

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Algoritmos Evolutivos Aplicados ao Problema de Roteamento e Produção

Luiz Fernando Rodrigues

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Luiz Fernando Rodrigues

Algoritmos Evolutivos Aplicados ao Problema de Roteamento e Produção

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. EXEMPLAR DE DEFESA

Área de concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Maristela Oliveira dos Santos

São Carlos
Setembro de 2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

R696a Rodrigues, Luiz Fernando
Algoritmos Evolutivos Aplicados ao Problema de
Roteamento e Produção / Luiz Fernando Rodrigues;
orientadora Maristela Oliveira dos Santos. -- São
Carlos, 2022.
95 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2022.

1. Problema de Roteamento e Produção. 2.
Algoritmos Evolutivos. 3. Otimização Inteira Mista.
I. dos Santos, Maristela Oliveira , orient. II.
Título.

Luiz Fernando Rodrigues

**Evolutionary Algorithms Applied to the Production
Routing Problem**

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, in partial fulfillment of the requirements for the degree of the Doctor in Science - Program in Computer Science and Computational Mathematics. EXAMINATION BOARD PRESENTATION COPY

Concentration area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Maristela Oliveira dos Santos

São Carlos
Setembro de 2022

À Belinha e Sansão.

AGRADECIMENTOS

Aos meus orientadores, Maristela e Bernardo, agradeço imensamente pela paciência, dedicação e valorosas contribuições que tornaram possível a realização deste trabalho. Em especial, à Maristela, por me acolher de forma tão especial como seu orientando e, principalmente, por ter acreditado neste projeto desde o início.

Aos professores que fizeram parte desta minha jornada formativa.

Aos servidores do ICMC pela prestimosidade e eficiência.

À Vânia pelo amor que me impulsiona todos os dias.

Ao Marcelo pela amizade e pelo incentivo de todas as horas.

À minha família pelo apoio incondicional.

Aos amigos que encontrei ao longo desta caminhada.

“In the preservation of favoured individuals and races, during the constantly-recurrent Struggle for Existence, we see the most powerful and ever-acting means of selection. The struggle for existence inevitably follows from the high geometrical ratio of increase which is common to all organic beings.”

Charles Darwin - On the Origin of Species (1859)

RESUMO

RODRIGUES, L.F. **Algoritmos Evolutivos Aplicados ao Problema de Roteamento e Produção**. Setembro de 2022. 96p. Tese (Doutorado em Ciências) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Setembro de 2022.

Neste trabalho, abordamos o Problema de Roteamento e Produção (PRP) que consiste em determinar de maneira integrada o planejamento da produção, dos estoques e o roteamento de veículos de modo a minimizar os custos envolvidos. Uma fábrica é responsável por produzir diversos tipos de produtos, visando atender a demanda conhecida de um conjunto de clientes ao longo do horizonte finito de planejamento. A distribuição dos produtos aos clientes é realizada por uma frota homogênea e limitada de veículos. Para o PRP, as abordagens evolutivas não foram exploradas em profundidade, especialmente para o problema com múltiplos produtos. Desse modo, este trabalho mitiga esta lacuna da literatura, desenvolvendo Algoritmos Genéticos e Meméticos e testando sua eficácia em conjuntos de instâncias geradas aleatoriamente, comparando os resultados obtidos com um solver comercial de otimização. Além disso, avaliamos o desempenho dos métodos propostos em instâncias clássicas da literatura para o PRP com um único item. Os experimentos computacionais foram executados com o objetivo de avaliar o impacto das inúmeras combinações de parâmetros envolvendo a meta-heurística e, a partir de análises estatísticas, evidenciamos a robustez das técnicas propostas. Os experimentos computacionais mostraram que o método proposto superou o solver comercial, determinando factíveis de alta qualidade, principalmente nas instâncias maiores do PRP com múltiplos produtos.

Palavras-chave: Problema de Roteamento e Produção. Algoritmos Evolutivos. Otimização Inteira Mista.

ABSTRACT

RODRIGUES, L.F. **Evolutionary Algorithms Applied to the Production Routing Problem.** Setembro de 2022. 96p. Thesis (Doctorate in Science) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Setembro de 2022.

In this thesis we address the Production Routing Problem (PRP), which consists of determining, in an integrated way, production and inventory planning and vehicle routing in order to minimize the costs involved. In the problem, a plant is responsible for producing several types of products in order to meet the known demand of a set of customers over the planning horizon, using a homogeneous fleet of vehicles. In the literature, evolutionary approaches have not been explored in depth for the PRP, specifically for the problem with multiple products. Thus, this work mitigates this gap, presenting the novel Memetic and Genetic Algorithms and testing its effectiveness on randomly generated sets of instances, comparing the results obtained with a commercial optimization solver. In our solution approach, several classic operators from the literature were implemented; Furthermore, we propose four novel genetic operators. In addition, we evaluated the proposed method's performance in classical instances of literature to the PRP with a single item. The computational experiments were carried out with the objective of evaluating the impact of the numerous parameter combinations involving the metaheuristic and, from statistical analyses, we evidence proposed technique's robustness. Computational experiments showed that our proposed method outperforms the state-of-the-art solver in determining feasibly high-quality solutions, mainly on large instances for the PRP with multiple items.

Keywords: Production Routing Problem. Evolutionary Algorithms. Mixed Integer Optimization.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Fluxograma típico de um Algoritmo Genético. | 44 |
| Figura 2 – Fluxograma típico de um Algoritmo Memético. | 44 |
| Figura 3 – Estrutura (classe) de um individuo. | 45 |
| Figura 4 – Estrutura populacional em árvore. | 46 |
| Figura 5 – <i>Gaps</i> médios obtidos pelas estratégias utilizadas para geração da população inicial (critério de parada: 300 segundos). | 64 |
| Figura 6 – <i>Gaps</i> médios em porcentagens para diferentes combinações dos mecanismos de seleção (critério de parada: 300 segundos). | 65 |
| Figura 7 – <i>Gaps</i> médios em porcentagens para diferentes combinações dos mecanismos de seleção (critério de parada: 300 segundos). | 66 |
| Figura 8 – <i>Gaps</i> médios dos conjuntos em porcentagens para diferentes combinações dos operadores genéticos (critério de parada: 300 segundos). | 67 |
| Figura 9 – <i>Gaps</i> médios em porcentagens para diferentes combinações dos operadores genéticos (critério de parada: 300 segundos). | 68 |
| Figura 10 – <i>Gaps</i> médios em porcentagens para diferentes taxas de mutação (critério de parada: 300 segundos). | 70 |
| Figura 11 – <i>Gaps</i> médios em porcentagens para diferentes estratégias de aplicação das buscas locais (critério de parada: 300 segundos). | 71 |
| Figura 12 – <i>Gaps</i> médios em porcentagens com e sem reinícios (critério de parada: 300 segundos). | 73 |
| Figura 13 – Evolução da taxa de diversidade discreta em uma população ao longo das gerações (critério de parada: 300 gerações). | 74 |
| Figura 14 – Evolução da incumbente em instâncias com 30 clientes (critério de parada: 21600 segundos). | 75 |
| Figura 15 – Evolução da incumbente em instâncias com 20 clientes (critério de parada: 21600 segundos). | 76 |
| Figura 16 – MA versus GA: <i>Gaps</i> médios em porcentagens para as instâncias com 5 clientes (critério de parada: 300 segundos). | 78 |
| Figura 17 – Rotas de individuo antes da aplicação da Busca Local. | 79 |
| Figura 18 – Rotas de individuo depois da aplicação da Busca Local. | 81 |
| Figura 19 – Melhoria obtida pela estratégia híbrida (<i>gaps</i> médios em porcentagens). | 84 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Classificação dos trabalhos envolvendo o PRP com múltiplos produtos e uma fábrica. | 32 |
| Tabela 2 – Parâmetros para geração das instâncias com múltiplos produtos. | 62 |
| Tabela 3 – Visão geral das instâncias com múltiplos produtos. | 63 |
| Tabela 4 – Descrição das classes de instâncias com múltiplos produtos. | 63 |
| Tabela 5 – Taxas de probabilidades testadas para aplicação dos critérios de seleção. | 65 |
| Tabela 6 – <i>Gaps</i> médios em porcentagens para diferentes combinações dos mecanismos de seleção (critério de parada: 300 segundos). | 66 |
| Tabela 7 – <i>Gaps</i> médios em porcentagens para diferentes combinações dos operadores genéticos (critério de parada: 300 segundos). | 68 |
| Tabela 8 – <i>Gaps</i> médios em porcentagens para diferentes estratégias de aplicação das buscas locais (critério de parada: 300 segundos). | 69 |
| Tabela 9 – <i>Gaps</i> médios em porcentagens para diferentes níveis de intensificação da busca local (critério de parada: 300 segundos). | 72 |
| Tabela 10 – <i>Gaps</i> médios em porcentagens com e sem reinícios (critério de parada: 300 segundos). | 72 |
| Tabela 11 – <i>Gaps</i> médios em porcentagens de acordo com a estrutura populacional e mecanismos de seleção adotados (critério de parada: 300 segundos). | 73 |
| Tabela 12 – <i>Gaps</i> médios em porcentagens usando a estratégia de decomposição (critério de parada: 300 segundos). | 76 |
| Tabela 13 – Efeito da Busca Local: <i>Gaps</i> médios em porcentagens para as instâncias com 5 clientes (critério de parada: 300 segundos). | 77 |
| Tabela 14 – <i>Gaps</i> médios em porcentagens para as instâncias com múltiplos produtos. | 80 |
| Tabela 15 – Tempos de execução médios (em segundos) para as instâncias com múltiplos produtos. | 82 |
| Tabela 16 – Número de soluções ótimas obtidas nas instâncias com múltiplos produtos. | 82 |
| Tabela 17 – <i>Gaps</i> médios em porcentagens das estratégias híbridas. | 83 |
| Tabela 18 – Visão geral das instâncias de Archetti <i>et al.</i> (2011). | 85 |
| Tabela 19 – Descrição das classes de Archetti <i>et al.</i> (2011). | 85 |
| Tabela 20 – Estado-da-arte das técnicas para o PRP clássico. | 85 |
| Tabela 21 – <i>Gaps</i> médios em porcentagens das melhores soluções conhecidas para o conjunto A. | 87 |
| Tabela 22 – Tempos médios de execução em segundos para o conjunto A. Nota: * = tempo estimado (Vadseth, Andersson e Stålhane (2022)). | 88 |

SUMÁRIO

| | | |
|------------|--|-----------|
| 1 | INTRODUÇÃO | 23 |
| 2 | REVISÃO BIBLIOGRÁFICA | 27 |
| 2.1 | Problema de Roteamento e Produção com apenas um produto | 27 |
| 2.2 | Problema de Roteamento e Produção com múltiplos produtos | 29 |
| 2.3 | Algoritmos Evolutivos aplicados ao PRP | 31 |
| 3 | MODELOS MATEMÁTICOS | 35 |
| 3.1 | Modelo para o PRP com múltiplos produtos e frota homogênea | 35 |
| 3.2 | Modelo para o PRP com apenas um produto | 38 |
| 3.3 | Modelo para o PRP com múltiplos produtos e frota heterogênea | 41 |
| 3.4 | Procedimento de decomposição | 41 |
| 4 | ALGORITMOS EVOLUTIVOS | 43 |
| 4.1 | Algoritmos Evolutivos com População Estruturada | 43 |
| 4.1.1 | Representação da solução | 43 |
| 4.1.2 | Função de avaliação | 45 |
| 4.1.3 | Estrutura populacional | 46 |
| 4.1.4 | População inicial | 47 |
| 4.1.4.1 | Solução inicial: Problema de roteamento | 47 |
| 4.1.4.2 | Solução inicial: Problema de produção | 48 |
| 4.1.5 | Mecanismos de seleção e substituição | 49 |
| 4.1.6 | Operadores genéticos | 50 |
| 4.1.7 | Algoritmos de factibilização | 52 |
| 4.1.8 | Busca local | 55 |
| 4.1.9 | Estratégias de diversificação e reinícios | 56 |
| 4.1.10 | Estrutura básica | 57 |
| 4.2 | Algoritmos Evolutivos Clássicos | 57 |
| 5 | EXPERIMENTOS COMPUTACIONAIS | 61 |
| 5.1 | Geração de instâncias | 61 |
| 5.2 | Ajuste dos melhores parâmetros e estratégias | 62 |
| 5.2.1 | População inicial | 63 |
| 5.2.2 | Mecanismos de seleção | 64 |
| 5.2.3 | Operadores genéticos | 66 |
| 5.2.4 | Busca local | 68 |
| 5.2.5 | Estratégias de diversificação e reinícios | 70 |

| | | |
|------------|---|-----------|
| 5.2.6 | Critério de parada | 71 |
| 5.2.7 | Estrutura populacional e mecanismos de seleção | 73 |
| 5.3 | Estratégia de decomposição | 74 |
| 5.4 | Comparação entre os algoritmos genético e memético | 76 |
| 5.5 | Resultados para as instâncias com múltiplos produtos | 78 |
| 5.5.1 | Estratégia híbrida usando o solver | 81 |
| 5.6 | Descrição das instâncias com apenas um produto | 83 |
| 5.7 | Resultados para as instâncias com apenas um produto | 84 |
| 6 | CONCLUSÕES E PERSPECTIVAS FUTURAS | 89 |
| | REFERÊNCIAS | 93 |

1 INTRODUÇÃO

Os problemas envolvendo o planejamento da produção, do estoque e da distribuição têm sido amplamente estudados nas últimas décadas, embora as publicações que consideram estas decisões integradas em seus diferentes níveis ainda sejam recentes na literatura (Absi *et al.* (2018), Moons *et al.* (2017), Adulyasak, Cordeau e Jans (2015)). O problema de roteamento e produção (PRP) aborda dois importantes aspectos do planejamento da cadeia de suprimentos. O primeiro é o problema de dimensionamento de lotes (LSP, *lot-sizing problem*) que envolve os aspectos inerentes ao planejamento da produção e dos estoques, ou seja, determinar quanto e quando produzir de modo a atender a demanda, minimizando os custos de estoques, preparação e produção. O segundo é o problema de roteamento dos veículos (VRP, *vehicle routing problem*) que, por sua vez, determina como as frotas de veículos serão utilizadas para entregar os produtos, atendendo a demanda dos clientes e minimizando os custos de distribuição. Tradicionalmente, estes problemas eram otimizados sequencialmente, ou seja, o planejamento da produção e dos estoques servia como parâmetro de entrada para a definição das rotas de entregas ou vice-versa. Embora a complexidade do problema integrado seja maior, essa estratégia proporciona importante e significativa redução dos custos da cadeia produtiva segundo Adulyasak, Cordeau e Jans (2015). A esse respeito, Chandra e Fisher (1994) mostraram que uma abordagem integrada pode gerar reduções nos custos de até 20% em comparação com a abordagem sequencial.

No PRP devemos decidir, em uma fábrica, a produção de múltiplos itens ao longo de um horizonte finito de planejamento discretizado em períodos de tempo. A capacidade de produção é limitada e a distribuição dos produtos aos clientes é realizada utilizando uma frota homogênea e limitada de veículos. Os produtos podem ser armazenados na fábrica e nos clientes com diferentes custos de estoques. Em resumo, o objetivo principal do problema é minimizar os custos de produção, de estoques na fábrica e nos clientes e os custos de roteamento dos veículos, respeitando-se as restrições de capacidade de modo a atender a demanda dos clientes. Neste trabalho, adotamos a nomenclatura mais conhecida para o PRP, mas este também é conhecido por PIDRP (*production, inventory, distribution and routing problem*) ou ainda por IPDP (*integrated production and distribution problem*), como destacado por Miranda *et al.* (2018). Algumas revisões sobre esse problema podem ser encontradas em Adulyasak, Cordeau e Jans (2015), Moons *et al.* (2017), Absi *et al.* (2018) e Darvish *et al.* (2021).

No PRP estudado, consideramos que a fábrica ou fornecedor será responsável pelo gerenciamento dos estoques nos clientes, política de integração denominada de VMI (*vendor managed inventory*). Portanto, a fábrica decidirá quando e como serão abastecidos os estoques dos clientes, o que garante o atendimento da demanda ao longo do horizonte

de planejamento. As vantagens dessa política integrada para o gerenciamento dos estoques no contexto do problema de estoques e roteamento (IRP, *inventory routing problem*) são abordadas no trabalho de Archetti e Speranza (2016). Ainda em relação aos estoques, adotamos a política de abastecimento de nível máximo (ML, *maximum level*), permitindo, dessa forma, que as quantidades entregues em cada cliente tenham qualquer valor positivo, desde que o nível máximo de estoque não seja excedido. Archetti *et al.* (2011) realizam um estudo comparativo entre a política ML e a política de pedidos até o nível máximo (OU, *order-up to level*), analisando os custos e as características das soluções em diferentes contextos. A política OU estabelece que as quantidades entregues a cada cliente elevem o nível do estoque sempre para o nível máximo permitido.

Neste trabalho, investigamos algoritmos evolutivos para lidar com o problema de roteamento e produção com múltiplos produtos, tendo como principal contribuição a utilização destas técnicas populacionais ainda pouco exploradas no contexto desse problema. Embora existam alguns trabalhos utilizando estas técnicas para outras abordagens do problema como, por exemplo, o trabalho pioneiro de Boudia e Prins (2009) que propõem um algoritmo memético com gerenciamento da população para o PRP com apenas um item. Um algoritmo genético híbrido foi proposto por Izadi, Ahmadizar e Arkat (2020) e um algoritmo memético por Yagmur e Kesen (2021), ambos para o PRP considerando a programação da produção. Desse modo, o objetivo principal dessa pesquisa é o desenvolvimento de algoritmos evolutivos, mais especificamente algoritmos genéticos e meméticos, utilizando uma estrutura populacional hierárquica para organizar os indivíduos e melhorar o desempenho geral dos métodos. Além disso, analisamos o desempenho desta abordagem em relação aos algoritmos evolutivos tradicionais que não utilizam populações estruturadas.

