consultar a Tabela 2, 100-2-4-2 indica a instância da Classe 1 (100 vértices), Subclasse 2 ($\lfloor |V/4| \rfloor$), instância kroD100 da biblioteca TSPLIB e $m = 2$, enquanto 318-3-1-4 indica a instância da Classe 4 (318 vértices), Subclasse 3 ($\lfloor |V|/2 \rfloor$), instância lin318 da biblioteca TSPLIB e $m = 4$.

Adicionalmente, para a implementação das restrições (19)–(30), os parâmetros auxiliares $\tilde{\beta}$ e $\beta$ usados para a linearização das restrições as restrições (6) e (7) foram configurados como $|V|$ e $\lceil |V|/m \rceil$, respectivamente. Apesar de serem escolhas conservadoras em relação a quantidade total de vértices visitados, elas garantem que $\tilde{\rho} \leq \tilde{\beta}$ e $\rho \leq \beta$.

### 6.2. Discussão dos Resultados

As Tabelas 3 e 4 resumem a estatística descritiva dos experimentos para cada instância e método de solução por meio do tempo computacional (colunas T(s)), o custo total da solução encontrada (colunas Custo), os gaps de otimalidade da solução (colunas GAP(%)) e a diferença de fato obtida no número de vértices entre as diferentes rotas (colunas $\tilde{r}$). A primeira coluna de cada tabela informa a instância, enquanto as demais colunas são destacadas em três blocos, cada bloco contendo os resultados dos métodos, onde AG-CRMC indica o algoritmo genético, BC indica o algoritmo *branch-and-cut* e BC/AG-CRMC indica o método que combina o BC com o AG-CRM, ou seja, o algoritmo *branch-and-cut* usando como solução inicial a melhor solução obtida com o algoritmo genético. O gap de solução é calculado usando a fórmula $GAP := \frac{|C_{VRL} - C_{VI}|}{C_{VRL}} \times 100$, onde $C_{VI}$ e $C_{VRL}$ são os valores da melhor solução inteira obtida e da solução da relaxação linear encontrada pelo *Solver*, respectivamente. Além disso, células vazias na tabela indicam que o respectivo método de solução não encontrou uma solução viável para a instância dentro do limite de tempo computacional estipulado. Observa-se que não ocorreu interrupções do algoritmo por falta de memória computacional.

A Tabela 3 mostra que para a Classe 1, considerando o tempo limite, o método BC (Ota et al., 2024) não encontrou uma solução viável para apenas 3 instâncias. Por outro lado, os métodos AG-CRMC e BC/AG-CRMC encontraram soluções viáveis para todas as instâncias. Os métodos que apresentam a melhor solução para cada instância têm o valor Custo (melhor solução) destacado. Nesse quesito, enquanto o BC venceu em $42,2\%$ o BC/AG-CRMC venceu em $55,5\%$ das instâncias, tendo ocorrido um empate. Comparando a média geral dos valores para Custo em cada método, BC/AG-CRMC performou melhor, de forma que foi $0,65\%$ e $54,98\%$ melhor que os métodos BC e AG-CRMC, respectivamente. Em relação aos tempos computacionais, a variação entre BC e BC/AG-CRMC não é significativa, uma vez que ambos os métodos utilizaram todo tempo disponibilizado para a maioria das instâncias. No entanto, AG-CRMC é relativamente rápido, uma vez que não superou 63 segundos em cada instância. Observa-se que os tempos computacionais relativamente baixos se justificam devido ao critério de parada de não avanço da melhor solução entre populações consecutivas após $\gamma$ iterações. Em relação aos gaps de otimalidade da solução, os valores encontrados são bem parecidos e equivalem em média a $21,8\%$ e $22,0\%$ para BC/AG-CRMC e BC, respectivamente. Por fim, o resultado do balanceamento entre rotas

Tabela 3: Resultados dos métodos de solução para a Classe 1.