Por se tratar de uma meta-heurística evolutiva populacional robusta, envolvendo o ajuste de muitos parâmetros além da necessidade de incorporação de estratégias desenvolvidas especificamente para o problema, sua aplicação neste contexto torna-se desafiadora. Neste sentido, para gerar um conjunto de soluções que irão formar a população inicial, dividimos o problema original em dois subproblemas, utilizando estratégias clássicas para revolver cada um deles separadamente, mas de modo que, ao final do processo, tenhamos uma solução para o problema integrado. Além disso, modificamos os algoritmos para viabilizar a geração do número necessário de soluções para compor a população inicial. Desenvolvemos ainda quatro novos operadores genéticos, de um ponto e dois pontos, para o problema integrado e adaptamos um outro operador clássico do TSP. Diversas estratégias foram adaptadas e utilizadas na busca local como parte do processo de intensificação das buscas por melhores soluções. Os métodos propostos foram avaliados em um conjunto de instâncias geradas aleatoriamente e os resultados foram comparados com um solver de otimização. É importante salientar que os algoritmos evolutivos propostos apresentaram desempenho muito superior ao solver nas instâncias com múltiplos itens geradas nesse trabalho. Por fim, comparamos o desempenho da técnica com o estado da arte envolvendo

um conjunto amplamente utilizado de instâncias do PRP com um único item. Embora todo o desenvolvimento tenha tido como foco o problema com múltiplos itens, as técnicas propostas apresentaram bom desempenho quando comparadas ao estado da arte envolvendo o problema com apenas um item.

Portanto, podemos elencar as principais contribuições desse trabalho da seguinte maneira:

- i) desenvolvemos algoritmos evolutivos para o PRP com múltiplos produtos e frota homogênea, mas que pode ser adaptado para frotas heterogêneas e, ainda ser usado para resolver o problema clássico com apenas um produto;
- ii) elaboramos quatro novos operadores genéticos (cruzamento e mutação), de um ponto e dois pontos, projetados especificamente para o PRP;
- iii) implementamos heurísticas de decomposição, a partir da adaptação de estratégias clássicas do LSP e VRP, com ótimo desempenho computacional para obter um conjunto numeroso e diverso de soluções factíveis para o problema integrado;
- iv) elaboramos uma heurística de melhoria que combina diversas estratégias clássicas da literatura capaz de explorar eficientemente o espaço de soluções e que, devido à simplicidade de implementação, pode ser usada em problemas reais envolvendo instâncias de grande porte;
- v) apresentamos um estudo comparativo entre os algoritmos genético e memético, além de análises acerca do impacto da estrutura populacional e dos mecanismos de seleção nos resultados alcançados pelas técnicas;
- vi) implementamos uma abordagem clássica de decomposição dos problemas de dimensionamento de lotes e de roteamento, utilizando os algoritmos evolutivos para resolver o problema de roteamento enquanto o solver atua no problema de produção;
- vii) geramos conjuntos de instâncias para execução de experimentos computacionais e que serão disponibilizadas para servir de base de dados visando futuras comparações e extensão dos métodos propostos.

Esse trabalho está organizado do seguinte modo: no Capítulo 2, apresentamos uma revisão da literatura com os mais recentes e relevantes trabalhos publicados. No Capítulo 3, definimos os modelos matemáticos com as características dos problemas abordados ao longo desse trabalho. No Capítulo 4, apresentamos os componentes dos algoritmos evolutivos propostos juntamente com alguns aspectos computacionais relacionados a sua implementação. No Capítulo 5, por sua vez, os resultados dos experimentos são reportados e analisados estatisticamente e, finalmente, no Capítulo 6 as conclusões do trabalho e algumas perspectivas para futuras pesquisas são apresentadas.

2 REVISÃO BIBLIOGRÁFICA

Iniciamos este capítulo com uma breve revisão acerca dos trabalhos envolvendo o PRP com apenas um item. Na sequência, elencamos os trabalhos que abordaram o problema com múltiplos produtos, foco principal deste trabalho. Finalmente, revisamos os artigos que utilizaram algoritmos evolutivos no contexto do PRP ou de suas extensões.

2.1 Problema de Roteamento e Produção com apenas um produto

Por se tratar de um problema NP-difícil, segundo Adulyasak, Cordeau e Jans (2015), Archetti *et al.* (2011), Boudia, Louly e Prins (2007), métodos exatos tem sua aplicação restrita a instâncias de pequeno e médio porte. Considerando o problema com apenas uma fábrica, um produto e utilizando a técnica *branch-and-cut* para resolução, temos o trabalho de Ruokokoski *et al.* (2010) que aborda o problema com apenas um veículo não capacitado e o de Qiu *et al.* (2018a) considerando uma frota homogênea de veículos capacitados e janelas de tempo. Os autores desenvolvem ainda uma heurística híbrida que utiliza várias famílias de cortes válidos para obter limitantes. Qiu, Qiao e Pardalos (2019) analisam o impacto das políticas de abastecimento de estoques em problemas reais, além de introduzir novas famílias de restrições para o PRP com produtos perecíveis.

Darvish, Archetti e Coelho (2019), por sua vez, propõem um modelo de otimização inteira mista para o PRP considerando a crescente preocupação com operações ecológicas e o impacto das decisões de curto prazo nas emissões de carbono, seus aspectos econômicos, ambientais e sociais no processo de otimização. Para isso, a técnica *branch-and-cut* e as instâncias adaptadas de Archetti *et al.* (2011) são utilizadas. Já em Ramos, Alves e Valério de Carvalho (2020) é desenvolvida uma formulação baseada em fluxo de redes para o problema, considerando múltiplas viagens com janelas de tempo e Zhang *et al.* (2021) desenvolvem uma técnica utilizando decomposição de Benders.

A maioria dos trabalhos para o problema com uma fábrica e um produto, utilizam estratégias híbridas de resolução combinando métodos exatos e heurísticos, além das instâncias propostas por Archetti *et al.* (2011) e Boudia, Louly e Prins (2007). Dentre eles, destacamos o trabalho de Adulyasak, Cordeau e Jans (2014) que propõe um método que utiliza uma heurística ANLS (*adaptive large neighborhood search*) para lidar com as variáveis binárias enquanto as variáveis contínuas são determinadas de maneira exata usando um modelo de fluxo em redes. Absi *et al.* (2015) apresentam uma heurística iterativa que resolve o problema de dimensionamento de lotes na primeira fase, considerando um custo fixo de distribuição. Na segunda fase, otimiza a rota de cada veículo como múltiplos problemas do caixeiro viajante e ajusta os custos de entregas da primeira fase, quando necessário.

Solyalı e Süral (2017) propõem uma heurística de decomposição para resolver o problema em cinco etapas, com a utilização de métodos exatos para resolver os subproblemas nestas fases. Russell (2017) apresenta duas técnicas baseadas em programação matemática para resolver o problema em etapas. Ao resolver o problema relaxado, determina um plano de produção com os níveis de estoques e a alocação das entregas aos veículos. Em seguida, utiliza uma heurística para otimizar as rotas dos veículos em cada período e, por fim, um procedimento iterativo determina a solução final. Qiu *et al.* (2018b) apresentam uma meta-heurística de busca baseada em VNS (*variable neighborhood search*) e VND (*variable neighborhood descent*) modificadas e combinadas com métodos exatos. A meta-heurística trata do problema de roteamento ao passo que as decisões de produção e estoques são resolvidas por um solver de otimização.

Avci e Yildiz (2019) apresentam dois modelos para o PRP considerando uma política de espaçamento entre as visitas, estabelecendo um intervalo mínimo e máximo para duas visitas consecutivas com o objetivo de melhorar a qualidade dos serviços oferecidos aos clientes. Os autores desenvolvem um algoritmo iterativo baseado em programação matemática para resolver o problema. Chitsaz, Cordeau e Jans (2019) propõem um modelo geral para o problema de roteamento e montagem (ARP, *assembly routing problem*) que consiste em planejar, simultaneamente, a montagem de um produto final na fábrica e o roteamento de veículos, coletando materiais nos fornecedores para garantir os níveis adequados de estoques de matéria-prima exigidos no processo de fabricação. Para isso, implementam uma técnica de decomposição combinada com métodos exatos que divide o problema em três fases. A primeira fase determina um cronograma de montagem dos produtos, enquanto a segunda otimiza as quantidades produzidas, as visitas aos fornecedores e as quantidades que serão entregues. Na terceira fase, o roteamento dos veículos para cada período é determinado. Além disso, essa técnica foi adaptada para resolver as instâncias clássicas do PRP.

Manousakis *et al.* (2022), por sua vez, apresentam uma nova formulação baseada em fluxo de redes (*two-commodity flow*) para o PRP e um algoritmo de duas fases que explora o espaço de soluções infactíveis. Na primeira fase, resolvem o problema relaxado para determinar os planos de produção e distribuição. Na segunda, as decisões de roteamento são otimizadas com a utilização de uma busca local que oscila entre o espaço de soluções factíveis e infactíveis. Um componente do método utiliza programação inteira para restaurar a viabilidade e diversificar a busca nesta fase. Vadseth, Andersson e Stålhane (2022) apresentam uma formulação de fluxo de caminho para o PRP e desenvolvem uma heurística de melhoria com múltiplos reinícios combinada com métodos exatos. O método determina um conjunto inicial de rotas resolvendo uma versão simplificada do problema original. Em seguida, utiliza outro problema inteiro misto para fazer alterações nas rotas iniciais enquanto resolve o problema original, permitindo, dessa forma, que o algoritmo consiga melhorar localmente uma determinada solução.

2.2 Problema de Roteamento e Produção com múltiplos produtos

Para o problema com múltiplos itens, Chandra e Fisher (1994) foram pioneiros ao introduzir o problema integrado de produção e roteamento, considerando a distribuição dos produtos para um conjunto de clientes utilizando uma frota de veículos homogêneos. Duas técnicas foram propostas, uma decompondo e resolvendo o problema de dimensionamento de lotes e o problema roteamento separadamente e outra resolvendo o problema de forma integrada. Com o intuito de comparar o desempenho das duas abordagens, com diferentes valores dos parâmetros do modelo utilizado, foram realizados experimentos computacionais.

Fumero e Vercellis (1999) formulam um modelo considerando uma fábrica e múltiplos produtos sendo distribuídos por uma frota limitada de veículos homogêneos. Propõem uma técnica que resolve o problema de dimensionamento de lotes capacitado e o de roteamento separadamente, utilizando relaxação lagrangiana para separar as variáveis de transporte e produção, mas preservando a perspectiva global de otimização. Essa técnica é comparada com os resultados obtidos por uma técnica de decomposição que primeiro resolve o problema de produção para, em seguida, determinar uma solução para o roteamento dos veículos.

Armentano, Shiguemoto e Løkketangen (2011) consideram a produção de múltiplos itens em uma única fábrica com restrições de capacidade e usando uma frota homogênea e ilimitada de veículos. Os autores apresentam dois métodos de resolução a partir da Busca Tabu e da Busca Tabu combinada com *path relinking*. Desta forma, novos conjuntos de instâncias com múltiplos produtos foram geradas aleatoriamente e, as instâncias de Boudia, Louly e Prins (2007) com um único produto foram utilizadas para avaliar os métodos propostos. As soluções encontradas nas instâncias de pequeno porte foram comparadas com as soluções exatas obtidas pelo solver. Para as instâncias maiores, além da comparação entre os dois métodos propostos também foram utilizados limitantes inferiores obtidos por meio de relaxação lagrangiana.

Brahimi e Aouam (2016) pesquisam o problema com uma fábrica produzindo múltiplos itens para atender um conjunto de clientes com capacidades de produção e de estoque limitadas. As demandas dos clientes podem ser atendidas com atraso, mas com a inserção de penalidades na função objetivo. Além disso, dois modelos integrados de programação inteiro misto foram formulados. Os autores utilizam uma heurística *relax-and-fix* para resolver uma versão modificada do problema de dimensionamento de lotes combinada com uma busca local para construir as rotas de entregas e, assim, determinar uma solução completa para o problema integrado.

Miranda *et al.* (2018) abordam o PRP no contexto de uma fábrica de móveis que considera a produção de diversos itens que são usados na montagem de produtos finais. Os autores incorporam ao modelo várias características realistas, como múltiplos produtos,

tempos de preparação dependentes da sequência, frota heterogênea de veículos, rotas que se estendem por mais de um período e múltiplas janelas de tempo. Além disso, uma heurística foi proposta para resolver o problema decompondo-o em duas etapas que são resolvidas iterativamente. A primeira etapa da heurística considera as decisões de planejamento da produção e designação das entregas aos clientes, usando uma aproximação para os custos de roteamento e os tempos de viagem. A segunda etapa resolve o problema de roteamento usando um algoritmo de busca local, utilizando a solução desta etapa para atualizar os custos de roteamento e os tempos de viagem aproximados na primeira etapa.

Miranda, Morabito e Ferreira (2018) apresentam um modelo para o PRP inspirado em pequenas empresas brasileiras de móveis com uma linha de produção fabricando itens que serão usados na montagem de produtos finais. Esses produtos serão entregues aos clientes por um veículo apenas e as rotas podem se estender por mais de um período, considerando múltiplas janelas de tempo e os prazos de entrega. Os autores propõem seis heurísticas *relax-and-fix*, que exploram diferentes critérios para particionar e fixar as variáveis. Implementam, ainda, duas heurísticas híbridas em que a solução inicial é construída e depois otimizada por um solver comercial de otimização.

Qiu *et al.* (2018c), por sua vez, formulam um modelo inteiro misto para o PRP com múltiplos produtos, múltiplos veículos e custos de inicialização. O custo de inicialização é cobrado quando uma máquina inicia a produção de um lote ou muda de um produto para outro. Além disso, os autores implementam um *branch-and-cut* usando três famílias de desigualdades válidas para ajustar mais fortemente o modelo proposto. Convém ressaltar que o método foi testado em um conjunto de instâncias geradas aleatoriamente e também envolvendo uma empresa de alimentos da China.

Neves-Moreira *et al.* (2019) propõem um modelo inspirado em um problema real que consiste em um único centro de processamento de carnes com várias linhas de produção. Os produtos perecíveis são distribuídos usando uma frota heterogênea de veículos e considerando janelas de tempo. Propõem uma técnica que combina métodos heurísticos e exatos em três fases. Na primeira fase, produtos similares e clientes mais próximos são agrupados para reduzir o tamanho do problema original. Na segunda fase, uma solução inicial do PRP é obtida decompondo vários problemas de roteamento e estoque (IRP) e um de dimensionamento de lotes. Na terceira fase, a solução inicial é melhorada por diferentes modelos e usando uma heurística *fix-and-optimize*. A técnica é testada tanto em instâncias adaptadas da literatura quanto em instâncias realistas envolvendo uma empresa europeia.

Mostafa e Eltawil (2019) apresentam estratégias para reduzir a complexidade do PRP, introduzindo desigualdades válidas para o problema com uma única fábrica, múltiplos produtos e uma frota heterogênea de veículos. Essas estratégias tem o intuito de melhorar a qualidade dos limitantes inferiores, o que reduz o tempo computacional sem um efeito

significativo na qualidade das soluções encontradas. Os autores concluíram que foi possível reduzir as diferenças percentuais de todas as instâncias testadas quando comparados aos limitantes obtidos por relaxação usando modelos de programação linear. As instâncias propostas por Archetti *et al.* (2011) foram usadas para validar as técnicas propostas e também usadas como base para gerar novas instâncias do PRP com múltiplos produtos.

Li *et al.* (2019) formulam um modelo para o problema com uma fábrica, produzindo múltiplos produtos que serão entregues por uma frota homogênea de veículos. A produção e a entrega de um produto podem ser terceirizados e, nesse caso, um custo unitário será computado. Os autores implementam uma heurística baseada em programação matemática de três níveis. No primeiro nível, uma técnica iterativa de duas fases é aplicada para obter uma solução inicial para problema. No segundo nível, caso a solução do primeiro nível seja inviável, um problema de produção e distribuição restrito com o cronograma de preparação da produção fixado é resolvido. Em seguida, são resolvidos problemas do caixeiro viajante para cada veículo utilizado, com o objetivo de melhorar a solução obtida para o roteamento. No terceiro nível, para melhorar a solução incumbente, um procedimento *fix-and-optimize* é aplicado em cada cliente com base em modelo de produção e distribuição modificado. Por fim, novamente são resolvidos os respectivos problemas do caixeiro viajante dos veículos.

Salehi Sarbijan e Behnamian (2020) estudam os efeitos da terceirização em um PRP com múltiplos produtos e considerando o impacto das emissões dos gases de efeito estufa. O objetivo do problema é minimizar os custos de produção, terceirização, manutenção e emissão de gases. Os autores investigam os efeitos da velocidade, carga e tipos de veículos sobre a emissão de poluentes. Além disso, propõem um novo modelo linear inteiro misto e um algoritmo PSO (*particle swarm optimization*) para resolver o problema. O desempenho da técnica proposta foi avaliada em um conjunto de instâncias geradas aleatoriamente e comparado com a heurística VNS proposta por Qiu *et al.* (2018b).

Na Tabela 1 temos um resumo dos trabalhos que consideram o PRP com múltiplos produtos e uma fábrica. É importante observar que todos os trabalhos propõem novas instâncias para o PRP com múltiplos produtos, mas somente 5 utilizam as instâncias clássicas do PRP com um único item para comparação. Dentre esses trabalhos, 4 utilizam o conjunto A de instâncias, 1 o conjunto B e apenas 1 trabalho utilizou os dois conjuntos. Além disso, 4 artigos apresentam instâncias envolvendo problemas reais e 7 trabalhos apresentam modelos considerando características adicionais, com destaque para janelas de tempo e terceirização. Por fim, convém ressaltar que, dentre as técnicas de resolução, apenas esse trabalho utilizou algoritmos evolutivos.