| Instância | AG-CRMC | | | BC | | | | BC/AG-CRMC | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | T(s) | Custo | $\tilde{r}$ | T(s) | Custo | $GAP$ (%) | $\tilde{r}$ | T(s) | Custo | $GAP$ (%) | $\tilde{r}$ |
| 100-1-1-2 | 14 | 13425 | 1 | 3600 | **9363** | 15,00 | 2 | 3600 | 9460 | 11,85 | 1 |
| 100-1-1-3 | 31 | 12516 | 2 | 3600 | **11312** | 20,56 | 2 | 3600 | 12041 | 27,60 | 2 |
| 100-1-1-4 | 32 | 14402 | 2 | 3600 | 13391 | 24,72 | 2 | 3600 | **13225** | 25,28 | 2 |
| 100-1-2-2 | 14 | 13817 | 1 | 3600 | 9458 | 22,22 | 1 | 3600 | **9420** | 22,42 | 1 |
| 100-1-2-3 | 20 | 13948 | 2 | 3600 | **12052** | 31,52 | 2 | 3600 | 12656 | 34,75 | 2 |
| 100-1-2-4 | 31 | 15201 | 2 | 3600 | 13973 | 32,65 | 2 | 3600 | **13258** | 28,69 | 2 |
| 100-1-3-2 | 12 | 10933 | 1 | 3600 | 8897 | 14,43 | 1 | 3600 | **8761** | 13,30 | 1 |
| 100-1-3-3 | 20 | 12326 | 1 | 3600 | **10565** | 23,74 | 2 | 3600 | 10956 | 24,98 | 2 |
| 100-1-3-4 | 62 | 12212 | 2 | 3600 | **11090** | 11,54 | 2 | 3600 | 11794 | 19,63 | 2 |
| 100-1-4-2 | 37 | 14181 | 2 | 3600 | **9991** | 12,70 | 1 | 3600 | 11273 | 22,73 | 1 |
| 100-1-4-3 | 35 | 16113 | 2 | 3600 | 11734 | 21,59 | 1 | 3600 | **11704** | 21,93 | 1 |
| 100-1-4-4 | 31 | 16988 | 2 | 3600 | **13865** | 28,60 | 2 | 3600 | 14145 | 28,73 | 2 |
| 100-1-5-2 | 15 | 12875 | 2 | 3600 | **9601** | 27,57 | 1 | 3600 | 9854 | 23,92 | 1 |
| 100-1-5-3 | 20 | 14941 | 0 | 3600 | **11674** | 31,74 | 2 | 3600 | 12076 | 35,77 | 2 |
| 100-1-5-4 | 24 | 14888 | 2 | 3600 | 14700 | 43,47 | 2 | 3600 | **13535** | 35,39 | 2 |
| 100-2-1-2 | 21 | 18643 | 1 | 3600 | **10594** | 5,43 | 0 | 3600 | 10634 | 8,47 | 0 |
| 100-2-1-3 | 22 | 16630 | 1 | 3600 | **13230** | 21,12 | 1 | 3600 | 13450 | 22,60 | 2 |
| 100-2-1-4 | 39 | 17241 | 2 | 3600 | 18851 | 38,14 | 1 | 3600 | **14698** | 19,48 | 2 |
| 100-2-2-2 | 44 | 18588 | 2 | 3600 | **10823** | 17,10 | 2 | 3600 | 11266 | 22,39 | 2 |
| 100-2-2-3 | 34 | 17215 | 1 | 3600 | 12100 | 19,70 | 2 | 3600 | **12069** | 21,43 | 2 |
| 100-2-2-4 | 28 | 17817 | 2 | 3600 | 14613 | 29,70 | 1 | 3600 | **13515** | 23,54 | 2 |
| 100-2-3-2 | 20 | 17634 | 2 | 3600 | **11764** | 10,82 | 2 | 3600 | 11871 | 10,96 | 2 |
| 100-2-3-3 | 24 | 20930 | 1 | 3600 | 15077 | 24,86 | 2 | 3600 | **14302** | 21,93 | 1 |
| 100-2-3-4 | 29 | 21473 | 2 | 3600 | 16312 | 25,55 | 2 | 3600 | **16104** | 25,03 | 2 |
| 100-2-4-2 | 27 | 19277 | 2 | 3600 | **10696** | 7,21 | 1 | 3600 | 11183 | 13,34 | 1 |
| 100-2-4-3 | 25 | 17323 | 2 | 3600 | **13284** | 20,42 | 2 | 3600 | 13595 | 24,98 | 2 |
| 100-2-4-4 | 41 | 20456 | 2 | 3600 | 16051 | 30,26 | 2 | 3600 | **15624** | 28,48 | 2 |
| 100-2-5-2 | 23 | 19143 | 2 | 3600 | 11091 | 16,85 | 1 | 3600 | **10609** | 13,03 | 2 |
| 100-2-5-3 | 23 | 18497 | 1 | 3600 | 13106 | 28,64 | 2 | 3600 | **12591** | 24,57 | 2 |
| 100-2-5-4 | 33 | 18776 | 2 | 3600 | 15654 | 35,60 | 2 | 3600 | **15240** | 33,86 | 2 |
| 100-3-1-2 | 34 | 28501 | 1 | 3600 | **12749** | 11,63 | 2 | 3600 | 12903 | 11,53 | 0 |
| 100-3-1-3 | 36 | 29318 | 2 | 3600 | 16702 | 27,81 | 2 | 3600 | **16135** | 21,99 | 2 |
| 100-3-1-4 | 38 | 27842 | 2 | | | | | 3600 | **19705** | 34,12 | 2 |
| 100-3-2-2 | 35 | 30954 | 1 | 3600 | 12929 | 6,61 | 2 | 3600 | **12863** | 4,26 | 1 |
| 100-3-2-3 | 36 | 34129 | 1 | 3600 | 17875 | 28,28 | 2 | 3600 | **17803** | 28,91 | 2 |
| 100-3-2-4 | 40 | 27072 | 2 | 3600 | 18314 | 25,48 | 2 | 3600 | **17778** | 23,18 | 2 |
| 100-3-3-2 | 59 | 34379 | 2 | 3600 | 15027 | 11,09 | 1 | 3600 | **14918** | 10,04 | 0 |
| 100-3-3-3 | 35 | 32115 | 2 | 3600 | 18537 | 24,38 | 2 | 3600 | **17633** | 19,73 | 2 |
| 100-3-3-4 | 45 | 31457 | 2 | 3600 | **19575** | 22,06 | 1 | 3600 | 21143 | 27,42 | 2 |
| 100-3-4-2 | 48 | 36692 | 2 | 3600 | 14380 | 14,24 | 1 | 3600 | **14264** | 13,72 | 2 |
| 100-3-4-3 | 41 | 32381 | 2 | 3600 | **17542** | 27,76 | 2 | 3600 | 18359 | 31,56 | 2 |
| 100-3-4-4 | 55 | 30727 | 1 | | | | | 3600 | **19317** | 28,96 | 2 |
| 100-3-5-2 | 63 | 37019 | 0 | 865 | **12859** | 0,00 | 0 | 2631 | **12859** | 0,00 | 0 |
| 100-3-5-3 | 37 | 35194 | 2 | 3600 | **18536** | 30,91 | 2 | 3600 | 18702 | 31,15 | 2 |
| 100-3-5-4 | 36 | 30637 | 2 | | | | | 3600 | **18020** | 24,48 | 2 |

Fonte: Autores.

dado por $\tilde{r}$ foi na maioria das vezes igual a 2, com média igual a 1,62, 1,60, e 1,62 para AG-CRMC, BC, e BC/AG-CRMC nas soluções, respectivamente, ou seja, por essa média os métodos têm comportamento similar.

A Tabela 4 mostra que o método BC (Ota et al., 2024) não conseguiu encontrar solução viável para 3 das 18 instâncias da Classe 2 e não encontrou nenhuma solução viável para as instâncias das Classes 3, 4 e 5 (maiores dimensionalmente), mostrando a dificuldade do BC em resolver instâncias relativamente grandes. Por outro lado, semelhante aos resultados apresentados na Tabela 3, o método AG-CRMC encontrou soluções viáveis para todas as instâncias. Em relação a qualidade das soluções medida pelo valor Custo na Classe 2, enquanto BC performou melhor em $38,8\%$ das instâncias, BC/AG-CRMC performou melhor $55,5\%$. Observe que BC/AG-CRMC encontrou a melhor solução para $100\%$ das instâncias nas Classes 3 e 4, enquanto o método BC não conseguiu encontrar soluções viáveis para essas classes. Por fim, para a Classe 5, o método AG-CRMC encontrou a melhor solução para $44,4\%$ das instâncias. Considerando os tempos computacionais, BC e BC/AG-CRMC novamente não apresentam diferença significativa, pois esgotam todo o tempo computacional disponível na maioria das instâncias. No entanto, os tempos computacionais apresentados pelo método AG-CRMC são novamente melhores, para os quais o tempo médio é de 38, 179, 208 e 795 segundos para as Classes 2, 3, 4 e 5, respectivamente. Adicionalmente, o resultado do balanceamento entre rotas dado por $\tilde{r}$ foi na maioria das vezes igual a 3, com média igual a 2,5, 1,53 e 2,52 para AG-CRMC, BC, e BC/AG-CRMC nas soluções, respectivamente,

Tabela 4: Resultados dos métodos de solução para as Classes 2-5.