2.3 Algoritmos Evolutivos aplicados ao PRP

Face aos desafios em termos da complexidade computacional, uma alternativa que tem se mostrado promissora é a utilização de meta-heurísticas, especialmente para

Tabela 1 – Classificação dos trabalhos envolvendo o PRP com múltiplos produtos e uma fábrica.

| Trabalhos | Estoque | | Rotamento | | Método de Solução | | Instâncias | | Extensão |
|--|----------|-------|-----------|--------------------------|-------------------|-----------|------------|------------------------------------|----------|
| | Política | Frota | #Veículos | Técnica | Clássicas | Realistas | | | |
| Chandra e Fisher (1994) | ML | Hom. | Limitada | Decomposição | | | | | |
| Fumero e Vercellis (1999) | ML | Hom. | Limitada | Relaxação Lagrangiana | | | | | |
| Armentano, Shigenmoto e Lokketangen (2011) | ML | Hom. | Limitada | Busca Tabu | B | | | | |
| Brahimi e Aouam (2016) | ML | Hom. | Único | Relax-and-fix | | | | Estoque negativos | |
| Miranda et al. (2018) | ML | Het. | Limitada | Decomposição | | | Sim | Múltiplas janelas de tempo | |
| Miranda, Morabito e Ferreira (2018) | | Hom. | Único | Relax-and-fix | | | Sim | Múltiplas janelas de tempo | |
| Qiu et al. (2018c) | | Hom. | Limitada | Branch-and-cut | | | Sim | Custo de inicialização | |
| Neves-Moreira et al. (2019) | ML | Het. | Limitada | Fix-and-Optimize | A | | Sim | Janelas de tempo | |
| Mostafa e Eltawil (2019) | | Het. | Limitada | Iterativo/Branch-and-cut | A1 | | | | |
| Li et al. (2019) | ML | Hom. | Limitada | Fix-and-Optimize | A, B | | | Terceirização | |
| Salehi Sarbjian e Behnamian (2020) | | Hom. | Limitada | Particle Swarm | | | | Terceirização e emissão de carbono | |
| Esta tese | ML | Hom. | Limitada | Algoritmo Memético | A1, A2 | | | | |

| Hom. - Homogênea | Het. - Heterogênea | A - (ARCHETTI *et al.*, 2011) | B - (BOUDIA; LOULY; PRINS, 2007) |

Fonte: Autoria própria

problemas de grande porte com características mais realistas. Resumidamente, destacamos alguns trabalhos que utilizaram os algoritmos genéticos e meméticos no contexto do PRP ou de suas extensões.

Boudia e Prins (2009) consideram o problema com uma única fábrica que produz um item que será distribuído por uma frota homogênea de veículos, utilizando um algoritmo memético com gerenciamento da população de indivíduos. Diversas versões com diferentes estratégias de gerenciamento populacional são avaliadas e comparadas com uma heurística de duas fases e um procedimento de busca baseada em GRASP.

Izadi, Ahmadizar e Arkat (2020) abordam o problema integrado de programação da produção, planejamento dos estoques e roteamento dos veículos. Na programação da produção consideram máquinas paralelas com tempos de preparação e a possibilidade de terceirização. A distribuição, por sua vez, será feita em lotes por uma frota homogênea de veículos capacitados. Para resolver o problema proposto, os autores utilizam propriedades de dominância da solução ótima como parte de um processo de hibridização de um algoritmo genético.

Yagmur e Kesen (2021) estudam o problema integrado de programação da produção e roteamento dos veículos em que os pedidos são processados por várias máquinas em série com a mesma sequência conhecida de permutação (*flow shop*). Os pedidos são distribuídos aos clientes usando uma frota heterogênea de veículos que podem realizar múltiplas viagens. Os autores apresentam uma formulação inteira mista e propõem dois métodos de resolução, um algoritmo memético e um *simulated annealing*. O algoritmo memético é dividido em duas fases, sendo que na primeira, os autores utilizam uma adaptação do algoritmo da divisão de Prins, considerando apenas o atraso total. Na segunda fase, uma heurística determina a designação de veículos para geração das rotas. No *simulated annealing* as rotas são geradas por operadores de vizinhanças e a atribuição de veículos é feita pela mesma heurística usada no algoritmo memético.

Kayé *et al.* (2021) pesquisam o problema de roteamento e produção com depósito externo que é uma extensão do PRP clássico. Nesse problema, a fábrica não tem estoque e toda produção será transportada e armazenada no depósito que será responsável por atender a demanda dos clientes. Os autores desenvolvem um algoritmo memético modificado no qual a mutação é substituída por três buscas locais que são aplicadas no roteamento dos veículos para melhorar a qualidade dos indivíduos da população.

Apresentamos, a seguir, a descrição do problema e os modelos matemáticos utilizados.

3 MODELOS MATEMÁTICOS

Apresentamos neste capítulo a definição do problema e os modelos matemáticos adotados para o PRP em seus diferentes contextos e que foram utilizados ao longo desta pesquisa.

3.1 Modelo para o PRP com múltiplos produtos e frota homogênea

Esse modelo é uma adaptação dos trabalhos de Armentano, Shiguemoto e Løketangen (2011) e Fumero e Vercellis (1999). Nesse caso, consideramos uma fábrica com capacidade limitada de produção e armazenamento, produzindo um conjunto de produtos que serão distribuídos a um conjunto de clientes para atender suas respectivas demandas em um horizonte finito de planejamento. Os produtos poderão ser armazenados na fábrica ou nos clientes com gerenciamento dos estoques feito pelo fornecedor (VMI). Além disso, adotamos as mesmas hipóteses da política de estoques de nível máximo (ML, *maximum level*) consideradas no trabalho original de Fumero e Vercellis (1999), permitindo que as quantidades entregues em cada cliente tenham qualquer valor positivo, desde que o nível máximo de estoque não seja excedido. A distribuição dos produtos é realizada por uma frota de veículos homogênea com a capacidade limitada, sendo considerados os custos fixos e variáveis pela utilização de cada veículo. Um cliente poderá ser visitado somente por um veículo em cada período e o veículo poderá realizar somente uma rota por período, partindo e voltando à fábrica ao final do trajeto. O objetivo do problema é minimizar os custos de produção e preparação, custos de estoques na fábrica e nos clientes, além dos custos de roteamento dos veículos. A seguir, definimos a forma de representação, os parâmetros e as variáveis de decisão do modelo.

Considere um grafo completo $G = (W, E)$ onde $W = \{0, 1, \dots, N\}$ é um conjunto de nós representando a fábrica e os clientes e $E = \{(i, k) : i, k \in W, i \neq k\}$ é o conjunto de arcos indicando as rotas entre a fábrica e os clientes. A fábrica, indicada pelo nó $i = 0$, gerencia os estoques dos clientes $i = 1, \dots, N$ em um horizonte finito de planejamento, $t = 1, \dots, T$. As entregas dos produtos, $p = 1, \dots, P$, serão feitas por uma frota limitada de veículos homogêneos $v = 1, \dots, V$ pertencentes a fábrica. Apresentamos a seguir um resumo desses parâmetros e as variáveis de decisão:

Parâmetros:

B = Capacidade de produção;

b_p = Tempo necessário para produzir o item p ;

c_p = Custo de produção do item p ;

s_p = Custo de preparação do item p ;

- M = Número grande (por exemplo: $\sum_p \sum_i \sum_t d_{pit}$).
- U_{pi} = Capacidade máxima de estoque do item p no cliente i ;
- I_{pi0} = Estoque inicial do item p no cliente i ;
- h_{pi} = Custo de estoque do item p no cliente i .
- C = Capacidade dos veículos;
- f = Custo fixo de transporte;
- a_{ik} = Custo de transporte para percorrer do nó i ao nó k ;
- d_{pit} = Demanda do item p no cliente i no período t .

Variáveis de decisão:

- $y_{pt} = \begin{cases} 1, & \text{se o item } p \text{ é produzido no período } t; \\ 0, & \text{caso contrário.} \end{cases}$
- x_{pt} = Quantidade do item p produzida no período t .
- I_{pit} = Estoque do item p no cliente i no final do período t .
- $z_{vikt} = \begin{cases} 1, & \text{se o veículo } v \text{ percorre o arco } (i, k) \text{ no período } t; \\ 0, & \text{caso contrário.} \end{cases}$
- r_{pvikt} = Quantidade do item p transportado pelo veículo v no arco (i, k) no período t ;
- q_{pvit} = Quantidade do item p entregue pelo veículo v no cliente i no período t .

Assumimos que as quantidades produzidas no período t estarão prontas para serem entregues neste mesmo período. Desse modo, temos o seguinte modelo geral para o PRP com múltiplos produtos:

Minimize

$$\sum_{p=1}^P \sum_{t=1}^T (s_p y_{pt} + c_p x_{pt}) + \sum_{p=1}^P \sum_{i=0}^N \sum_{t=1}^T h_{pi} I_{pit} + \sum_{v=1}^V \sum_{k=1}^N \sum_{t=1}^T f z_{v0kt} + \sum_{v=1}^V \sum_{\substack{i,k=0 \\ i \neq k}}^N \sum_{t=1}^T a_{ik} z_{vikt} \quad (3.1)$$

Sujeito a:

$$x_{pt} + I_{p0,t-1} - \sum_{v=1}^V \sum_{i=1}^N q_{pvit} = I_{p0t} \quad 1 \leq p \leq P \quad 1 \leq t \leq T \quad (3.2)$$

$$\sum_{v=1}^V q_{pvit} + I_{pi,t-1} - d_{pit} = I_{pit} \quad 1 \leq p \leq P \quad 1 \leq i \leq N \quad 1 \leq t \leq T \quad (3.3)$$

$$\sum_{p=1}^P b_p x_{pt} \leq B \quad 1 \leq t \leq T \quad (3.4)$$

$$x_{pt} \leq My_{pt} \quad 1 \leq p \leq P \quad 1 \leq t \leq T \quad (3.5)$$

$$I_{pit} \leq U_{pi} \quad 1 \leq p \leq P \quad 0 \leq i \leq N \quad 1 \leq t \leq T \quad (3.6)$$

$$\sum_{\substack{i=0 \\ i \neq k}}^N r_{pvikt} - \sum_{\substack{l=0 \\ l \neq k}}^N r_{pvklt} = q_{pvkt} \quad 1 \leq p \leq P \quad 1 \leq v \leq V \quad 1 \leq k \leq N \quad 1 \leq t \leq T \quad (3.7)$$

$$\sum_{v=1}^V \sum_{k=1}^N r_{pv0kt} - \sum_{v=1}^V \sum_{i=1}^N r_{pvi0t} = \sum_{v=1}^V \sum_{l=1}^N q_{pvlt} \quad 1 \leq p \leq P \quad 1 \leq t \leq T \quad (3.8)$$

$$\sum_{p=1}^P r_{pvikt} \leq Cz_{vikt} \quad 1 \leq v \leq V \quad 0 \leq i \leq N \quad 0 \leq k \leq N \quad 1 \leq t \leq T \quad i \neq k \quad (3.9)$$

$$\sum_{k=1}^N z_{v0kt} \leq 1 \quad 1 \leq v \leq V \quad 1 \leq t \leq T \quad (3.10)$$

$$\sum_{\substack{i=0 \\ i \neq k}}^N z_{vikt} - \sum_{\substack{l=0 \\ l \neq k}}^N z_{vklt} = 0 \quad 1 \leq v \leq V \quad 0 \leq k \leq N \quad 1 \leq t \leq T \quad (3.11)$$

$$\sum_{v=1}^V \sum_{\substack{i=0 \\ i \neq k}}^N z_{vikt} \leq 1 \quad 1 \leq k \leq N \quad 1 \leq t \leq T \quad (3.12)$$

$$y_{pt}, z_{vikt} \in \{0, 1\} \quad 1 \leq p \leq P \quad 1 \leq v \leq V \quad 0 \leq i \leq N \quad 0 \leq k \leq N \quad 1 \leq t \leq T \quad i \neq k \quad (3.13)$$

$$x_{pt}, I_{pit}, r_{pvikt}, q_{pvit} \geq 0 \quad 1 \leq p \leq P \quad 1 \leq v \leq V \quad 0 \leq i \leq N \quad 0 \leq k \leq N \quad 1 \leq t \leq T \quad i \neq k \quad (3.14)$$

A função objetivo (3.1) minimiza os custos de estoque, produção e roteamento, sendo que o primeiro termo representa os custos de preparação e produção dos itens. No segundo termo, temos os custos de estoques dos produtos na fábrica e nos clientes. Por fim, os custos fixos pela utilização de cada veículo e os custos variáveis para percorrer as rotas durante a distribuição dos itens aos clientes são computados. As restrições (3.2)

estabelecem o balanceamento de estoque na fábrica de acordo com a produção e as quantidades enviadas aos clientes. Nas restrições (3.3) ocorrem o balanceamento dos estoques dos clientes, considerando as entregas recebidas e o atendimento das demandas. Fumero e Vercellis (1999) destacam que se os produtos não fossem armazenados nos clientes então nas restrições (3.3) teríamos que $\sum q = d$. Logo, as demandas seriam usadas nas restrições (3.2) e, conseqüentemente, os problemas de produção e distribuição poderiam ser tratados separadamente. As restrições (3.4) definem a capacidade máxima de produção da fábrica em cada período. As restrições (3.5) por sua vez estabelecem a relação entre as variáveis de produção e preparação, ou seja, se não houver preparação então não poderá ocorrer produção do item neste período. As restrições (3.6) delimitam a capacidade máxima de estoque dos itens na fábrica e nos clientes. As restrições (3.7) e (3.8) são as equações de conservação do fluxo de itens, assegurando o balanceamento em cada cliente e a coleta na fábrica. Ainda de acordo com Fumero e Vercellis (1999), as restrições (3.7) foram definidas separadamente para cada veículo para impedir o transbordo em nós intermediários, ou seja, essas restrições, juntamente com a não-negatividade da variável q_{pvit} , impedem que um veículo realize o transporte de itens entre os clientes. Além disso, as restrições (3.7) e (3.8) também garantem a eliminação de sub-rotas. A capacidade máxima de carga dos veículos é delimitada pelas restrições (3.9). Fumero e Vercellis (1999) ressaltam que bastaria verificar a capacidade dos veículos na fábrica ($i = 0$), pois os veículos não recolhem produtos nos clientes. No entanto, as restrições são mantidas para todos os locais com o objetivo de assegurar a relação entre as variáveis binárias (z_{vikt}) e as variáveis de transporte (r_{pvikt}). As restrições (3.10) impõem no máximo uma rota para cada veículo por período e as restrições (3.11) asseguram que as rotas podem terminar somente na fábrica, garantindo assim o retorno dos veículos ao local de partida no final de cada período. As restrições (3.12) garantem que no máximo um veículo pode visitar um cliente em cada período, evitando assim que ocorram entregas fracionadas. Por fim, as restrições (3.13) e (3.14) definem os tipos das variáveis.

No sentido de obter um modelo mais geral, optamos por não adotar um nível mínimo de segurança para os estoques do mesmo modo que o trabalho de Fumero e Vercellis (1999). Além disso, não estabelecemos o tamanho máximo de uma rota ao assumirmos que todos os veículos da frota são capazes de percorrer qualquer uma das possíveis rotas, mas limitados naturalmente pelas capacidades de carga dos veículos.

3.2 Modelo para o PRP com apenas um produto

No sentido de viabilizar a comparação com o estado da arte envolvendo o PRP clássico com apenas um produto, adaptamos o modelo inicial retirando os índices referentes aos produtos. Com isso, detalhamos a seguir os parâmetros e as variáveis desse modelo modificado:

Parâmetros:

B_t = Capacidade (tempo) de produção da fábrica no período t ;

b = Tempo de processamento do produto;

c_t = Custo de produção do produto no período t ;

s_t = Custo de preparação do item p no período t ;

U_i = Capacidade máxima de estoque no local i ;

I_{i0} = Estoque do item no local i no período 0;

h_{it} = Custo de estoque no cliente i no período t ;

C = Capacidade dos veículos;

f = Custo fixo para utilização do veículo;

a_{ik} = Custo de transporte para percorrer a aresta (i, k) ;

d_{it} = Demanda do item no cliente i no período t .

Variáveis de decisão:

$$y_t = \begin{cases} 1, & \text{se houver produção no período } t; \\ 0, & \text{caso contrário.} \end{cases}$$

x_t = Quantidade produzida no período t ;

I_{it} = Estoque no local i ao final do período t ;

$$z_{vikt} = \begin{cases} 1, & \text{se o veículo } v \text{ percorre a aresta } (i, k) \text{ no período } t; \\ 0, & \text{caso contrário.} \end{cases}$$

r_{vikt} = Quantidade transportada pelo veículo v na aresta (i, k) no período t ;

q_{vit} = Quantidade entregue pelo veículo v no cliente i no período t .

E assim, temos o seguinte modelo para o PRP com apenas um produto:

Minimize

$$\sum_{i=0}^N \sum_{t=1}^T h_{it} I_{it} + \sum_{t=1}^T (s_t y_t + c_t x_t) + \sum_{v=1}^V \sum_{k=1}^N \sum_{t=1}^T f z_{v0kt} + \sum_{v=1}^V \sum_{\substack{i,k=0 \\ i \neq k}}^N \sum_{t=1}^T a_{ik} z_{vikt} \quad (3.15)$$

Sujeito a:

$$x_t + I_{0,t-1} - \sum_{v=1}^V \sum_{i=1}^N q_{vit} = I_{0t} \quad 1 \leq t \leq T \quad (3.16)$$

$$\sum_{v=1}^V q_{vit} + I_{i,t-1} - d_{it} = I_{it} \quad 1 \leq i \leq N \quad 1 \leq t \leq T \quad (3.17)$$

$$bx_t \leq B_t y_t \quad 1 \leq t \leq T \quad (3.18)$$

$$I_{it} \leq U_i \quad 0 \leq i \leq N \quad 1 \leq t \leq T \quad (3.19)$$

$$\sum_{\substack{i=0 \\ i \neq k}}^N r_{vikt} - \sum_{\substack{l=0 \\ l \neq k}}^N r_{vkl t} = q_{vkt} \quad 1 \leq v \leq V \quad 1 \leq k \leq N \quad 1 \leq t \leq T \quad (3.20)$$

$$\sum_{v=1}^V \sum_{k=1}^N r_{v0kt} - \sum_{v=1}^V \sum_{i=1}^N r_{vi0t} = \sum_{v=1}^V \sum_{l=1}^N q_{vlt} \quad 1 \leq t \leq T \quad (3.21)$$

$$r_{vikt} \leq C z_{vikt} \quad 1 \leq v \leq V \quad 0 \leq i \leq N \quad 0 \leq k \leq N \quad 1 \leq t \leq T \quad i \neq k \quad (3.22)$$

$$\sum_{k=1}^N z_{v0kt} \leq 1 \quad 1 \leq v \leq V \quad 1 \leq t \leq T \quad (3.23)$$

$$\sum_{\substack{i=0 \\ i \neq k}}^N z_{vikt} - \sum_{\substack{l=0 \\ l \neq k}}^N z_{vkl t} = 0 \quad 1 \leq v \leq V \quad 0 \leq k \leq N \quad 1 \leq t \leq T \quad (3.24)$$

$$\sum_{v=1}^V \sum_{\substack{i=0 \\ i \neq k}}^N z_{vikt} \leq 1 \quad 1 \leq k \leq N \quad 1 \leq t \leq T \quad (3.25)$$

$$y_t, z_{vikt} \in \{0, 1\} \quad 1 \leq v \leq V \quad 0 \leq i \leq N \quad 0 \leq k \leq N \quad 1 \leq t \leq T \quad i \neq k \quad (3.26)$$

$$x_t, I_{it}, r_{vikt}, q_{vit} \geq 0 \quad 1 \leq v \leq V \quad 0 \leq i \leq N \quad 0 \leq k \leq N \quad 1 \leq t \leq T \quad i \neq k \quad (3.27)$$

A função objetivo (3.15) minimiza os custos de produção, estoque e roteamento. As restrições (3.16) estabelecem o balanceamento de estoque na fábrica e as restrições (3.17), o balanceamento dos estoques nos clientes. As restrições (3.18) estabelecem a capacidade máxima de produção da fábrica em cada período e a relação entre as variáveis de produção e preparação. As restrições (3.19) delimitam a capacidade máxima de estoque na fábrica e nos clientes. As restrições (3.20) e (3.21) são as equações de conservação do fluxo de itens, assegurando o balanceamento em cada cliente, a coleta na fábrica e a eliminação de sub-rotas. As restrições (3.22) delimitam a capacidade máxima de carga dos veículos. As restrições (3.23) impõem no máximo uma rota para cada veículo por período. As restrições (3.24) asseguram que as rotas podem terminar somente na fábrica. As restrições (3.25) garantem que no máximo um veículo pode visitar um cliente em cada período. Por fim, as restrições (3.26) e (3.27) definem os tipos das variáveis.