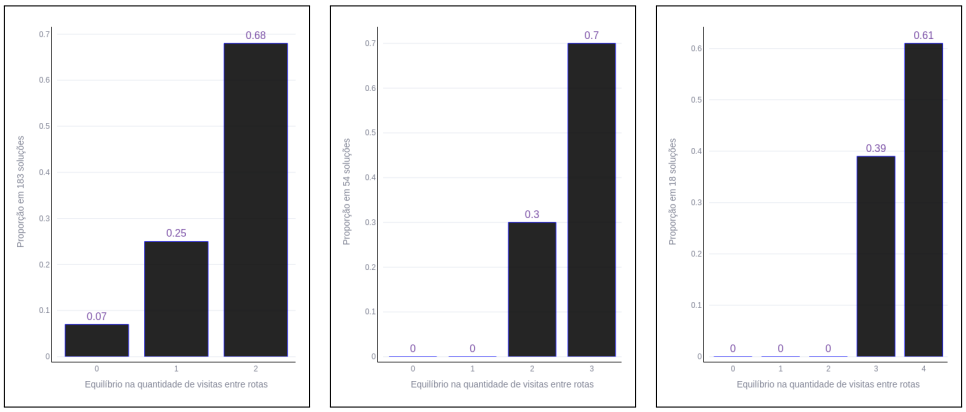
| Instância | AG-CRMC | | | BC | | | | BC/AG-CRMC | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | T(s) | Custo | $\tilde{r}$ | T(s) | Custo | $GAP$ (%) | $\tilde{r}$ | T(s) | Custo | $GAP$ (%) | $\tilde{r}$ |
| 150-1-1-2 | 33 | 14803 | 2 | 3600 | 9302 | 10,83 | 2 | 3600 | **9146** | 13,82 | 1 |
| 150-1-1-3 | 27 | 13355 | 2 | 3600 | **11414** | 22,94 | 2 | 3600 | 11769 | 25,79 | 2 |
| 150-1-1-4 | 46 | **12991** | 2 | 3600 | 13723 | 30,06 | 2 | 3600 | **12991** | 28,51 | 2 |
| 150-1-2-2 | 40 | 11420 | 2 | 3600 | **8974** | 12,83 | 0 | 3600 | 9025 | 10,07 | 1 |
| 150-1-2-3 | 30 | 11231 | 2 | 3600 | 10526 | 15,93 | 2 | 3600 | **10165** | 11,71 | 2 |
| 150-1-2-4 | 24 | 12507 | 2 | 3600 | **11591** | 15,08 | 2 | 3600 | 11978 | 18,28 | 2 |
| 150-2-1-2 | 20 | 19022 | 1 | 3600 | **10485** | 8,89 | 0 | 3600 | 10743 | 7,96 | 0 |
| 150-2-1-3 | 23 | 18801 | 2 | 3600 | 15288 | 31,92 | 1 | 3600 | **14338** | 28,34 | 2 |
| 150-2-1-4 | 49 | 15742 | 2 | 3600 | 15155 | 25,06 | 2 | 3600 | **14282** | 19,02 | 2 |
| 150-2-2-2 | 22 | 19099 | 1 | 3600 | **11808** | 11,92 | 2 | 3600 | 11937 | 12,29 | 2 |
| 150-2-2-3 | 40 | 19421 | 2 | 3600 | 14182 | 20,44 | 2 | 3600 | **14124** | 19,70 | 1 |
| 150-2-2-4 | 41 | 20090 | 2 | 3600 | 17676 | 33,24 | 0 | 3600 | **16106** | 26,08 | 2 |
| 150-3-1-2 | 32 | 27175 | 1 | 3600 | **12388** | 7,17 | 2 | 3600 | 12437 | 6,55 | 2 |
| 150-3-1-3 | 36 | 26470 | 1 | 3600 | 15267 | 20,58 | 2 | 3600 | **15159** | 19,89 | 2 |
| 150-3-1-4 | 41 | 25891 | 1 | 3600 | **17059** | 19,79 | 2 | 3600 | 17888 | 24,17 | 1 |
| 150-3-2-2 | 55 | 32538 | 2 | | | | | 3600 | **14579** | 8,30 | 2 |
| 150-3-2-3 | 39 | 28708 | 1 | | | | | 3603 | **18102** | 21,30 | 2 |
| 150-3-2-4 | 85 | 29462 | 2 | | | | | 3600 | **19202** | 21,09 | 2 |
| 200-1-1-3 | 85 | 18371 | 2 | | | | | 3600 | **15140** | 48,22 | 3 |
| 200-1-1-4 | 114 | **19936** | 3 | | | | | 3600 | **19936** | 57,55 | 3 |
| 200-1-1-5 | 196 | **19847** | 3 | | | | | 3600 | **19847** | 53,19 | 3 |
| 200-1-2-3 | 115 | 33179 | 3 | | | | | 3600 | **23318** | 60,45 | 3 |
| 200-1-2-4 | 141 | 27638 | 3 | | | | | 3600 | **21621** | 55,36 | 3 |
| 200-1-2-5 | 177 | **26795** | 3 | | | | | 3600 | **26795** | 61,48 | 3 |
| 200-2-1-3 | 163 | 34410 | 2 | | | | | 3600 | **19216** | 39,81 | 3 |
| 200-2-1-4 | 164 | 30672 | 2 | | | | | 3600 | **21858** | 43,08 | 2 |
| 200-2-1-5 | 263 | 34000 | 3 | | | | | 3600 | **26906** | 50,64 | 3 |
| 200-2-2-3 | 145 | 41468 | 3 | | | | | 3600 | **26301** | 57,74 | 3 |
| 200-2-2-4 | 148 | 35447 | 2 | | | | | 3600 | **34984** | 66,75 | 2 |
| 200-2-2-5 | 164 | 36358 | 3 | | | | | 3600 | **27314** | 55,27 | 3 |
| 200-3-1-3 | 240 | 60258 | 2 | | | | | 3600 | **35550** | 56,88 | 2 |
| 200-3-1-4 | 244 | **57906** | 3 | | | | | 3600 | **57906** | 72,16 | 3 |
| 200-3-1-5 | 119 | 56449 | 3 | | | | | 3600 | **45645** | 62,87 | 3 |
| 200-3-2-3 | 345 | 65336 | 2 | | | | | 3600 | **64253** | 76,00 | 3 |
| 200-3-2-4 | 236 | 64580 | 2 | | | | | 3600 | **61346** | 73,97 | 2 |
| 200-3-2-5 | 167 | 56957 | 3 | | | | | 3600 | **54247** | 69,55 | 2 |
| 318-1-1-3 | 80 | **12796** | 3 | | | | | 3600 | **12796** | 29,71 | 3 |
| 318-1-1-4 | 141 | **16140** | 3 | | | | | 3600 | **16140** | 26,84 | 3 |
| 318-1-1-5 | 138 | 18141 | 3 | | | | | 3600 | **17967** | 18,04 | 3 |
| 318-2-1-3 | 134 | 16347 | 2 | | | | | 3600 | **14076** | 26,83 | 3 |
| 318-2-1-4 | 258 | 19978 | 3 | | | | | 3600 | **17958** | 29,18 | 3 |
| 318-2-1-5 | 239 | 22861 | 3 | | | | | 3600 | **21212** | 27,10 | 2 |
| 318-3-1-3 | 276 | 30484 | 2 | | | | | 3600 | **18645** | 38,86 | 2 |
| 318-3-1-4 | 339 | 33211 | 3 | | | | | 3600 | **23215** | 42,39 | 3 |
| 318-3-1-5 | 271 | 33228 | 3 | | | | | 3600 | **27989** | 43,59 | 3 |
| 400-1-1-4 | 1441 | **17227** | 4 | | | | | 3600 | **17227** | 77,32 | 4 |
| 400-1-1-5 | 1290 | 17329 | 3 | | | | | 3600 | **16209** | 74,89 | 4 |
| 400-1-1-6 | 1313 | 16714 | 4 | | | | | 3600 | **15052** | 71,59 | 3 |
| 400-2-1-4 | 1128 | 26680 | 4 | | | | | 3600 | **25957** | 79,69 | 4 |
| 400-2-1-5 | 397 | 26872 | 4 | | | | | 3600 | **26806** | 79,90 | 4 |
| 400-2-1-6 | 251 | 26078 | 4 | | | | | 3600 | **24261** | 76,99 | 3 |
| 400-3-1-4 | 915 | **40656** | 3 | | | | | 3600 | **40656** | 76,30 | 3 |
| 400-3-1-5 | 278 | **41977** | 3 | | | | | 3600 | **41977** | 77,90 | 3 |
| 400-3-1-6 | 141 | **44777** | 4 | | | | | 3600 | **44777** | 78,06 | 4 |

Fonte: Autores.

ou seja, por essa média BC obteve mais equilíbrio para aquelas soluções alcançadas, enquanto AG-CRMC e BC/AG-CRMC têm comportamento similar.