3.3 Modelo para o PRP com múltiplos produtos e frota heterogênea

Por fim e considerando que as entregas dos produtos serão realizadas por uma frota heterogênea de veículos, ajustamos os valores dos parâmetros relacionados aos veículos e, assim, temos os seguintes parâmetros alterados em relação ao modelo inicial:

C_v = Capacidade do veículo v ;

f_v = Custo fixo para utilização do veículo v ;

a_{vik} = Custo de transporte do veículo v para percorrer o arco (i, k) ;

Portanto, temos o seguinte modelo geral para o PRP com múltiplos produtos e frota heterogênea:

Minimize

$$\sum_{p=1}^P \sum_{i=0}^N \sum_{t=1}^T h_{pit} I_{pit} + \sum_{p=1}^P \sum_{t=1}^T (s_{pt} y_{pt} + c_{pt} x_{pt}) + \sum_{v=1}^V \sum_{k=1}^N \sum_{t=1}^T f_v z_{v0kt} + \sum_{v=1}^V \sum_{\substack{i,k=0 \\ i \neq k}}^N \sum_{t=1}^T a_{vik} z_{vikt} \quad (3.28)$$

Sujeito a:

$$(3.2), (3.3), (3.4), (3.5), (3.6), (3.7), (3.8)$$

$$\sum_{p=1}^P r_{pvikt} \leq C_v z_{vikt} \quad 1 \leq v \leq V \quad 0 \leq i \leq N \quad 0 \leq k \leq N \quad 1 \leq t \leq T \quad i \neq k \quad (3.29)$$

$$(3.10), (3.11), (3.12), (3.13), (3.14)$$

Note que a função objetivo (3.28) foi modificada para incluir os custos fixos e os custos de transporte que dependem, nesse novo modelo, do tipo de veículo utilizado. Além disso, as restrições (3.9) delimitam a capacidade máxima de carga para cada tipo de veículo. Por outro lado, as demais restrições permanecem inalteradas.

3.4 Procedimento de decomposição

A partir do trabalho de Fumero e Vercellis (1999), adotamos um procedimento de decomposição para tentar decompor o problema integrado em dois subproblemas mais fáceis de serem resolvidos: o subproblema de dimensionamento de lotes e o de roteamento dos veículos. Um procedimento heurístico foi utilizado para combinar as soluções obtidas por um solver para o problema de produção (dimensionamento de lotes) com as soluções do problema de roteamento dos veículos obtidas pelo algoritmo evolutivo. Considerando que as variáveis do problema dimensionamento de lotes estão presentes nesse subproblema,

temos o seguinte modelo para o problema de produção com entregas fixadas para cada período de tempo:

Minimize

$$\sum_{p=1}^P \sum_{t=1}^T h_{p0t} I_{p0t} + \sum_{p=1}^P \sum_{t=1}^T (s_{pt} y_{pt} + c_{pt} x_{pt}) \quad (3.30)$$

Sujeito a:

$$x_{pt} + I_{p0,t-1} - \sum_{i=1}^N d_{pit} = I_{p0t} \quad 1 \leq p \leq P \quad 1 \leq t \leq T \quad (3.31)$$

$$\sum_{p=1}^P b_p x_{pt} \leq B_t \quad 1 \leq t \leq T \quad (3.32)$$

$$x_{pt} \leq M y_{pt} \quad 1 \leq p \leq P \quad 1 \leq t \leq T \quad (3.33)$$

$$I_{p0t} \leq U_{p0} \quad 1 \leq p \leq P \quad 1 \leq t \leq T \quad (3.34)$$

$$y_{pt} \in \{0, 1\} \quad 1 \leq p \leq P \quad 1 \leq t \leq T \quad (3.35)$$

$$x_{pt}, I_{p0t} \geq 0 \quad 1 \leq p \leq P \quad 1 \leq t \leq T \quad (3.36)$$

A função objetivo (3.30) minimiza os custos de estoques dos produtos na fábrica e os custos de preparação e produção dos itens. As restrições (3.31) estabelecem o balanceamento de estoque na fábrica de acordo com a produção e as demandas dos clientes. É importante observar que, nas restrições (3.31), utilizamos as demandas modificadas ao assumirmos que $\sum q = d$, garantindo assim a independência entre os problemas. Caso as quantidades de entregas ainda não tenham sido definidas, atualizamos as demandas originais subtraindo os estoques iniciais da fábrica e dos clientes. As restrições (3.32) definem a capacidade máxima de produção da fábrica em cada período. As restrições (3.33) definem a relação entre as variáveis de produção e preparação e as restrições (3.34) definem a capacidade máxima de estoque dos itens na fábrica. As restrições (3.35) definem as variáveis binárias e as restrições (3.36) a positividade das variáveis inteiras.

Apresentamos no capítulo a seguir as meta-heurísticas evolutivas desenvolvidas neste trabalho.

4 ALGORITMOS EVOLUTIVOS

Apresentamos neste capítulo os algoritmos genéticos e meméticos propostos para determinar uma solução do problema de produção e roteamento. Os algoritmos genéticos, propostos por John Holland em meados da década de setenta, surgiram a partir da analogia entre os mecanismos de seleção natural e da genética, além das técnicas de busca para obter soluções de problemas de otimização.

Em um algoritmo genético, um conjunto de indivíduos, chamado de população inicial, é gerado aleatoriamente. A cada indivíduo é atribuído um valor de *fitness*, em geral, relacionado com o valor da função objetivo. Desse modo, cada indivíduo representa um ponto do espaço de busca do problema. Os operadores de reprodução, cruzamento e mutação são então aplicados em pares de indivíduos (pais) que foram escolhidos por meio de um mecanismo de seleção, dando origem a novos indivíduos (filhos). As repetições deste processo produzirão um novo conjunto de indivíduos que formarão uma nova população, a princípio mais evoluída, considerando tratar-se de um modelo de evolução que incorpora os conceitos de sobrevivência e seleção do mais adaptado.

Ao incorporar estratégias de busca local nos algoritmos genéticos, temos os chamados algoritmos meméticos (Moscato e Norman (1992)) ou ainda algoritmos genéticos híbridos. Nos algoritmos meméticos, os indivíduos passam por um processo de evolução cultural que ocorre por meio de uma busca local aplicada aos filhos após a execução dos operadores de reprodução. Nesse caso, os indivíduos recebem a denominação de agentes e possuem informações específicas do problema que foram incorporadas pela busca local aplicada.

Nas Figuras 1 e 2 ilustramos, em fluxogramas, a estrutura elementar de funcionamento destas meta-heurísticas.

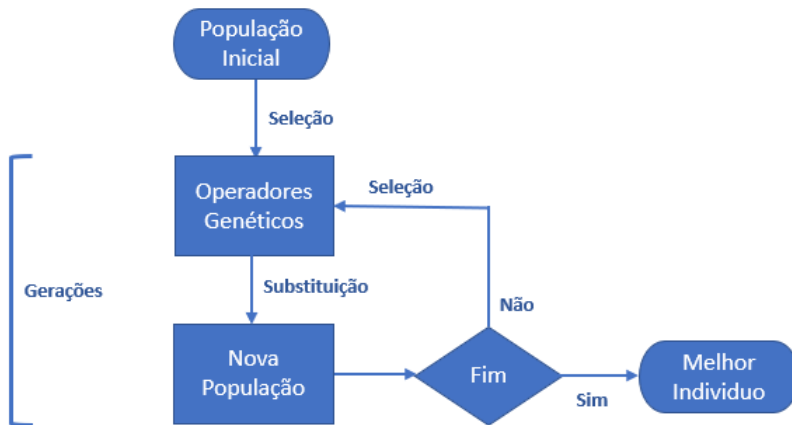
4.1 Algoritmos Evolutivos com População Estruturada

Descrevemos nas próximas seções, os elementos básicos que definem os algoritmos evolutivos propostos, assim como os aspectos relevantes e necessários à sua implementação no sentido de obter uma solução para o PRP. Importante destacar que estes elementos serão idênticos para ambos os métodos, exceção feita à busca local que será aplicada depois dos operadores de reprodução somente no algoritmo memético.

4.1.1 Representação da solução

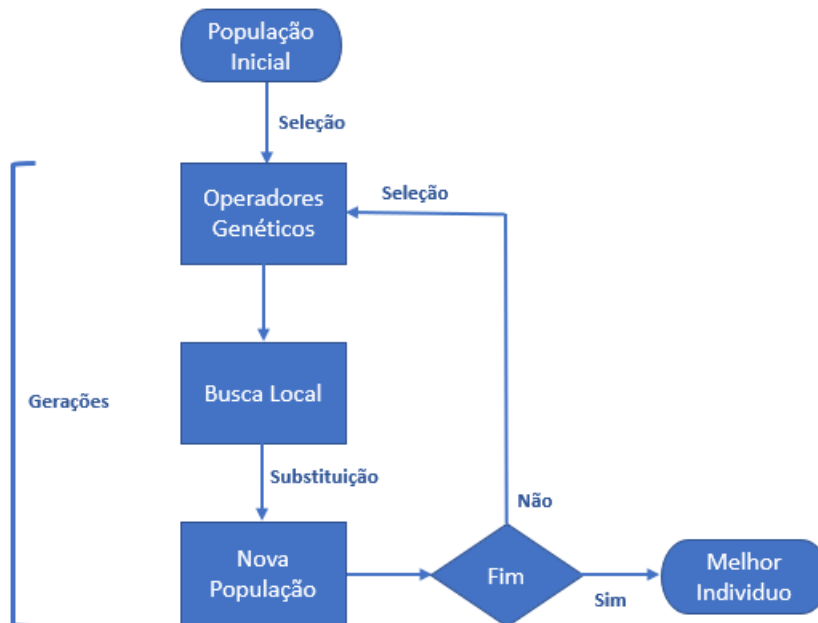
Uma das primeiras questões que surgem na implementação de um algoritmo genético ou memético é como representar as características de uma solução usando um indivíduo da população. Como o PRP possui um grande número de variáveis, a primeira preocupação é

Figura 1 – Fluxograma típico de um Algoritmo Genético.



Fonte: Autoria própria

Figura 2 – Fluxograma típico de um Algoritmo Memético.



Fonte: Autoria própria

adotar uma estrutura eficiente do ponto de vista computacional, que permita manipular informações do problema e viabilize a aplicação dos operadores de reprodução, garantindo assim o transcurso do processo evolutivo.

Optamos por utilizar inicialmente nesse trabalho uma representação completa da solução do problema, ou seja, um indivíduo armazenará as quantidades produzidas, os

estoques e as informações referentes as rotas percorridas pelos veículos. Na Figura 3 temos uma representação gráfica desta estrutura.

Figura 3 – Estrutura (classe) de um indivíduo.

| Indivíduo | | |
|-----------------|----------------|-------------------|
| Produção | Estoque | Roteamento |
| x_{pt} | I_{pit} | r_{pvikt} |
| y_{pt} | | q_{pvit} |
| | | z_{vikt} |

Fonte: Autoria própria

4.1.2 Função de avaliação

A função de avaliação estabelece um valor, denominado *fitness*, para cada indivíduo como uma medida da sua capacidade de adaptação dentro da população e que reflete a qualidade desta solução para o problema. Importante ressaltar que os indivíduos mais adaptados são aqueles que conseguem perpetuar suas características nas gerações vindouras por meio dos mecanismos de reprodução.

Uma ideia natural que surge é utilizar a própria função objetivo como função de avaliação, ou seja, usar o custo desta solução como sendo o valor de *fitness*. No entanto, em uma população podemos ter soluções factíveis e infactíveis, inviabilizando apenas o uso do custo. Alternativamente, podemos definir uma função de avaliação que armazena o custo e uma medida relacionada ao nível de infactibilidade de cada indivíduo. Assim, estabelecemos diferentes níveis de infactibilidade de acordo com a quantidade de restrições que estão sendo violadas, isto é, um número inteiro indicando os tipos de restrições não atendidas, como, por exemplo, restrições de capacidade, estoque na fábrica ou nos clientes, capacidade dos veículos e assim por diante. Desse modo, definimos a seguinte função de avaliação

$$f(ind) = (custo, \mu)$$

Sendo:

ind = indivíduo;

$custo$ = valor da solução na função objetivo;

μ = número de restrições violadas.

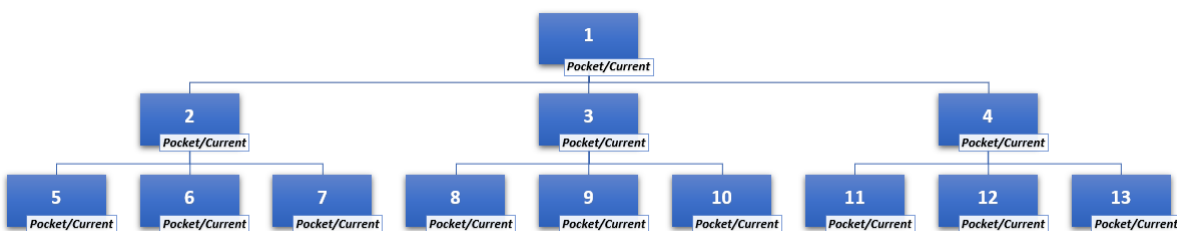
Portanto, o valor de *fitness* será definido por um par ordenado que armazena o custo e a quantidade de restrições violadas por este indivíduo. Obviamente, se o indivíduo for factível então $\mu = 0$.

4.1.3 Estrutura populacional

Diversas estratégias tem sido utilizadas nas últimas décadas para o gerenciamento eficiente das populações. Boudia e Prins (2009) analisam diversas estratégias com o objetivo de enfrentar a redução de diversidade ao longo das gerações de um algoritmo memético aplicado ao PRP com apenas um item. Um aspecto diretamente relacionado refere-se ao tamanho da população, pois populações muito pequenas podem reduzir demais o espaço de busca ao passo que populações muito grandes podem afetar o desempenho computacional do método. Populações muito pequenas tendem a perder rapidamente diversidade após aplicações sucessivas do operador de cruzamento. Embora resultados teóricos indiquem uma relação exponencial entre o tamanho ideal da população e o tamanho da solução de um problema, na prática muitos pesquisadores tem obtido excelentes resultados quando utilizado populações com um número pequeno de indivíduos.

Desse modo, com o intuito de organizar os indivíduos adotamos uma população estruturada em árvore. Esta estrutura foi proposta inicialmente por Berretta e Moscato (1999) para o problema de partição de números e também foi utilizada em outros problemas de planejamento da produção (França, Mendes e Moscato (2001), Mendes, Franca e Moscato (2002), Berretta e Rodrigues (2004), Toledo *et al.* (2009)). O modelo consiste em uma árvore ternária com três níveis e um total de 26 indivíduos, conforme a Figura 4.

Figura 4 – Estrutura populacional em árvore.



Fonte: Adaptado de Berretta e Moscato (1999)

A população é organizada hierarquicamente por líderes e subordinados, classificados de acordo com o valor atribuído pela função de avaliação. Desse modo, temos uma estrutura ordenada de acordo com a qualidade das soluções que estes indivíduos representam para o problema. Ao analisar cada nível da árvore, podemos dividir a estrutura em 4 sub-populações. Assim, o agente 1 será o líder da sub-população que tem como subordinados os agentes 2, 3 e 4. O agente 2 será líder da sub-população que tem como subordinados os

agentes 5, 6 e 7 e assim por diante. Cada agente armazena dois indivíduos, um denominado *pocket* e o outro denominado *current*, totalizando 26 indivíduos na população. O *pocket* terá sempre melhor valor do que o *current* e quando isso não ocorre os indivíduos são trocados de posição. O *pocket* de um líder possuirá melhor valor do que os seus subordinados. Sempre que modificações forem realizadas nos indivíduos, um procedimento de atualização hierárquica será efetuado, garantindo, assim, a verificação entre *pockets* e *currents* e também entre líderes e subordinados. Desse modo, teremos sempre armazenado no indivíduo *pocket* do agente 1, o melhor indivíduo da população.

4.1.4 População inicial

A população inicial pode ser gerada aleatoriamente ou utilizando métodos mais específicos que consideram as características do problema abordado. Em problemas com muitas restrições, esta estratégia pode ser bastante útil, pois auxilia na geração de indivíduos representando soluções factíveis para o problema. Nesse caso, por se tratar de um problema extremamente restrito, utilizamos uma combinação de estratégias, amplamente empregadas para obter soluções do problema de roteamento e do problema de dimensionamento de lotes isoladamente, para gerar um conjunto inicial de soluções que irão compor a população inicial.

Dessa forma, o processo para obter uma solução para o PRP foi dividido em duas etapas. Na primeira etapa, uma heurística foi utilizada para determinar uma solução para o problema de roteamento dos veículos, garantindo o atendimento das demandas dos clientes. Com a definição das entregas, os estoques dos clientes são atualizados. Na segunda etapa, uma heurística determina a solução do problema de dimensionamento de lotes, utilizando as demandas estabelecidas na primeira etapa. Por fim, incorporando as decisões de roteamento com o plano de produção temos uma solução para o problema integrado e, portanto, um indivíduo para a população inicial. Convém destacar que a ordem destas etapas pode ser invertida.

4.1.4.1 Solução inicial: Problema de roteamento

Detalhamos, a seguir, as heurísticas implementadas para determinar as soluções para o problema de roteamento dos veículos.

- i) **BFD**: A heurística BFD (*best fit decreasing*) resolve um problema de empacotamento binário em cada período, atribuindo as cargas aos veículos de modo a atender a demanda dos clientes. Os pedidos dos clientes são carregados nos veículos, do maior para o menor, respeitando-se as capacidades de carga. Se o número de veículos da frota for insuficiente para atender a demanda deste período então um procedimento de ajuste é executado. Tal procedimento realiza tentativas de carregamento dos lotes

remanescentes em períodos anteriores, priorizando veículos com maior capacidade ociosa.

- ii) **BFD-Inverse**: Com o intuito de gerar vários indivíduos para compor a população inicial implementamos uma segunda versão da heurística BFD, invertendo a ordem de carregamento dos pedidos nos veículos, ou seja, os pedidos serão carregados do menor para o maior pedido. Do mesmo modo que na BFD, se a demanda não for atendida para algum período então um procedimento de ajuste é executado.
- iii) **BFD-Rand**: Por fim, na terceira variante da heurística BFD, a ordem de carregamento dos pedidos será aleatória, ou seja, será obtida a partir do sorteio dos pedidos em cada período. Analogamente as versões anteriores, o procedimento de ajuste, caso necessário, será aplicado.
- iv) **CW**: Implementamos também uma versão sequencial da heurística de Clarke e Wright (1964), conhecida como método das economias, que é amplamente utilizada para o problema de roteamento de veículos (Cordeau *et al.* (2002), Alvarenga, Oliveira e George H. G. Fonseca (2020)), assim como para o PRP (Armentano, Shiguemoto e Løkketangen (2011)). Esta heurística considera uma frota homogênea de veículos capacitados partindo de um depósito para atender um conjunto de clientes. A princípio, considera que existe um veículo atendendo cada cliente e voltando para o depósito. Em seguida, determina a economia obtida com a inserção de outro cliente na rota ou mesmo da junção de duas rotas, desde que satisfeitas as restrições de capacidade. Para o cálculo das economias são utilizadas as distâncias entre os clientes. O processo é continuado até que todos os clientes estejam inseridos em alguma rota e nenhuma economia seja possível. Caso o número de veículos não seja suficiente, novamente um procedimento de factibilização se faz necessário.
- v) **CW-Rand**: Para viabilizar a geração da população inicial, modificamos, aleatoriamente, os parâmetros para o cálculo das economias em cada período, de modo a gerar um número suficiente e diverso de soluções. Com a matriz de distâncias modificada a heurística CW é aplicada normalmente.

4.1.4.2 Solução inicial: Problema de produção

A seguir, apresentamos as heurísticas implementadas para determinar as soluções para o problema de produção.

- i) **WW**: Adaptamos o algoritmo de Wagner e Whitin (1958) que utiliza programação dinâmica para resolver de maneira ótima o problema de dimensionamento de lotes não capacitado com apenas um item ao longo de um horizonte de planejamento. Se o plano de produção obtido violar as restrições de capacidade então um procedimento

de ajuste deverá ser aplicado. Nesse procedimento são realizadas tentativas de transferências de produção para os períodos anteriores de modo a factibilizar as restrições violadas.

- ii) **WW-Rand**: A fim de garantir a geração não apenas de um, mas de diversos indivíduos para comporem a população inicial, introduzimos uma componente aleatória que modifica os custos de preparação para garantir a diversidade das soluções obtidas. A partir dos custos de preparação modificados, a heurística WW é aplicada.
- iii) **LL**: Por fim, a heurística lote por lote foi implementada para obter um plano de produção. Essa heurística determina, em cada período, as quantidades necessárias de produção para atender a demanda sem violar as restrições de capacidade. Caso algum lote não tenha sido produzido, por exceder a capacidade de produção, um procedimento de ajuste será aplicado para verificar a possibilidade de antecipação desses lotes remanescentes em períodos anteriores, desde que tenham ainda capacidade ociosa.

Cabe ainda observar que os procedimentos para ajuste das cargas e das restrições de capacidade podem falhar e, assim, teríamos a inserção de uma solução infactível na população inicial. Devemos evitar um número elevado de soluções infactíveis na população, pois poderia inviabilizar o processo de busca pela solução ótima do problema. Por outro lado, a presença de algumas soluções infactíveis pode ajudar a manter a diversidade da população, evitando a estagnação da busca e possibilitando a exploração de outras regiões do espaço de soluções.

4.1.5 Mecanismos de seleção e substituição

Considerando que a população inicial foi constituída, precisamos definir como os indivíduos serão selecionados para aplicação dos operadores de reprodução e, em seguida, como os novos indivíduos serão introduzidos na população para formar uma nova geração de indivíduos como parte do processo evolutivo desta população.

A partir da estrutura populacional adotada, definimos dois critérios para seleção e substituição de indivíduos na população.

- i) **selectPockets**: Nesse critério serão selecionados somente os indivíduos denominados de *pockets*. Em cada sub-população, o *pocket* do agente líder será selecionado para aplicação dos operadores de reprodução (cruzamento e mutação) com os indivíduos *pockets* dos agentes subordinados. Desta forma, temos quatro sub-populações e, em cada sub-população, existem três subordinados. Portanto, serão realizadas três recombinações em cada sub-população, gerando, assim, doze novos indivíduos a cada etapa deste processo que denominamos de uma nova geração.

- ii) **selectCurrents**: Nesse caso, somente os indivíduos *currents* serão selecionados. Do mesmo modo que no critério *selectPockets*, o indivíduo *current* do agente líder será selecionado para aplicação dos operadores de reprodução juntamente com os indivíduos *currents* dos agentes subordinados.

A cada aplicação dos operadores de reprodução será gerado um indivíduo filho que substituirá o indivíduo *current* do agente subordinado. Supondo que o *pocket* do agente 1 (líder) tenha sido selecionado, juntamente com o *pocket* do agente 2 (subordinado), o indivíduo resultante desta operação substituirá o indivíduo *current* do agente 2. O critério de substituição será o mesmo se o critério *selectCurrents* for aplicado. Além disso, esta substituição não precisa estar condicionada a qualidade da solução gerada, pois temos a garantia que a solução substituída não é melhor do que os pais que foram selecionados.

4.1.6 Operadores genéticos

Com o objetivo de combinar as características dos indivíduos para gerar um novo indivíduo que será introduzido na população são executados os operadores de cruzamento e mutação. No operador de cruzamento ocorre a troca de material genético entre os pais para gerar novos indivíduos, denominados filhos. O objetivo principal deste operador é combinar as melhores características dos pais para gerar filhos mais evoluídos. Dessa forma, a aplicação sucessiva deste operador provocará a convergência da população ao longo das gerações. No entanto, a convergência prematura da população pode fazer com que o processo de busca fique estagnado em algum ótimo local, deixando de explorar outras regiões promissoras. Assim como ocorre na natureza, o operador de mutação é então aplicado para aumentar a diversidade da população. Este operador consiste em, de acordo com uma certa taxa de probabilidade, efetuar pequenas alterações nos filhos no momento em que estão sendo gerados.

Uma das maiores dificuldades na implementação de um algoritmo memético está na adaptação dos operadores de reprodução tradicionais para problemas envolvendo um grande número de restrições, como é o caso do PRP. Ao efetuar o cruzamento entre dois indivíduos podemos gerar soluções infactíveis, surgindo então a necessidade de incorporação dos procedimentos para tentar recuperar a factibilidade destas soluções. Embora sejam permitidas soluções infactíveis na população, estes procedimentos precisam ser aplicados para preservar minimamente a factibilidade da população.

Considerando os critérios de seleção previamente definidos, denotaremos por mãe o indivíduo líder e por pai o indivíduo subordinado, com o intuito de facilitar a apresentação dos operadores propostos. Importante destacar que, com exceção do operador *crossOX1* que é um operador clássico do TSP, os demais operadores são contribuições desse trabalho. Dentre os 5 operadores de cruzamento e de mutação descritos a seguir, temos que o operador *crossOPP* modifica apenas as variáveis de produção, os operadores *crossOPR* e

crossOX1 modificam apenas o roteamento e os demais, crossXLF e crossTWP, modificam a produção e o roteamento simultaneamente.

- i) **crossXLF**: Trata-se de um operador de cruzamento de um ponto, no qual o filho recebe o plano de produção da mãe e o roteamento do pai ou vice-versa. Essa escolha ocorre por meio de um sorteio com igual probabilidade. O filho será factível sempre que não houver ajustes no roteamento que foi herdado dos pais, uma vez que durante o ajuste podem ocorrer antecipações de entregas, o que poderia afetar o plano de produção original. De qualquer modo, se o filho for inactível aplica-se um procedimento de factibilização. O operador de mutação consiste em permutar, conforme a taxa de mutação definida, dois clientes em uma rota escolhida ao acaso na solução filho. O funcionamento desse operador encontra-se detalhado no Algoritmo 1.
- ii) **crossOPR**: Nesse operador, descrito no Algoritmo 2, o filho recebe todas as informações da mãe e um ponto de corte, escolhido ao acaso entre $[1, T]$, define quais informações do roteamento do pai serão herdadas pelo filho. O operador de mutação não permitirá a troca de informações para um período qualquer dentro do intervalo de tempo sorteado, segundo uma taxa de probabilidade. Logo, para esse período escolhido o filho permanecerá com as rotas da mãe.
- iii) **crossOPP**: Como descrito no Algoritmo 3, o filho receberá todas as informações da mãe e um ponto de corte, selecionado ao acaso entre $[1, T]$, o que define quais informações do plano de produção do pai serão herdadas pelo indivíduo filho. Analogamente ao crossOPR, o operador de mutação será então aplicado.
- iv) **crossTWP**: Esse operador de dois pontos equivale a aplicação simultânea dos operadores crossOPP e crossOPR, ou seja, o filho receberá as informações da mãe e, uma vez definidos os pontos de cortes, receberá uma parte do plano de produção (crossOPP) e outra parte do roteamento (crossOPR) do pai. O operador de mutação será aplicado do mesmo modo que ocorre nos operadores individualmente, ou seja, nesse caso será aplicado duas vezes.
- v) **crossOX1**: Esse operador é uma adaptação do OX1 (*order crossover*) proposto por (DAVIS, 1985) para o TSP. O filho recebe todas as informações da mãe e o roteamento é modificado pela aplicação do OX1. Assim, a cada período, as rotas dos veículos são unidas em uma grande rota com múltiplos veículos. Em seguida, um segmento da rota da mãe é copiado para o filho e o restante da rota é completado usando-se a ordem de visitas dos clientes presentes na rota do pai. Por fim, as rotas são distribuídas entre os veículos disponíveis a partir da rota ampliada e respeitando-se a capacidades dos veículos. Note que o número de veículos usados pode ser diferente das soluções pais ou até mesmo inactível.

Algoritmo 1: crossXLF

```

Data: pocket[i], pocket[j]
Result: child
if rand.nextDouble() < 0.5 then
  | child ← pocket[i].(xpt, ypt), ∀p, t;
  | child ← pocket[j].(rpvikt, qpvit, zvikt), ∀p, v, i, k, t;
else
  | child ← pocket[j].(xpt, ypt), ∀p, t;
  | child ← pocket[i].(rpvikt, qpvit, zvikt), ∀p, v, i, k, t;
end
child.updateInventory();
child.setFitness();
child.setFeasible();
if not child.isFeasible() then
  | child.repair();
end
return child

```

Quando trocamos uma pequena quantidade de informações entre os indivíduos, estamos na prática, explorando a vizinhança destas soluções e, portanto, intensificando a busca em uma determinada região. Por outro lado, quando trocamos muitas informações entre os indivíduos ocorre um processo de diversificação das soluções, ou seja, ampliamos o espaço de busca atingindo outras regiões.

A aplicação destes operadores pode provocar perda de factibilidade local ou generalizada nos indivíduos, dependendo das informações que estão sendo alteradas. Embora sejam permitidas soluções infactíveis na população, procedimentos de factibilização precisam ser aplicados para preservar minimamente a factibilidade da população. A seguir descrevemos destes procedimentos.

4.1.7 Algoritmos de factibilização

Os procedimentos de factibilização propostos efetuam deslocamentos das quantidades produzidas, estocadas e carregadas nos veículos para períodos anteriores ou posteriores com o objetivo de reestabelecer as capacidades de produção, de estoque e de carga dos veículos. Neste trabalho, implementamos 6 operadores de factibilização, dentre eles temos i), ii) e iii) que efetuam ajustes na produção e os demais promovem ajustes no roteamento.

- i) **Violação da capacidade de produção:** Esse procedimento efetua transferências de produção para períodos anteriores e com capacidade ociosa, sempre que alguma violação da capacidade da fábrica for identificada. Note que este procedimento pode provocar aumento nos custos, pois os períodos sem produção podem passar a ter que produzir quantidades pequenas para ajustar o limite de capacidade de outro período.

Algoritmo 2: crossOPR

```

Data: pocket[i], pocket[j]
Result: child
child ← pocket[i];
onePoint ← rand.nextInt() ∈ [1, T];
notExchange ← 0;
for t ← onePoint to T do
  if rand.nextDouble() < taxMutation then
    notExchange = t;
    t ← T+1;
  end
end
for t ← onePoint to T do
  if t ≠ notExchange then
    | child ← pocket[j].(rpvikt, qpvit, zvikt), ∀p, v, i, k;
  end
end
child.updateInventory();
child.setFitness();
child.setFeasible();
if not child.isFeasible() then
  | child.repair();
end
return child

```

Além disso, a antecipação da produção pode provocar aumentos relacionados aos custos de estoques.

- ii) **Estoques da fábrica negativos:** Pode indicar que a produção foi feita com atraso, em geral, por conta da antecipação de alguma entrega após a definição do plano de produção inicial. Nesse caso, o lote produzido com atraso precisa ser localizado e antecipado, a partir do período atual ou anterior. Caso não seja possível antecipar o lote inteiro, evitando custos de preparação adicionais, o procedimento tentará antecipar a quantidade mínima necessária para positivar o estoque daquele período.
- iii) **Excesso de produção:** Se os estoques não estão zerados no último período do horizonte de planejamento então, nesse caso, pode ter havido produção em demasia e essas quantidades podem ser retiradas diretamente do plano de produção. Os excessos podem ter sido causados pelos operadores *crossOPP* e *crossTWP*. Embora essas quantidades possam ser utilizadas em um próximo horizonte e não impliquem diretamente em inafectibilidade, na prática provocam aumento direto dos custos de produção e estoques.
- iv) **Estoques dos clientes negativos:** Esse pode ser um indicativo que a demanda não foi atendida ou atendida com atraso. Se o produto foi entregue com atraso então

Algoritmo 3: crossOPP

```

Data: pocket[i], pocket[j]
Result: child
child ← pocket[i];
onePoint ← rand.nextInt() ∈ [1, T];
notExchange ← 0;
for t ← onePoint to T do
    if rand.nextDouble() < taxMutation then
        notExchange = t;
        t ← T+1;
    end
end
for t ← onePoint to T do
    if t ≠ notExchange then
        child ← pocket[j].(xpt, ypt), ∀p;
    end
end
child.updatePlantInventory();
child.repairPlantInventory();
child.setFitness();
child.setFeasible();
if not child.isFeasible() then
    child.repair();
end
return child

```

essa quantidade precisa ser localizada e antecipada para um período menor ou igual ao identificado. Caso contrário, essa demanda precisa ser incluída em algum veículo para entrega também a partir do período atual ou anterior. A prioridade em ambos os casos é a manutenção das rotas já existentes e, caso não seja possível, as rotas serão modificadas ou novas rotas criadas.

- v) **Excesso de entregas:** Se os estoques dos clientes estão positivos no último período para algum produto então o procedimento efetua tentativas de retirada deste excesso em alguma das entregas realizadas. O excesso de entregas pode ter sido causado pelos operadores *crossOPR* e *crossTWP* e provocará estoques negativos na fábrica.
- vi) **Violação da Capacidade de Estoques dos Clientes:** Esse procedimento realiza tentativas de transferências desses excessos para períodos posteriores, priorizando a inclusão em algum carregamento existente daquele cliente. Caso contrário, a rota de algum veículo com capacidade ociosa será alterada.

Nos testes realizados, a aplicação dos operadores genéticos combinada com os procedimentos de factibilização, não detectamos impacto negativo no processo evolutivo

causado pelo excesso de indivíduos inactiváveis na população de indivíduos ao longo das gerações, embora o PRP seja extremamente restrito.

4.1.8 Busca local

A cada novo indivíduo factível inserido na população aplicamos um procedimento para melhoria da qualidade desta solução. Este mecanismo está associado ao processo de evolução cultural dos indivíduos nos algoritmos meméticos, também é comumente chamado de hibridização do algoritmo genético, pois incorpora conhecimentos específicos do problema na busca por soluções melhores. Descrevemos, a seguir, as 9 estratégias utilizadas nesse trabalho, todas aplicadas com o objetivo de melhorar a qualidade das rotas presentes no indivíduo filho que será introduzido na população.

- i) **backwardEmptyVehicle**: O objetivo desse procedimento é esvaziar o veículo com maior capacidade de carga ociosa localizado a partir da segunda metade do horizonte de planejamento. A opção pela parte final do horizonte tem por objetivo facilitar possíveis remanejamentos dessa carga para períodos anteriores, ou seja, antecipar as entregas. Identificado o veículo, são realizadas tentativas de transferências do carregamento para um ou mais veículos em períodos anteriores. Caso o procedimento tenha êxito, o custo fixo pela utilização do veículo deixará de ser computado naquele período. É importante observar que pode ocorrer o esvaziamento parcial do veículo, uma vez que o procedimento analisa a viabilidade das transferências por cliente.
- ii) **forwardEmptyVehicle**: Similar ao procedimento anterior, contudo o objetivo agora passa a ser esvaziar o veículo localizado na primeira metade do horizonte de planejamento postergando as entregas. Ao localizar o veículo, são realizadas tentativas de transferências do carregamento para um ou mais veículos em períodos posteriores.
- iii) **backwardLoad**: Sorteia-se uma rota localizada em um período da segunda metade do horizonte de planejamento e uma tentativa de transferir a carga do primeiro cliente desta rota para trás é executada. Se o movimento for bem sucedido e melhorar o custo total dessa solução então uma nova tentativa será realizada.
- iv) **forwardLoad**: Nesse caso, são realizadas tentativas para postergar as entregas, desde que o atendimento da demanda não seja prejudicado. Uma rota localizada em um período da primeira metade do horizonte de planejamento é sorteada e faz-se uma tentativa de transferir a carga do primeiro cliente desta rota para frente. Se o movimento for bem sucedido então uma nova rota será sorteada e uma nova transferência será analisada.
- v) **2Opt**: Essa heurística consiste em eliminar duas arestas não adjacentes e reconectá-las de outro modo. Para que seja possível sua aplicação, a rota deve ter no mínimo

quatro nós. Se houver melhoria na solução então a alteração é aceita e o processo se repete, além disso a rota e as arestas para eliminação são escolhidas ao acaso.