A última coluna das Tabelas 3 e 4 pode ser usada para analisar os resultados numéricos do ponto de vista do equilíbrio atingido entre as rotas na solução final. As células vazias nas tabelas indicam que o respectivo método de solução não encontrou uma solução viável para a instância dentro do limite de tempo computacional estipulado, logo todos os experimentos retornaram no final um total de 255 soluções viáveis. Cada gráfico de barras da Figura 6 ilustra a proporção (no eixo $y$) de ocorrência para o valor $\tilde{r}$ (no eixo $x$) em relação ao total de soluções obtidas, ou seja, a proporção de $\tilde{r}$ que atinge o limite de balanceamento $r$ exigido em cada instância, sendo que as Classes 1 e 2, Classes 3 e 4 e Classes 5 têm a restrição de balanceamento com $r = 2, 3, 4$, respectivamente. Note na Figura 6(a) que $\tilde{r} = 0$, $\tilde{r} = 1$ e $\tilde{r} = 2$ aparecem nas proporções 7%, 25% e 68% do total de 183 soluções, respectivamente. Similarmente, note na Figura 6(b) que $\tilde{r} = 0$, $\tilde{r} = 1$, $\tilde{r} = 2$ e $\tilde{r} = 3$ aparecem nas proporções 0%, 0%, 30% e 70% do total de 54 soluções e, na Figura 6(c), $\tilde{r} = 0$, $\tilde{r} = 1$, $\tilde{r} = 2$, $\tilde{r} = 3$ e $\tilde{r} = 4$ aparecem nas proporções 0%, 0%, 0%, 39% e

Figura 6: Análise entre o balanceamento encontrado $\tilde{r}$ e o balanceamento permitido $r$.




(a) Classes 1 e 2: $r = 2$ (b) Classes 3 e 4: $r = 3$ (c) Classes 5: $r = 4$

Fonte: Autores.

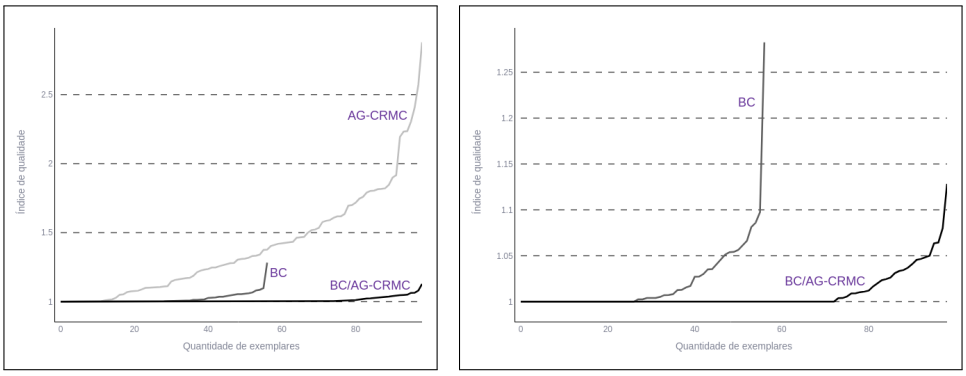
61% do total de 18 soluções. Geralmente, ao desejar soluções com algum tipo de balanceamento ou equidade (*fairness*) em problemas de tomada de decisão, os custos gerenciais aumentam quanto mais equidade é exigida e vice-versa. Note nos gráficos da Figura 6 que de forma significativa o limite de equilíbrio $r$ é alcançado pelo valor $\tilde{r}$, ou seja, significativamente as soluções que atingem maiores valores para $\tilde{r}$ são escolhidas pelos métodos como solução final, ou seja, aquelas soluções que apresentam relativamente pouco equilíbrio (maior valor para $\tilde{r}$), as quais são exatamente as soluções com custo de função objetivo menor. Logo, mais balanceamento (menor valor para $\tilde{r}$), implica em mais custos (distância total das rotas) e vice-versa.

### 6.3. Síntese do Comportamento Numérico dos Métodos de Solução

Esta seção apresenta um estudo sobre o perfil de desempenho dos métodos de solução e, brevemente, sobre o comportamento do algoritmo genético em relação ao seu fator aleatório por meio de simulações.

A comparação principal é realizada por meio do índice de qualidade (IQ) do método de solução, ou seja, o custo (tamanho total das rotas) do método de solução dividido pelo menor custo obtido entre todos os métodos em cada experimento, mostrando assim o quão bom foi o desempenho daquele método em comparação com o de menor custo. Portanto, o IQ do método com melhor custo é igual 1, os demais têm IQ maior que 1. A Figura 7 ilustra um tipo de perfil de desempenho que compara os métodos de solução por meio do IQ. Em cada gráfico da figura, o valor do IQ aparece no eixo $y$ e a quantidade de instâncias resolvidas aparece no eixo $x$. Cada curva nos gráficos, uma para cada método, é obtida calculando o IQ do método para cada instância e depois os valores obtidos são ordenados em ordem crescente, iniciando em IQ=1, o que fornece o perfil de desempenho apresentado na figura. Quanto mais IQ é próximo de 1 ao longo do eixo $x$, melhor foi o desempenho do método. A Figura 7(a) ilustra o perfil de desempenho para os três métodos de solução. Note que AG-CRMC, BC e BC/AG-CRMC obtiveram melhor performance em aproximadamente 15, 37, e 78 instâncias em um total de 99 instâncias, respectivamente. Então, AG-CRMC teve IQ igual a 1 na maioria das instâncias. Adicionalmente, o valor IQ para o método AG-CRMC tem valor menor frequentemente ao longo do eixo $x$ e BC não é capaz de resolver todas as instâncias, pois a sua curva de perfil de desempenho encerra na quantidade de instância $x = 57$. Note que nas Tabelas 3 e 4 aparecem um total de 42 entradas vazias para o método BC. Os demais métodos apresentam soluções viáveis para todas as instâncias. A Figura 7(b) ilustra o impacto em acrescentar as soluções do AG-CRMC no método BC. Essa forma conjunta de solução permitiu que o BC fosse capaz de resolver todas as instâncias, aumentando de aproximadamente 37 para 78 instâncias com melhor desempenho (IQ=1) e o IQ do BC diminuiu, no pior caso, de 1.28 para 1.13,

Figura 7: Perfil de desempenho para os métodos de solução gerados pelo IQ.




(a) Perfil de desempenho dos três métodos (b) Impacto do AG-CRMC sobre o BC

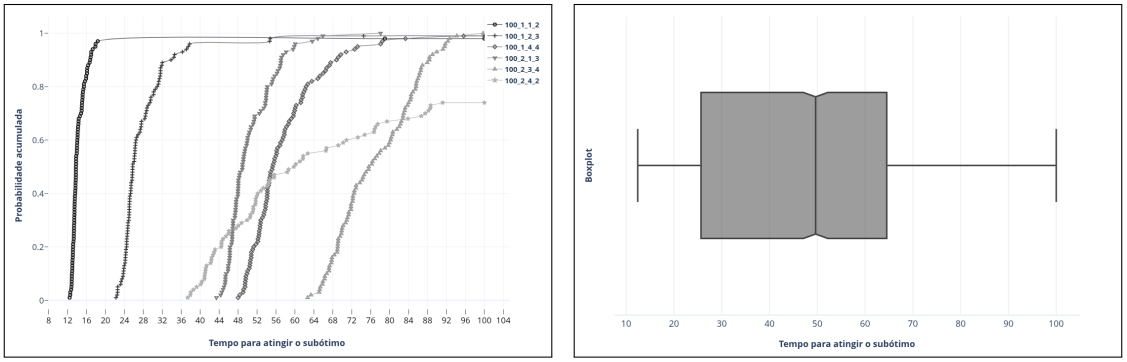
Fonte: Autores.

ou seja, no pior caso a inclusão do AG-CRMC no método BC melhorou a função objetivo em 15%.

A técnica *time-to-target plots* (TTTplots) descrita por Reyes e Ribeiro (2018) é uma excelente ferramenta complementar para avaliar resultados do ponto de vista estatístico. Ela é usada neste artigo para analisar, por meio de simulações, o comportamento do algoritmo genético em relação ao seu fator aleatório. O TTTplot é um gráfico descritivo que apresenta no eixo das ordenadas uma função de distribuição acumulada do tempo de processamento (eixo das abscissas) que um algoritmo gasta para atingir uma solução de qualidade mínima pré-definida (*target*). Uma amostra da Classe 1 de instâncias foi selecionada para avaliar o comportamento do algoritmo genético. Foram escolhidas as seis instâncias (100-1-1-2, 100-1-2-3, 100-1-4-4, 100-2-1-3, 100-2-3-4, 100-2-4-2), com os respectivos valores de custo (9363, 12052, 13865, 13230, 16312, 10696) obtidos pelo método BC sendo utilizados como *targets*. Considerou-se uma variação de $+10\%$ em relação ao *target* para gerar o gráfico TTTplot, por meio de 100 simulações do método AG-CRMC e com tempo máximo limitado a 100 segundos por simulação. Incluir a variação do *target* é importante numericamente, para evitar que o *target* seja atingido na exatidão, ou seja, cada simulação do algorítimo genético foi realizada até o valor da função objetivo atingir valor menor ou igual a $1{,}1 \times target$. Quando esse limiar é atingido, a simulação registra o tempo. Caso contrário, consideramos que o exemplar não foi resolvido nessa simulação e uma nova simulação é realizada.

A Figura 8 ilustra os resultados das simulações realizadas na amostra de instâncias. Ao considerar o tempo de processamento menor ou igual a 100 segundos e valor de custo menor ou igual a $1{,}1 \times target$, a distribuição do tempo de processamento para cada instância foi registrada no gráfico TTTplot da Figura 8(a) e para todas as instâncias no gráfico boxplot da Figura 8(b).

Figura 8: Estatísticas para avaliar o comportamento do método AG-CRMC.




(a) Gráfico TTTplot para o AG-CRMC (b) Gráfico Boxplot para o AG-CRMC

Fonte: Autores.

No gráfico TTTplot, curvas que sobem mais rápido (à esquerda) indicam melhor desempenho, enquanto uma curva que atinge o valor 1 mais cedo mostra que todas as execuções foram eficientes. Nesse sentido, observamos que o algoritmo performou bem em 5 instâncias, com destaque para a Instância 100-1-1-2. Por outro lado, a Instância 100-2-4-2 foi resolvida 75% das simulações antes de 100 segundos. As curvas mostram que a configuração dos parâmetros e o fator aleatório do algoritmo genético permitem atingir uma proximidade do *target* dentro do tempo estipulado. Além disso, o crescimento acelerado das curvas mostram uma diferença relativamente pequena entre os tempos de execução de cada simulação para resolver a instância, isso reforça que o fator aleatório do algoritmo não tem influência negativa na capacidade de resolução, mostrando consistência do algoritmo. O gráfico boxplot da Figura 8(b) também sugere que o desempenho típico (mediana) do tempo de resolução do algoritmo, considerando todas as instâncias, é de um pouco menos de 50 segundos. Os quartis (25% e 75%) indicam a dispersão central dos tempos de resolução nas simulações, sendo que na caixa mais curta, a variação no tempo de execução é menor, indicando que a resolução não vai superar 65 segundos em 75% dos casos. Por outro lado, a caixa mais larga mostra que 25% das simulações teve tempo de resolução variando entre 25 e 50 segundos. Além disso, observa-se que os extremos no tempo são um pouco mais de 10 segundos e obviamente o limite estabelecido de 100 segundos.