- vi) **2OptOut**: Diferente da versão anterior, agora temos duas arestas selecionadas e removidas em duas rotas diferentes, dividindo assim, as rotas em duas partes. Em seguida, a parte inicial de uma rota é conectada a parte final da outra rota, gerando duas novas rotas. Duas novas rotas serão sorteadas e o processo se repetirá se houver melhoria na solução.
- vii) **3Opt**: Nessa heurística são eliminadas três arestas não adjacentes que depois serão reconectadas de um modo diferente. Portanto, existem $2^3 - 1$ possibilidades de religamento e a rota deve ter no mínimo seis nós. A rota e as arestas são escolhidas ao acaso e o processo continua enquanto houver melhoria na solução.
- viii) **swapCustomer**: Nesse caso, em uma rota selecionada ao acaso, dois clientes dessa rota também são sorteados e trocados de posição.
- ix) **relocateCustomer**: Um cliente, selecionado aleatoriamente, é removido de uma rota e reinserido na mesma rota, em posição diferente da original, ou ainda, inserido em outra rota que pode ser do mesmo período ou de períodos diferentes.

Embora sejam heurísticas clássicas no contexto dos problemas de roteamento, essas estratégias ou variações delas foram implementadas para o PRP em diferentes situações. Destacamos as buscas locais propostas por Senoussi *et al.* (2018) que utilizam versões mais completas de iii) e iv). Além disso, temos algumas das demais estratégias implementadas por Yagmur e Kesen (2021) para o PRP considerando a programação da produção e frotas heterogêneas e, ainda, o trabalho de Manousakis *et al.* (2022) que considera uma fábrica e um produto.

4.1.9 Estratégias de diversificação e reinícios

Sempre que o processo evolutivo converge para um ótimo local, estagnando a busca por novas soluções, precisamos de estratégias que possibilitem a exploração de outras regiões do espaço de busca. Para monitorar a convergência dos indivíduos ao longo das gerações, definimos dois critérios com o intuito de determinar a diversidade da população. O primeiro, discreto, determina o número de indivíduos com valores de *fitness* diferentes. O segundo critério, contínuo, determina a distância entre o menor e o maior valor de *fitness* da população atual.

Durante o processo de ajuste dos parâmetros observamos que apenas o reinício da população, por meio das estratégias usadas na geração da população inicial, não foi eficiente no sentido de garantir uma nova população capaz de permitir a exploração de diferentes regiões. Utilizamos então as mesmas heurísticas da busca local, mas agora

adaptadas de modo a alterar os indivíduos com movimentos repetidos e combinados entre as diversas estratégias adotadas, o que promove a diversidade necessária entre os indivíduos para compor uma nova população e, reinicia assim, a busca a partir de outros pontos do espaço de busca. Dessa forma, sempre os critérios de diversidade atingirem níveis críticos a população poderá então ser reiniciada por meio dessas estratégias.

Com a definição dos componentes básicos, apresentamos a seguir uma visão geral dos algoritmos evolutivos propostos.

4.1.10 Estrutura básica

Inicialmente são gerados os vinte e seis indivíduos que irão compor a população inicial (veja Figura 3), utilizando os procedimentos descritos na Seção 4.1.4. Caso uma solução seja inactível, um dos procedimentos descritos na Seção 4.1.7 para factibilização desta solução é então aplicado. Se o procedimento falhar o indivíduo será inserido na população mesmo sendo inactível. Em seguida, a função de avaliação (Seção 4.1.2) é aplicada em cada um dos novos indivíduos e a população inicial é ordenada de acordo com a estrutura populacional (Seção 4.1.3). As taxas de diversidade são calculadas e a partir dos mecanismos de seleção estabelecidos (Seção 4.1.5), os operadores de cruzamento e mutação (Seção 4.1.6) são aplicados. Os novos indivíduos gerados e que são inactíveis passam novamente pelo processo de factibilização (Seção 4.1.7). Em seguida, no caso do algoritmo memético, a busca local (Seção 4.1.8) é aplicada somente nos descendentes que são factíveis. Dessa forma, a população é novamente organizada e as taxas de diversidade (Seção 4.1.9) são atualizadas. Se os índices de diversidade atingirem o nível crítico estabelecido então a população será reiniciada (Seção 4.1.9) e reorganizada de acordo com os valores de *fitness*. Por fim, se o critério de parada for atingido então o algoritmo retornará a melhor solução encontrada que estará armazenada no indivíduo *pocket* do agente 1. Caso contrário, novos indivíduos são selecionados para aplicação dos operadores e o processo evolutivo continuará sendo repetido, conforme a Figura 2 e o Algoritmo 4 para o algoritmo memético e a Figura 1 e o Algoritmo 5 no caso do Algoritmo Genético.

Com o objetivo de avaliar o impacto da estrutura populacional na eficiência dos métodos propostos, implementamos, também, uma versão clássica dessas meta-heurísticas, utilizando população não estruturada com critérios de seleção e substituição definidos especificamente para este cenário.

4.2 Algoritmos Evolutivos Clássicos

Apresentamos, nesta seção, alguns aspectos comuns aos algoritmos evolutivos tradicionais que não utilizam população estruturada e fazem uso de mecanismos de seleção menos elitistas do que os definidos anteriormente.

Algoritmo 4: Estrutura básica do MA

```

Data: Parameters - Instance and MA
Result: Best individual at population
Initialize population;
Update populational structure;
Evaluate diversity;
while stop == false do
  forall Population do
    if rand.nextDouble() < 0.85 then
      | A,B ← selectPockets(Population);
    else
      | A,B ← selectCurrents(Population);
    end
    child ← GeneticOperators(A, B);
    if child is not feasible then
      | Repair(child);
    end
    if child is feasible then
      | LocalSearch(child);
    end
  end
  Update populational structure;
  Evaluate diversity;
  if critical diversity achieved then
    | Restart population;
    | Update populational structure;
    | Evaluate diversity;
  end
  if stop criterion reached then
    | stop ← true;
  end
end
return Best individual;

```

Na seleção por torneio, tradicionalmente utilizada nos algoritmos genéticos e meméticos, um conjunto com k indivíduos são selecionados aleatoriamente dentro da população. Dentre esses k indivíduos os melhores são selecionados para aplicação dos operadores de reprodução. Em geral, utiliza-se $k = 2$, ou seja, um torneio com 2 indivíduos selecionados ao acaso que competem entre si e o vencedor (melhor *fitness*) é selecionado como um dos pais. O processo é repetido para seleção do outro pai e, assim, por diante até a seleção de todos os pais, conforme o Algoritmo 6.

Com o intuito de comparar esse mecanismo com os anteriores, fixamos $k = 2$ e o mesmo número de indivíduos que serão selecionados a partir da estrutura em árvore, isto é, serão efetuados 12 cruzamentos e os filhos serão inseridos no lugar dos pais selecionados

Algoritmo 5: Estrutura básica do GA

Data: Parameters - Instance and MA
Result: Best individual at population
 Initialize population;
 Update populational structure;
 Evaluate diversity;
while *stop* == *false* **do**
 forall *Population* **do**
 if *rand.nextDouble()* < 0.85 **then**
 | A,B ← selectPockets(*Population*);
 else
 | A,B ← selectCurrents(*Population*);
 end
 child ← GeneticOperators(A, B);
 if *child is not feasible* **then**
 | Repair(child);
 end
 end
 Update populational structure;
 Evaluate diversity;
 if *critical diversity achieved* **then**
 | Restart population;
 | Update populational structure;
 | Evaluate diversity;
 end
 if *stop criterion reached* **then**
 | *stop* ← true;
 end
end
return Best individual;

com piores valores de *fitness*.

Optamos pela utilização da seleção por torneios porque a escolha dos pais não depende da qualidade (*fitness*) destes indivíduos ou de um conhecimento acerca da população como um todo, o que a torna diametralmente oposta aos critérios de seleção adotados inicialmente.

Apresentamos, a seguir, os experimentos computacionais executados a partir dos algoritmos evolutivos propostos em um conjunto de instâncias geradas aleatoriamente para o PRP com múltiplos produtos. Além disso, comparamos seu desempenho com o estado da arte, envolvendo um conjunto de instâncias clássicas do PRP com um único item.

Algoritmo 6: Seleção por torneios

Data: Population**Result:** Parentscount \leftarrow 1;**while** *count* \leq 24 **do** pai1 \leftarrow selectRand(Population); pai2 \leftarrow selectRand(Population); **if** (*pai1.isFactive()* and *pai2.isFactive()*) or (*!pai1.isFactive()* and *!pai2.isFactive()*) **then** **if** *pai1.getFitness()* \leq *pai2.getFitness()* **then**

| Parentes.add(pai1);

else

| Parentes.add(pai2);

end **else if** *pai1.isFactive()* **then**

| Parentes.add(pai1);

else

| Parentes.add(pai2);

end count \leftarrow count + 1**end****return** Parents

5 EXPERIMENTOS COMPUTACIONAIS

Os Algoritmos Evolutivos propostos nesta tese e suas variantes foram implementados na linguagem Java, JDK versão 14.0.2, com a utilização do solver comercial Gurobi versão 9.1.2, também implementado usando a interface Java disponível para o ambiente. Os testes foram executados em um computador com processador Intel Core i7-10510U (2.3 GHz, cache de 8MB, quad-core), 16Gb RAM DDR4 2666MHz, com sistema operacional Windows 10 Home.

Segundo nosso conhecimento acerca do problema, não foi possível encontrar um conjunto de instâncias com múltiplos produtos que pudesse ser utilizado como referência nesse trabalho. Desse modo, geramos aleatoriamente um conjunto de instâncias com múltiplos produtos, tendo como base as instâncias de Archetti *et al.* (2011) e parâmetros similares aos utilizados por trabalhos predecessores, tais como Armentano, Shiguemoto e Løkketangen (2011) e Mostafa e Eltawil (2019). É importante destacar que serão disponibilizadas todas as instâncias utilizadas conjuntamente aos resultados obtidos pelos métodos propostos nesse trabalho com o intuito de facilitar futuras análises e comparações, além do aperfeiçoamento das técnicas implementadas.

Na próxima seção, descrevemos em detalhes a geração de instâncias com múltiplos produtos. Na Seção 5.2, apresentamos um resumo dos testes executados para obter a melhor combinação de parâmetros e estratégias dos algoritmos propostos. Na Seção ??, por meio de alguns experimentos, evidenciamos o efeito da busca local nos métodos desenvolvidos neste trabalho. Por fim, na Seção 5.5, temos os resultados obtidos para as instâncias com múltiplos itens e, na Seção 5.7, analisamos os resultados para um conjunto clássico de instâncias do PRP com um único item.

5.1 Geração de instâncias

Na Tabela 2 detalhamos os parâmetros utilizados na geração das instâncias com múltiplos produtos. Em relação aos trabalhos da literatura, as principais alterações propostas encontram-se nos parâmetros usados na geração das capacidades de estoques e de estoques iniciais dos clientes, além da capacidade dos veículos. Para os demais parâmetros, utilizamos os trabalhos de Archetti *et al.* (2011) e Armentano, Shiguemoto e Løkketangen (2011) como referências principais.

A Tabela 3 apresenta um resumo geral dos quatro grupos de instâncias com 5, 10, 20 e 30 clientes e 3, 5, 8 e 10 produtos, respectivamente. Os dois primeiros grupos com 6 períodos e os dois maiores com 12 períodos. Além disso, o primeiro grupo de instâncias tem apenas 1 veículo, o segundo 2 veículos e os dois seguintes possuem 5 veículos. As

Tabela 2 – Parâmetros para geração das instâncias com múltiplos produtos.

| | |
|---|--|
| Demandas dos clientes | $d_{pit} \in [10, 100]$ |
| Capacidade máxima de estoque dos clientes | $U_{pi} = \lfloor (\sum_t d_{pit}/T) \cdot \lambda \rfloor$, sendo $\lambda \in \{2, 3, 4, 5\}$ |
| Estoque inicial da fábrica | $I_{p0} = 0$ |
| Estoque inicial dos clientes | $I_{pi0} = \sum_{t=1}^{\tau} d_{pit}$, sendo $\tau \in \{1, 2, 3, 4\}$ |
| Custos de estoque da fábrica | $h_{p0} = 3$ and 8 |
| Custos de estoque dos clientes | $h_{pi} \in [1, 10]$ |
| Capacidade de produção | $B = \lfloor 3 \cdot (\sum_p \sum_i \sum_t d_{pit}/T) \rfloor$ |
| Tempos de processamento | $b_p = 1$ |
| Custos de produção | $c_p = 10 \cdot h_{p0}$ |
| Custos de preparação | $s_p = 100 \cdot h_{p0}$ |
| Coordenadas | $x_k \in [0, 500]$ $y_k \in [0, 1000]$ |
| Capacidade dos veículos | $C = P \cdot (\max_{p,i} U_{pi})$ |
| Custos de transporte | $a_{ik} = \lfloor \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} + 0.5 \rfloor$ |
| Custo fixo de transporte | $f = (N + 1) \cdot (\max_{i,k} a_{ik})$ |

Fonte: Autoria própria

demandas são variáveis e os estoques iniciais nos clientes não nulos. A capacidade de produção da planta é limitada e a de armazenamento é ilimitada, mas os estoques iniciais estão zerados. Assim como Archetti *et al.* (2011), dividimos os grupos em quatro classes conforme a Tabela 4. A Classe I (instâncias de 1 a 10) tem a configuração básica de custos de produção, estoques e de transporte, servindo de base para a geração das demais. A classe II (11 a 20) tem altos custos de produção, o que equivale aos custos da Classe I multiplicados por 10, a Classe III (21 a 30), por sua vez, tem altos custos de transportes, ou seja, os custos serão 5 vezes maiores do que a Classe I. Por fim, a Classe IV (31 a 40) não tem custos de estoques para os clientes. Cada classe tem 10 instâncias com 5 sementes cada e, portanto, temos 200 instâncias para cada um dos grupos, totalizando 800 novas instâncias no conjunto gerado.

5.2 Ajuste dos melhores parâmetros e estratégias

Com o intuito de obter a melhor combinação de parâmetros e estratégias visando a máxima performance dos algoritmos evolutivos, realizamos diversos testes preliminares utilizando amostras dos conjuntos de instâncias propostos nesse trabalho. No sentido de compatibilizar os recursos computacionais disponíveis com a análise desempenho dos métodos utilizando as instâncias geradas para múltiplos produtos, escolhemos a primeira semente de cada um dos grupos de instâncias e fixamos o critério de parada em 300 segundos. Dessa forma, exceto em algumas situações específicas, utilizamos 160 instâncias distribuídas entre os diferentes grupos e classes do conjunto de instâncias gerado nesse trabalho.

Tabela 3 – Visão geral das instâncias com múltiplos produtos.

| instâncias | 5C | 10C | 20C | 30C |
|-------------------------------------|-----------|------------|------------|------------|
| Número de instâncias | 200 | 200 | 200 | 200 |
| Número de períodos | 6 | 6 | 12 | 12 |
| Número de produtos | 3 | 5 | 8 | 10 |
| Número de clientes | 5 | 10 | 20 | 30 |
| Número de veículos | 1 | 2 | 5 | 5 |
| Número de sementes | 5 | 5 | 5 | 5 |
| Demandas | Variável | Variável | Variável | Variável |
| Capacidades de produção | Constante | Constante | Constante | Constante |
| Capacidades de estoque da fábrica | Ilimitado | Ilimitado | Ilimitado | Ilimitado |
| Capacidades de estoque dos clientes | Constante | Constante | Constante | Constante |
| Estoques iniciais da fábrica | 0 | 0 | 0 | 0 |
| Estoques iniciais dos clientes | Variável | Variável | Variável | Variável |
| Capacidades dos veículos | Constante | Constante | Constante | Constante |

Fonte: Autoria própria

Tabela 4 – Descrição das classes de instâncias com múltiplos produtos.

| Classe | Tipo | Descrições |
|---------------|-------------|--|
| Classe I | 1-10 | instâncias padrão |
| Classe II | 11-20 | Custos de produção elevados (Classe I x 10) |
| Classe III | 21-30 | Custos de transporte elevados (Classe I x 5) |
| Classe IV | 31-40 | Custos de estoques dos clientes nulos |

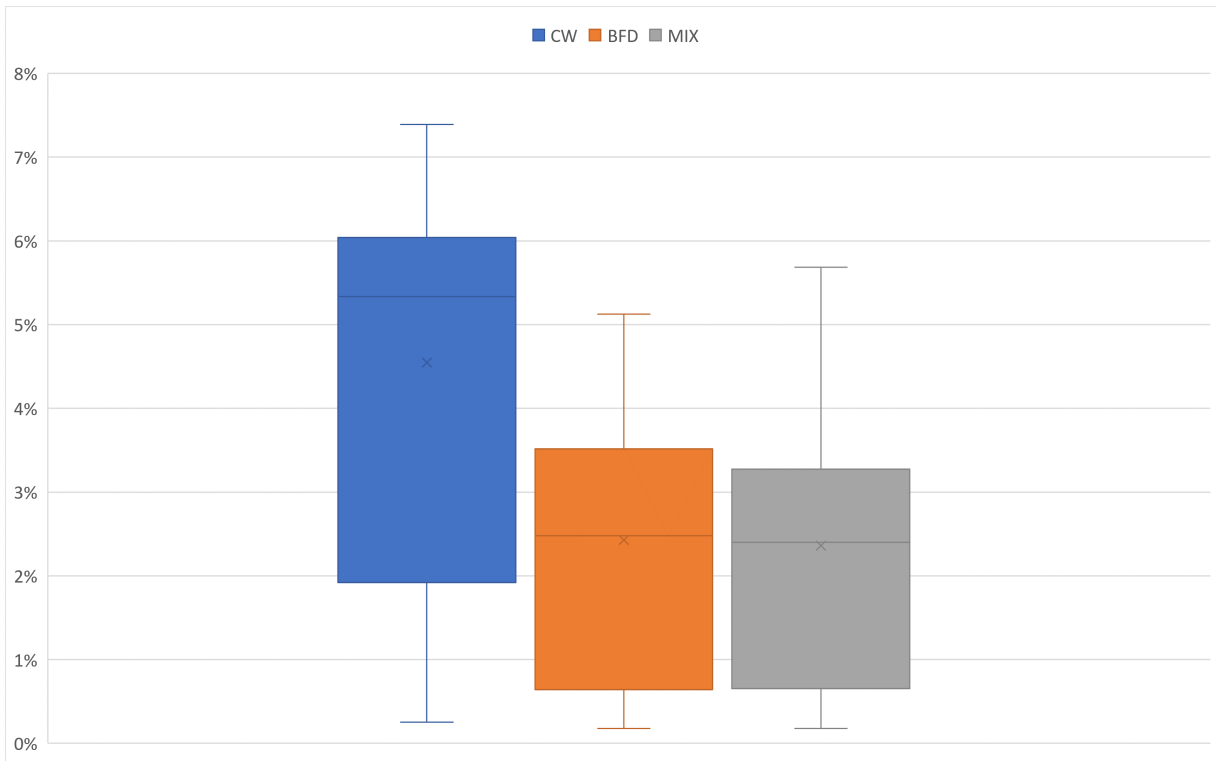
Fonte: Autoria própria

5.2.1 População inicial

Na geração da população inicial, avaliamos a diversidade e a qualidade das soluções obtidas pelas técnicas isoladamente ou por combinações de ambas. Resumidamente, temos na Figura 5 os *gaps* médios obtidos por três estratégias utilizadas na geração. A estratégia CW combina os métodos WW e CW, descritos na Subseção 4.1.4, para obter uma solução completa do problema. A estratégia BFD utiliza os métodos WW e BFD, enquanto a estratégia MIX combina todas as heurísticas implementadas.