## 7. Conclusões

Este artigo estudou uma variante do problema de rotas de cobertura multiveículo, que consiste em determinar um conjunto de rotas para uma frota de veículos disponível, de tal forma que a distância total percorrida seja mínima e as rotas utilizadas fiquem balanceadas. Neste problema, cada vértice representa uma localização de interesse e esses vértices são divididos em três conjuntos: vértices que podem ser visitados; vértices que devem ser visitados; e vértices que devem ser cobertos, porém não visitados, no sentido de que cada um deles deve estar próximo de um vértice visitado por uma rota. Para lidar com a dificuldade computacional em encontrar soluções do problema, três métodos de solução foram utilizados: um algoritmo genético codificado com o operador Crossover de Rota de Melhor Custo; um algoritmo *branch-and-cut* codificado com várias desigualdades válidas; e um método que combina essas duas técnicas, em que a melhor solução encontrada pelo algoritmo genético é incorporada no algoritmo *branch-and-cut* como uma solução inicial. As contribuições deste artigo consistem do estudo e aplicação de métodos de solução para o *m*-PRC com equilíbrio nas visitas, incluindo a proposta de um algoritmo genético, o estudo de um método *branch-and-cut* da literatura e a proposta e análise do método *branch-and-cut* usando a solução do algoritmo genético como solução inicial. O algoritmo genético desenvolvido agrega de maneira prática três rotinas difundidas na literatura.

A análise dos resultados computacionais mostraram que o algoritmo *branch-and-cut* não teve sucesso em encontrar soluções viáveis para as instâncias dimensionalmente grandes. No entanto, o algoritmo genético foi capaz de encontrar soluções viáveis para todas as instâncias com tempo computacional satisfatório. Além disso, por meio da curva de perfil de desempenho, os resultados mostraram o impacto em relação a qualidade das soluções quando a melhor solução encontrada pelo algoritmo genético é utilizada como solução inicial (*MIP start*) ao rodar o algoritmo exato. Essa forma conjunta de solução permitiu que o método *branch-and-cut* proposto fosse capaz de resolver todas as instâncias e, na métrica proposta chamada de índice de qualidade, aumentasse de aproximadamente 37% para 79% das instâncias com melhor desempenho. Além disso, no pior caso, melhorou a função objetivo em 15%. Os resultados também demonstraram que o limite estipulado de balanceamento foi alcançado, o que significa que quanto maior for o balanceamento (menor diferença entre as rotas) maior será o custo (tamanho total das rotas).

Existem ainda várias questões interessantes que podem ser examinadas em pesquisas futuras. Por exemplo, aprimorar as diferentes fases do algoritmo genético investigando oportunidades na exploração do espaço de busca e na calibração dos parâmetros. Mais especificamente, analisar

novas alternativas nos critérios de parada, seja permitindo retornos à fase inicial, com o objetivo de aumentar a diversificação de componentes das soluções, seja relaxando os limites de iterações sem a ocorrência de melhoria na solução, a fim de favorecer uma exploração mais intensiva. Também seria interessante investigar a aplicação de outros operadores de troca de características entre indivíduos da população. Além disso, outra possibilidade a ser explorada é a aplicação da relaxação lagrangiana, que pode permitir a decomposição do problema e, assim, viabilizar a obtenção de soluções interessantes do ponto de vista da velocidade computacional. Uma extensão final consistiria em estudar o problema sob uma perspectiva multiobjetivo, de modo a capturar o conflito identificado entre a minimização do custo total das rotas e o balanceamento entre elas.

***Agradecimentos.*** Este trabalho foi apoiado pelo CNPq (processos nºs 309925/2021-5, 405048/2018-1 e 304561/2021-5), FAPESP (processos nºs 2021/09386-5, 2022/05803-3, e BI0S – *Brazilian Institute of Data Science*, processo nº 2020/09838-0), e FAEPEX/Unicamp (processo nº 2396/23).

## Referências

Baldacci, R., Boschetti, M. A., Maniezzo, V., e Zamboni, M. Scatter search methods for the covering tour problem. In: Sharda, R., Voß, S., Rego, C., e Alidaee, B. (ee.), *Metaheuristic Optimization via Memory and Evolution,*p. 59–91. Boston, MA: Springer US, 2005.

Baldacci, R., Hadjiconstantinou, E., e Mingozzi, A. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations research*, v. 52, n. 5, p. 723–738, 2004.

Bektaş, T., Gouveia, L., Martinez-Sykora, A., e Salazar-González, J. J. Balanced vehicle routing: Polyhedral analysis and branch-and-cut algorithm. *European Journal of Operational Research*, v. 273, n. 2, p. 452–463, 2019.

Bektaş, T. e Letchford, A. N. Using $\ell^p$-norms for fairness in combinatorial optimisation. *Computers & Operations Research*, v. 120, p. 104975, 2020.

Current, J. R. *Multiobjective design of transportation networks*. . Tese de Doutorado. Department of Geography and Environmental Engineering, The Johns Hopkins University, Baltimore, 1981.

Doerner, K. F. e Hartl, R. F. Health care logistics, emergency preparedness, and disaster relief: new challenges for routing problems with a focus on the Austrian situation. In: *The vehicle routing problem: latest advances and new challenges*p. 527–550. : Springer, 2008.

Dutta, J., Barma, P. S., Mukherjee, A., Kar, S., e De, T. A hybrid multi-objective evolutionary algorithm for open vehicle routing problem through cluster primary-route secondary approach. *International Journal of Management Science and Engineering Management*, v. 17, n. 2, p. 132–146, 2022.

Feng, B. e Wei, L. An improved multi-directional local search algorithm for vehicle routing problem with time windows and route balance. *Applied Intelligence*, v. 53, n. 10, p. 11786–11798, 2023.

Flores-Garza, D. A., Salazar-Aguilar, M. A., Ngueveu, S. U., e Laporte, G. The multi-vehicle cumulative covering tour problem. *Annals of Operations Research*, v. 258, n. 2, p. 761–780, 2017.

Fröhlich, G. E., Gansterer, M., e Doerner, K. F. Safe and secure vehicle routing: a survey on minimization of risk exposure. *International Transactions in Operational Research*, v. 30, n. 6, p. 3087–3121, 2023.