Considerando que as três estratégias atingiram a mesma taxa média de diversidade nas instâncias avaliadas, escolhemos a estratégia com o melhor desempenho que foi claramente, pela Figura 5, a que combina todas as heurísticas implementadas para gerar a população com 26 indivíduos, conforme estabelece o Algoritmo 7.

Figura 5 – *Gaps* médios obtidos pelas estratégias utilizadas para geração da população inicial (critério de parada: 300 segundos).



Fonte: Autoria própria

Algoritmo 7: Geração da população inicial.

Data: Instance parameters

Result: Population

pocket[1] \leftarrow CW() + WW();

pocket[2] \leftarrow BFD-Inverse() + WW();

current[1] \leftarrow WW() + BFD();

current[2] \leftarrow CW() + LL();

for $i \leftarrow 3$ to 13 **do**

 pocket[i] \leftarrow WW-Rand() + CW-Rand();

 current[i] \leftarrow BFD-Rand() + WW-Rand();

end

return Population

5.2.2 Mecanismos de seleção

Os mecanismos de seleção dos indivíduos, descritos na Subseção 4.1.5, foram aplicados de acordo com uma taxa de probabilidade. Para obter a melhor combinação, testamos taxas de probabilidades no intervalo variando de 1,0 até 0,5, conforme a Tabela 5. Logo, em um extremo apenas o critério *selectPockets* seria aplicado e, no outro, os dois

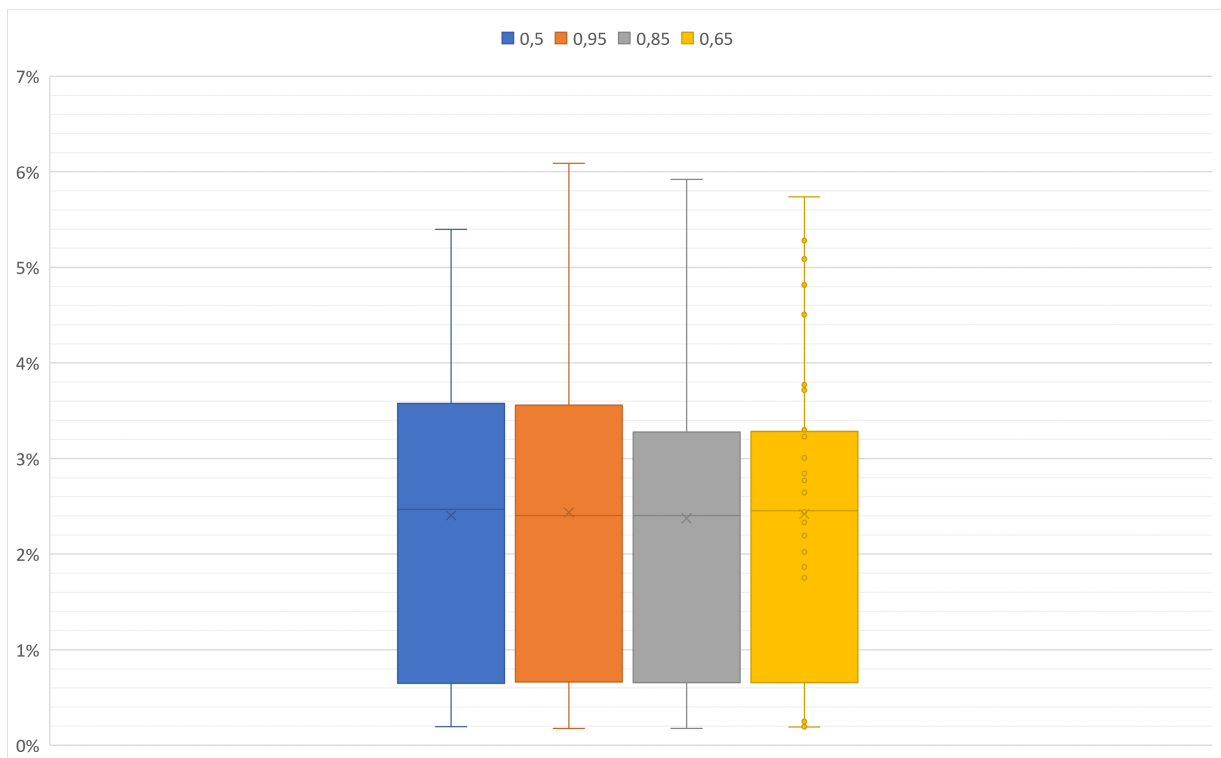
critérios seriam aplicados com igual probabilidade. Importante observar a natureza elitista do critério *selectPockets* e que, somente a sua aplicação, poderia causar a convergência prematura da população, enquanto o segundo critério, por outro lado, ajudaria a retardar esse processo trazendo mais diversidade ao longo das gerações.

Tabela 5 – Taxas de probabilidades testadas para aplicação dos critérios de seleção.

| Critério | Taxas de Probabilidades | | | | | |
|-----------------------|-------------------------|------|-----|-------------|-----|-----|
| <i>selectPockets</i> | 1,0 | 0,95 | 0,9 | 0,85 | ... | 0,5 |
| <i>selectCurrents</i> | 0,0 | 0,05 | 0,1 | 0,15 | ... | 0,5 |

Fonte: Autoria própria

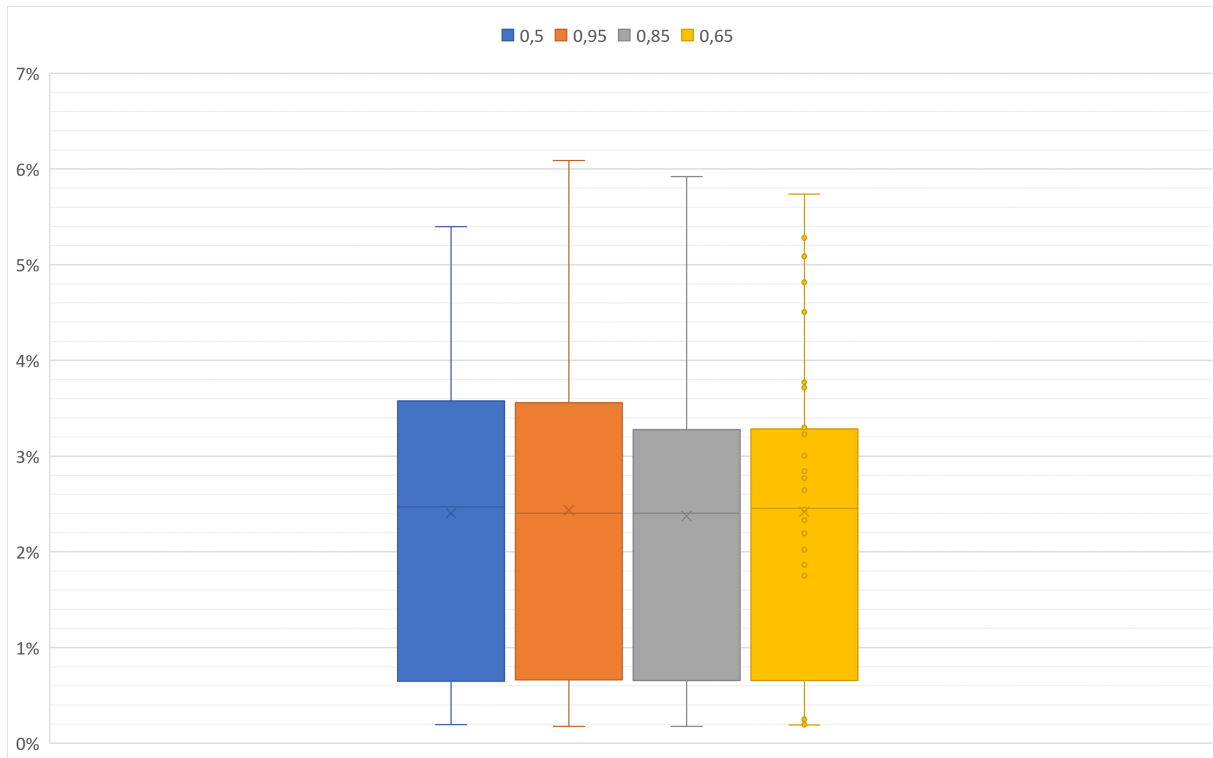
Figura 6 – *Gaps* médios em porcentagens para diferentes combinações dos mecanismos de seleção (critério de parada: 300 segundos).



Fonte: Autoria própria

Na Figura 7 e na Tabela 6 temos os resultados resumidamente apresentados para quatro combinações dos critérios de seleção testados. Embora o desempenho seja bem próximo, podemos concluir que a melhor estratégia para combinação dos critérios de seleção foi obtido pela taxa 0,85, ou seja, o critério que seleciona somente as *pockets* (*selectPockets*) será aplicado com uma taxa de 85%, caso contrário, o segundo critério (*selectCurrents*), que seleciona apenas as *currents*, será então aplicado, conforme o Algoritmo 4.

Figura 7 – *Gaps* médios em porcentagens para diferentes combinações dos mecanismos de seleção (critério de parada: 300 segundos).



Fonte: Autoria própria

Tabela 6 – *Gaps* médios em porcentagens para diferentes combinações dos mecanismos de seleção (critério de parada: 300 segundos).

| Taxas | Classes | | | | Média |
|-------------|---------|------|------|------|-------------|
| | I | II | III | IV | |
| 0,5 | 2,08 | 0,22 | 4,63 | 2,68 | 2,41 |
| 0,95 | 2,05 | 0,23 | 4,65 | 2,81 | 2,44 |
| 0,85 | 2,04 | 0,23 | 4,51 | 2,71 | 2,37 |
| 0,65 | 2,11 | 0,24 | 4,63 | 2,70 | 2,42 |

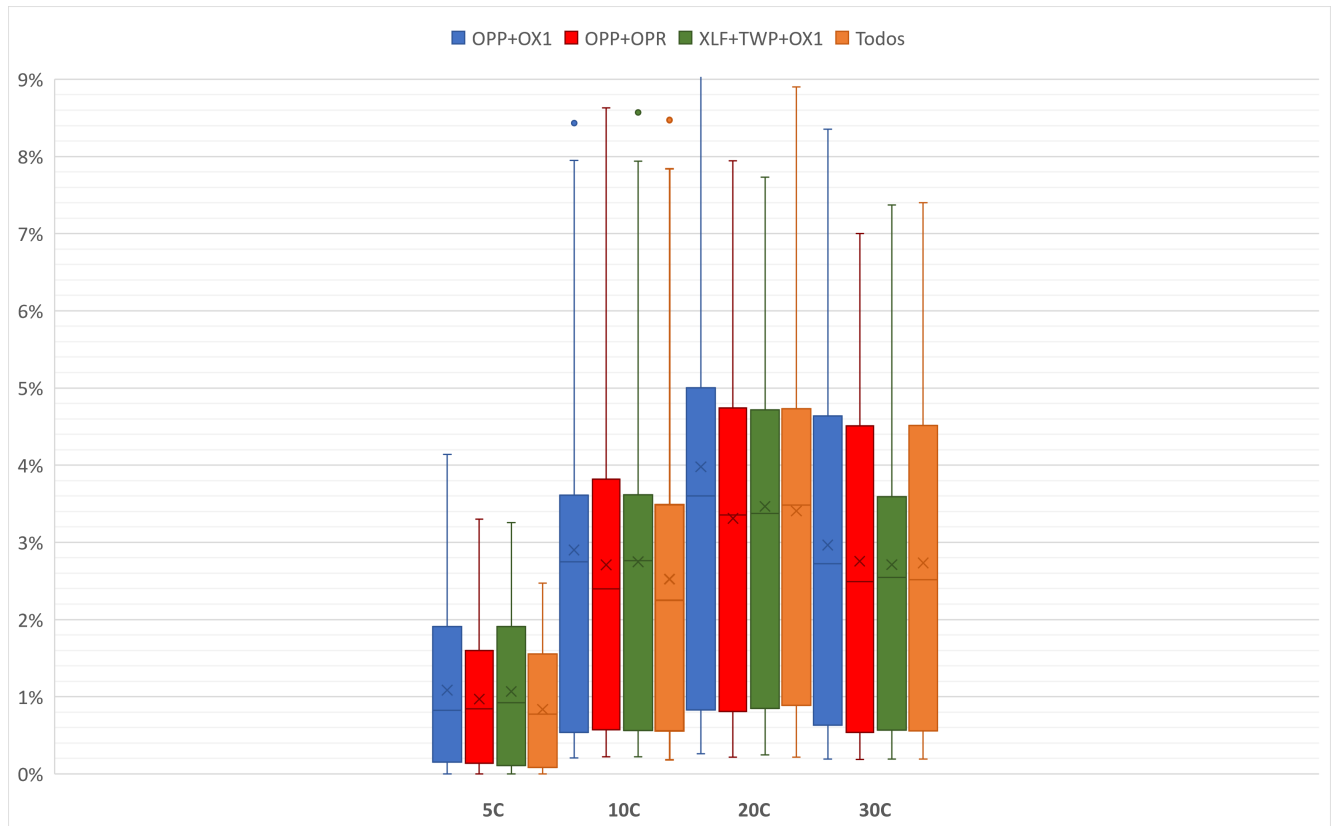
Fonte: Autoria própria

5.2.3 Operadores genéticos

Os operadores de reprodução também tiveram sua eficiência avaliada individualmente ou combinados ao longo do processo evolutivo.

Nas Figuras 8 e 9 e na Tabela 7 temos o desempenho de quatro estratégias combinando os diferentes operadores genéticos propostos neste trabalho. Na Figura 8, apresentamos os *boxplots* com as distribuições envolvendo os *gaps* médios em porcentagens,

Figura 8 – *Gaps* médios dos conjuntos em porcentagens para diferentes combinações dos operadores genéticos (critério de parada: 300 segundos).

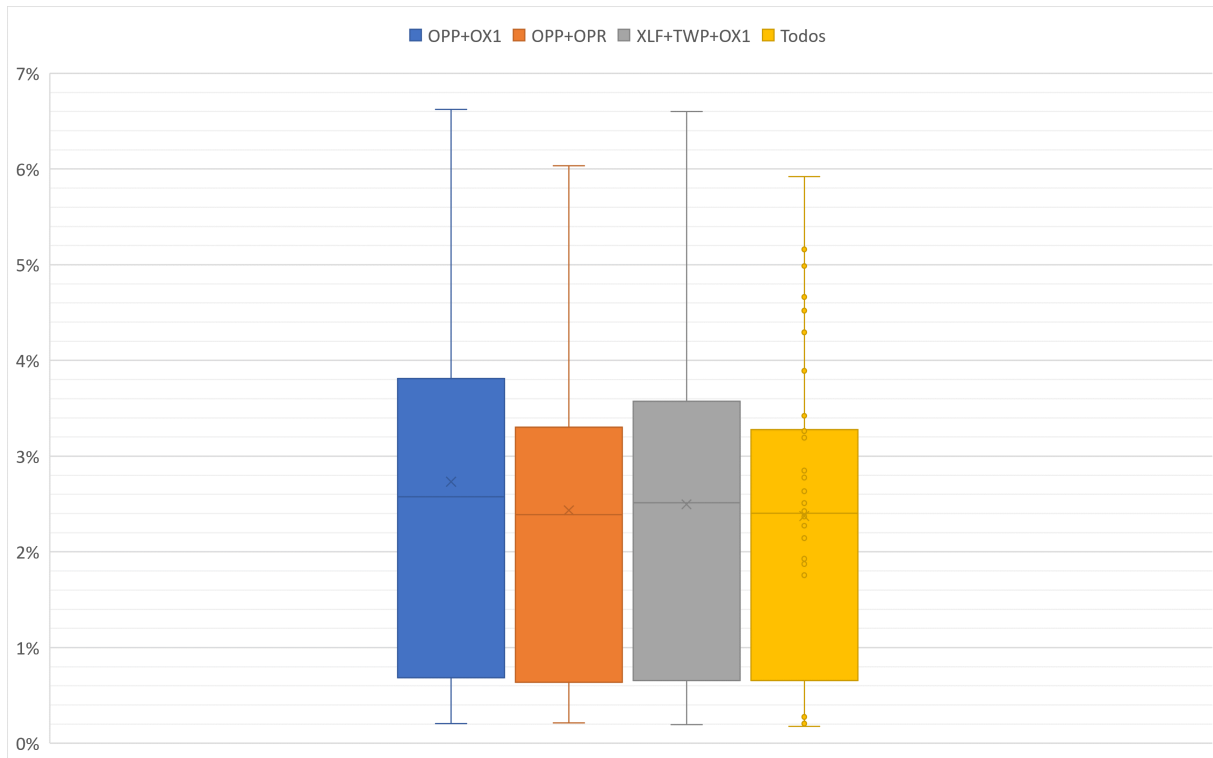


Fonte: Autoria própria

separados de acordo com o número de clientes do conjunto de testes. Podemos observar que a estratégia usando todos os operadores, de maneira geral, foi superior as demais, tendo sido superada apenas no grupo com 20 clientes pela estratégia que utiliza apenas os operadores *crossOPP* e *crossOPR*. Na Figura 9, podemos observar mais facilmente que essas duas estratégias são ligeiramente superiores as demais. Por fim, na Tabela 7, temos os *gaps* médios em porcentagens evidenciando que o melhor resultado geral foi alcançado ao combinar todos os operadores, conforme estabelece o Algoritmo 8.