Gendreau, M., Laporte, G., e Semet, F. The covering tour problem. *Operations Research*, v. 45, n. 4, p. 568–576, 1997.

Glize, E., Roberti, R., Jozefowiez, N., e Ngueveu, S. U. Exact methods for mono-objective and bi-objective multi-vehicle covering tour problems. *European Journal of Operational Research*, v. 283, n. 3, p. 812–824, 2020.

Hà, M. H., Bostel, N., Langevin, A., e Rousseau, L. An exact algorithm and a metaheuristic for the multi-vehicle covering tour problem with a constraint on the number of vertices. *European Journal of Operational Research*, v. 226, n. 2, p. 211–220, 2013.

Hachicha, M., Hodgson, M. J., Laporte, G., e Semet, F. Heuristics for the multi-vehicle covering tour problem. *Computers & Operations Research*, v. 27, n. 1, p. 29–42, 2000.

Jiang, H., Lu, M., Tian, Y., Qiu, J., e Zhang, X. An evolutionary algorithm for solving capacitated vehicle routing problems by using local information. *Applied Soft Computing*, v. 117, p. 108431, 2022.

Jozefowiez, N., Semet, F., e Talbi, E. The bi-objective covering tour problem. *Computers & operations research*, v. 34, n. 7, p. 1929–1942, 2007.

Kammoun, M., Derbel, H., e Jarboui, B. A general variable neighborhood search with mixed VND for the multi-vehicle multi-covering tour problem. In: *International Conference on Variable Neighborhood Search*. Springer Nature Switzerland, 2019. p. 259–273.

Kammoun, M., Derbel, H., Ratli, M., e Jarboui, B. An integration of mixed VND and VNS: the case of the multivehicle covering tour problem. *International Transactions in Operational Research*, v. 24, n. 3, p. 663–679, 2017.

Karaoğlan, İ., Erdoğan, G., e Koç, Ç. The multi-vehicle probabilistic covering tour problem. *European Journal of Operational Research*, v. 271, n. 1, p. 278–287, 2018.

Labbé, M. e Laporte, G. Maximizing user convenience and postal service efficiency in post box location. *Belgian Journal of Operations Research, Statistics and Computer Science*, v. 26, p. 21–35, 1986.

Lehuédé, F., Péton, O., e Tricoire, F. A lexicographic minimax approach to the vehicle routing problem with route balancing. *European journal of operational research*, v. 282, n. 1, p. 129–147, 2020.

Lopes, R., Souza, V. A. A., e da Cunha, A. S. A branch-and-price algorithm for the multi-vehicle covering tour problem. *Electronic Notes in Discrete Mathematics*, v. 44, p. 61–66, 2013.

Mancini, S., Gansterer, M., e Hartl, R. F. The collaborative consistent vehicle routing problem with workload balance. *European journal of operational research*, v. 293, n. 3, p. 955–965, 2021.

Mara, S. T. W., Kuo, R., e Asih, A. M. S. Location-routing problem: a classification of recent research. *International Transactions in Operational Research*, v. 28, n. 6, p. 2941–2983, 2021.

Margolis, J. T., Song, Y., e Mason, S. J. A multi-vehicle covering tour problem with speed optimization. *Networks*, v. 79, n. 2, p. 119–142, 2022.

Matl, P., Hartl, R. F., e Vidal, T. Workload equity in vehicle routing: The impact of alternative workload resources. *Computers & Operations Research*, v. 110, p. 116–129, 2019.

Oliveira, W. A., Moretti, A. C., e Reis, E. F. Multi-vehicle covering tour problem: building routes for urban patrolling. *Brazilian Operations Research Society, Pesquisa Operacional*, v. 35, n. 3, p. 617–644, 2015.

Ombuki-Berman, B. e Hanshar, F. T. Using genetic algorithms for multi-depot vehicle routing. In: *Bio-inspired algorithms for the vehicle routing problem*p. 77–99. Berlin, Heidelberg: Springer, 2009.

Ota, C. T., Fiorotto, D. J., Ghidini, C. T. L. S., e Oliveira, W. A. A flow-based model for the multivehicle covering tour problem with route balancing. *International Transactions in Operational Research*, v. 31, n. 5, p. 2930–2955, 2024.

Ota, C. T., Oliveira, W. A., e Moretti, A. C. Um algoritmo exato e um algoritmo genético para o problema de rotas de cobertura multiveículo. In: *Anais do XLIX Simpósio Brasileiro de Pesquisa Operacional*. SOBRAPO, 2017. p. 1–12.

Pham, T. A., Hà, M. H., e Nguyen, X. H. Solving the multi-vehicle multi-covering tour problem. *Computers & operations research*, v. 88, p. 258–278, 2017.

Reyes, A. e Ribeiro, C. C. Extending time-to-target plots to multiple instances. *International Transactions in Operational Research*, v. 25, n. 5, p. 1515–1536, 2018.

Tan, H. T. e Hill, R. R. The in-transit vigilant covering tour problem for routing unmanned ground vehicles. In: *Operations Research for Unmanned Systems,*p. 7–26. : Wiley Online Library, 2016.

Torabi, P., Hemmati, A., Oleynik, A., e Alendal, G. A deep reinforcement learning hyperheuristic for the covering tour problem with varying coverage. *Computers & Operations Research*, v. 174, p. 106881, 2025.

Torabi, P., Oleynik, A., Hemmati, A., e Alendal, G. Covering tour problem with varying coverage: Application to marine environmental monitoring. *Applied Mathematical Modelling*, v. 124, p. 279–299, 2023.

Tricoire, F., Graf, A., e Gutjahr, W. J. The bi-objective stochastic covering tour problem. *Computers & Operations Research*, v. 39, n. 7, p. 1582–1592, 2012.