Em relação à mutação, avaliamos sua aplicação com taxas variando no intervalo $[0,01; 0,2]$ de acordo com as diferentes abordagens implementadas e, na Figura 10, temos os resultados para três diferentes taxas. Embora os resultados sejam bastante próximos, a taxa de 10% de probabilidade foi a que apresentou melhor desempenho e, portanto, foi a taxa adotada para todos os operadores de mutação implementados.

Figura 9 – *Gaps* médios em porcentagens para diferentes combinações dos operadores genéticos (critério de parada: 300 segundos).



Fonte: Autoria própria

Tabela 7 – *Gaps* médios em porcentagens para diferentes combinações dos operadores genéticos (critério de parada: 300 segundos).

| Operadores | Conjuntos | | | | Média |
|--------------------|-----------|------|------|------|-------------|
| | 5C | 10C | 20C | 30C | |
| OPP+OX1 | 1,08 | 2,90 | 3,98 | 2,97 | 2,73 |
| OPP+OPR | 0,97 | 2,71 | 3,31 | 2,75 | 2,44 |
| XLF+TWP+OX1 | 1,07 | 2,75 | 3,46 | 2,71 | 2,50 |
| Todos | 0,84 | 2,52 | 3,41 | 2,73 | 2,37 |

Fonte: Autoria própria

5.2.4 Busca local

Do mesmo modo que no caso dos operadores genéticos, avaliamos o desempenho das heurísticas individualmente e combinadas em grupos menores, de acordo com as suas principais características. Além disso, testamos diversas combinações com diferentes níveis de intensificação das buscas. Com isso, apresentamos a seguir um resumo desses testes com as estratégias que apresentaram melhor desempenho.

Algoritmo 8: Operadores genéticos

```

Data: pocket[i], pocket[j], generation
Result: child
if (generation%5) == 1 then
  | child = crossXLF(pocket[i], pocket[j]);
else if generation %5 == 2 then
  | child = crossOPR(pocket[i], pocket[j]);
else if generation %5 == 3 then
  | child = crossOPP(pocket[i], pocket[j]);
else if generation %5 == 4 then
  | child = crossTWP(pocket[i], pocket[j]);
else
  | child = crossOX1(pocket[i], pocket[j]);
end
return child

```

Depois de concluir que as heurísticas utilizadas na busca local funcionavam melhor juntas, definimos três estratégias para avaliar seu desempenho. A primeira, denominada *Padrão*, é a versão tradicional das buscas locais, ou seja, os movimentos que provocarem melhoria na solução serão aceitos e o processo terá continuidade. Dessa forma, estamos utilizando as heurísticas exatamente da forma como foram definidas na Subseção 4.1.8. Na segunda estratégia, denominada *OneStep*, apenas um movimento é realizado independentemente se houver melhoria ou não da solução. Por fim, a estratégia denominada *Short* é uma versão reduzida das heurísticas em que apenas uma tentativa de movimento será realizada. No caso das heurísticas em que mais de um período ou veículo são verificados, nesta versão eles serão sorteados para garantir que apenas uma tentativa de movimento será efetuada.

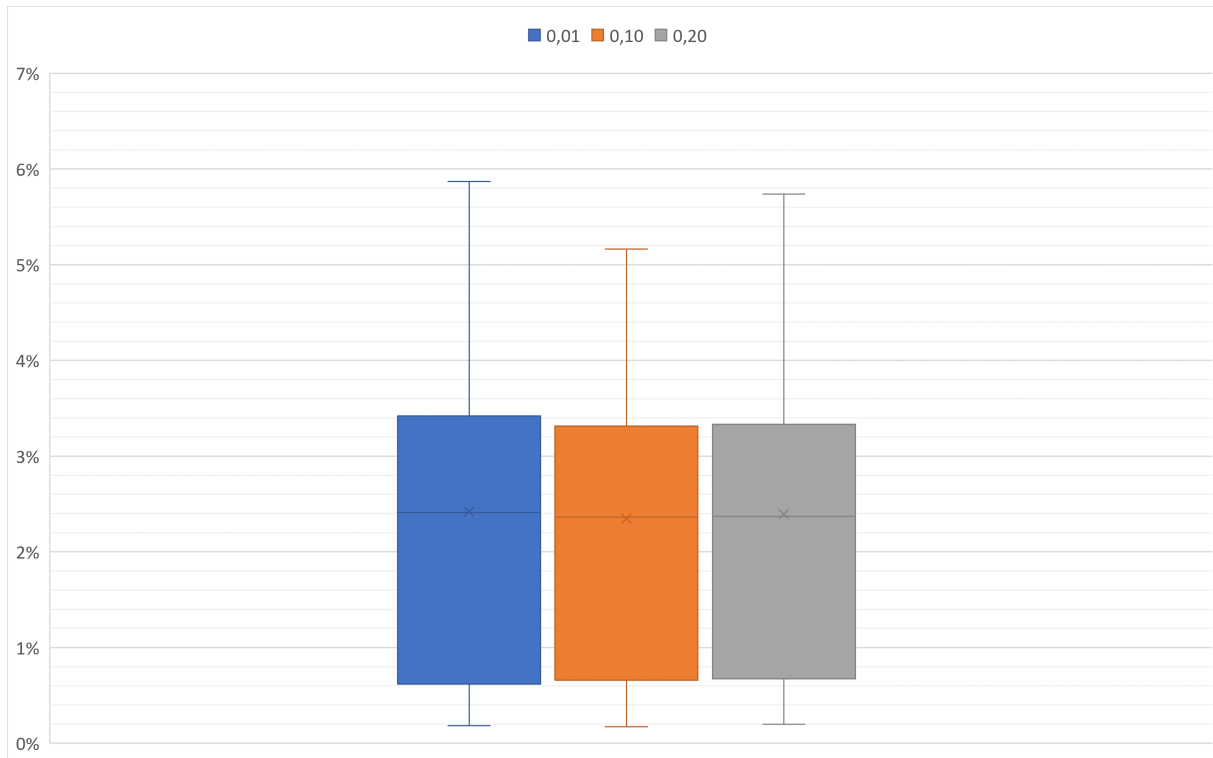
Tabela 8 – *Gaps* médios em porcentagens para diferentes estratégias de aplicação das buscas locais (critério de parada: 300 segundos).

| Busca Local | Classes | | | | Média |
|----------------|---------|------|------|------|-------------|
| | I | II | III | IV | |
| <i>Short</i> | 2,11 | 0,24 | 4,67 | 2,81 | 2,46 |
| <i>OneStep</i> | 2,08 | 0,23 | 4,61 | 2,56 | 2,37 |
| Padrão | 2,05 | 0,23 | 4,53 | 2,59 | 2,35 |

Fonte: Autoria própria

Por fim, avaliamos a necessidade de intensificação da busca local ao longo do processo evolutivo, ou seja, a busca local seria aplicada não apenas uma vez, mas sim repetidas vezes até um determinado número de iterações. Na Tabela 9 temos os *gaps* médios para comparação entre a busca local aplicada uma vez, com 10 e 50 repetições. Claramente, a melhor performance foi atingida sem o processo de intensificação.

Figura 10 – *Gaps* médios em porcentagens para diferentes taxas de mutação (critério de parada: 300 segundos).



Fonte: Autoria própria

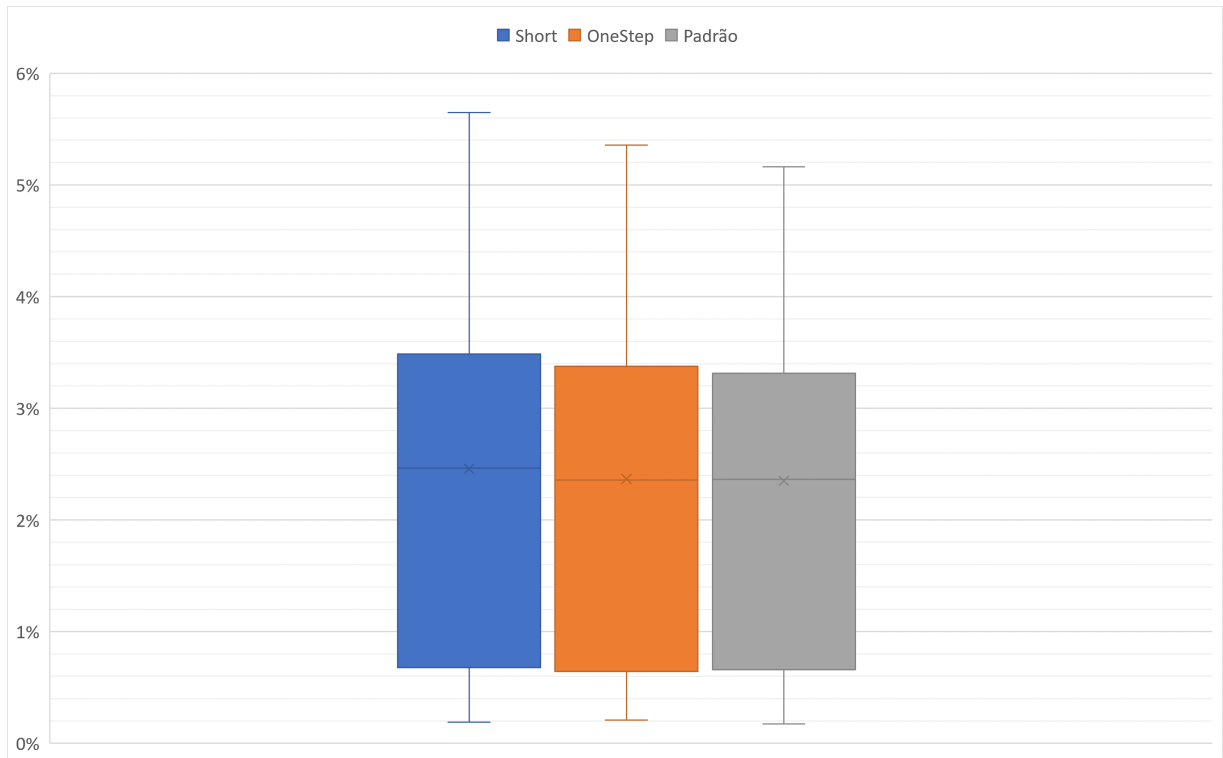
5.2.5 Estratégias de diversificação e reinícios

Ao longo do processo evolutivo, sempre que os índices de diversidade atingiram níveis críticos, mais especificamente a taxa de diversidade contínua for menor que 100 ou a taxa discreta for menor que 4, o mecanismo para diversificação da população foi acionado. Além disso, após 100 gerações sem atualização da incumbente uma nova população inicial foi gerada, reiniciando a busca a partir de uma nova população.

Podemos notar na Tabela 10 e na Figura 12 o impacto na qualidade das soluções, por meio dos *gaps* médios, das estratégias de reinicialização utilizadas. Nos grupos com 5 e 10 clientes temos notadamente desempenho superior das estratégias com reinícios. Por outro lado, nos grupos com 20 e 30 clientes, não há tempo suficiente para convergência da população dentro do limite de tempo máximo de 300 segundos considerado como critério de parada nestes testes e, conseqüentemente, nestas instâncias as duas técnicas tem os mesmos resultados. Assim, a média geral entre as estratégias fica mais próxima influenciada pelos resultados iguais desses dois grupos.

Na Figura 13 podemos observar o efeito das estratégias de reinicialização da população nas taxas de diversidade discreta na primeira instância da Classe II do grupo

Figura 11 – *Gaps* médios em porcentagens para diferentes estratégias de aplicação das buscas locais (critério de parada: 300 segundos).



Fonte: Autoria própria

com 10 clientes. Ao longo de 300 gerações é possível notar que as taxas de diversidade atingiram níveis críticos três vezes, mais especificamente nas gerações 120, 195 e 300. Dessa forma, duas reinicializações ocorreram nas gerações 120 e 195 e os níveis de diversidade foram restabelecidos atingindo resultados próximos aos obtidos com a população inicial.

5.2.6 Critério de parada

Com o objetivo de estabelecer o tempo de parada ideal para execução dos experimentos, avaliamos o comportamento das técnicas durante 6 horas de execução para algumas instâncias dos diferentes grupos de clientes.

Na Figura 14 podemos observar a evolução da incumbente (*fitness versus tempo*) em uma amostra com 4 instâncias, a primeira de cada classe do grupo com 30 clientes. Embora a maior redução de custo ocorra dentro da primeira hora de execução (3600 segundos), podemos notar que o MA continua evoluindo, ainda que mais lentamente ao longo das 6 horas de execução. Esse comportamento também pode ser observado nas instâncias do grupo com 20 clientes, conforme a Figura 15. Contudo, a redução dos custos mostrou-se mais acentuada na primeira hora de execução. A partir dessas análises, foi

Algoritmo 9: Busca local

```

Data: child
Result: child
if child.isFeasible() then
  for  $i \leftarrow 1$  to 10 do
    relocateCustomer(child);
    swapCustomer(child);
    2OptOut(child);
    2Opt(child);
    3Opt(child);
    backEmptyVehicle(child);
    backwardLoad(child);
    forwardLoad(child);
    forEmptyVehicle(child);
  end
end
return child

```

Tabela 9 – *Gaps* médios em porcentagens para diferentes níveis de intensificação da busca local (critério de parada: 300 segundos).

| Intensificação | Conjuntos | | | | Média |
|---------------------|-----------|------|------|------|-------------|
| | 5C | 10C | 20C | 30C | |
| Padrão | 0,72 | 2,39 | 3,51 | 2,78 | 2,35 |
| 10 iterações | 0,79 | 2,78 | 3,82 | 2,96 | 2,59 |
| 50 iterações | 0,82 | 3,06 | 3,98 | 3,07 | 2,73 |

Fonte: Autoria própria

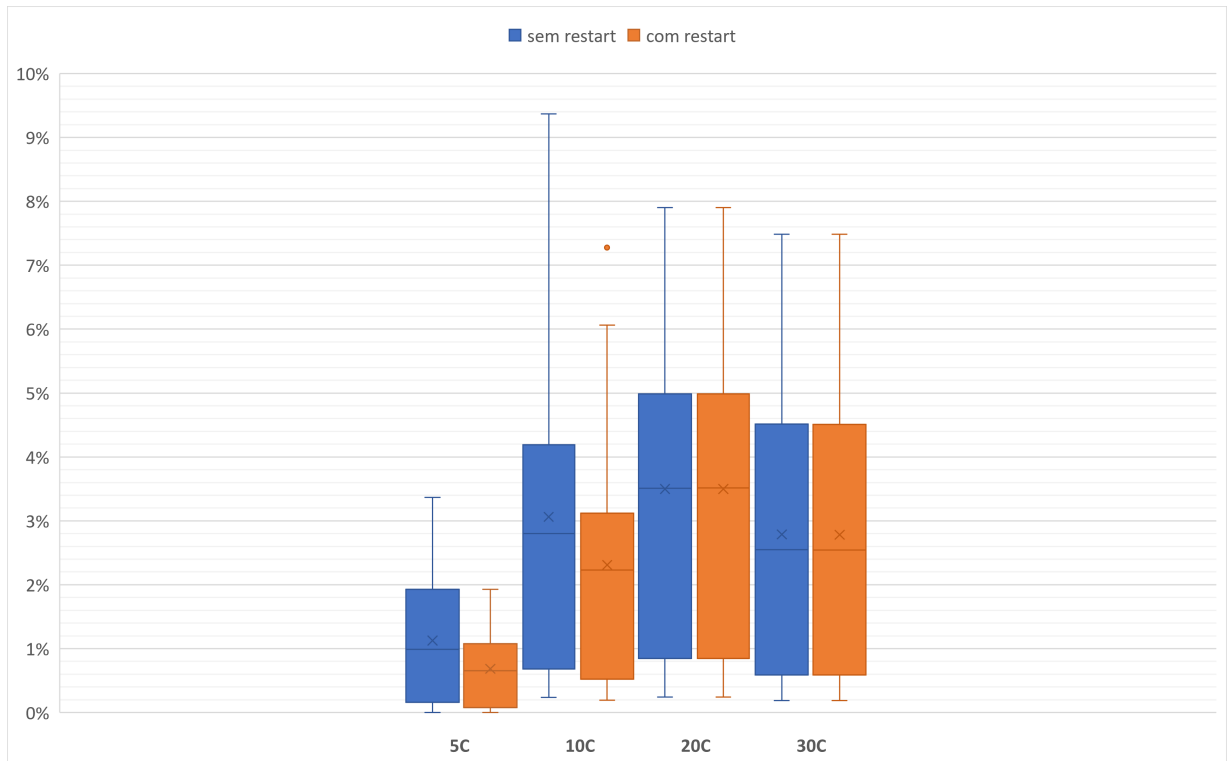
Tabela 10 – *Gaps* médios em porcentagens com e sem reinícios (critério de parada: 300 segundos).

| | Conjuntos | | | | Média |
|----------------------|-----------|------|------|------|-------|
| | 5C | 10C | 20C | 30C | |
| Sem reinícios | 1,13 | 3,06 | 3,50 | 2,78 | 2,62 |
| Com reinícios | 0,68 | 2,31 | 3,50 | 2,78 | 2,32 |

Fonte: Autoria própria

possível inferir que o critério de parada ideal para as instâncias maiores seriam 2 e 3 horas, respectivamente, para os grupos 20C e 30C. Além disso, a evolução da incumbente em grupos com 5 e 10 clientes ficou estagnada dentro da primeira hora de execução. Diante desse cenário e com a busca do equilíbrio entre o número de instâncias e os recursos computacionais disponíveis para execução dos testes, definimos como critério de parada o tempo máximo de 3600 segundos (1 hora) para o solver e o MA em todos os grupos de instâncias.

Figura 12 – *Gaps* médios em porcentagens com e sem reinícios (critério de parada: 300 segundos).



Fonte: Autoria própria

5.2.7 Estrutura populacional e mecanismos de seleção

Analizamos o impacto da estrutura populacional no desempenho do MA proposto ao comparar os resultados obtidos utilizando uma população não estruturada aliada a mecanismos de seleção dos indivíduos tradicionalmente utilizados nos algoritmos genéticos, conforme descrevemos na Seção 4.2.

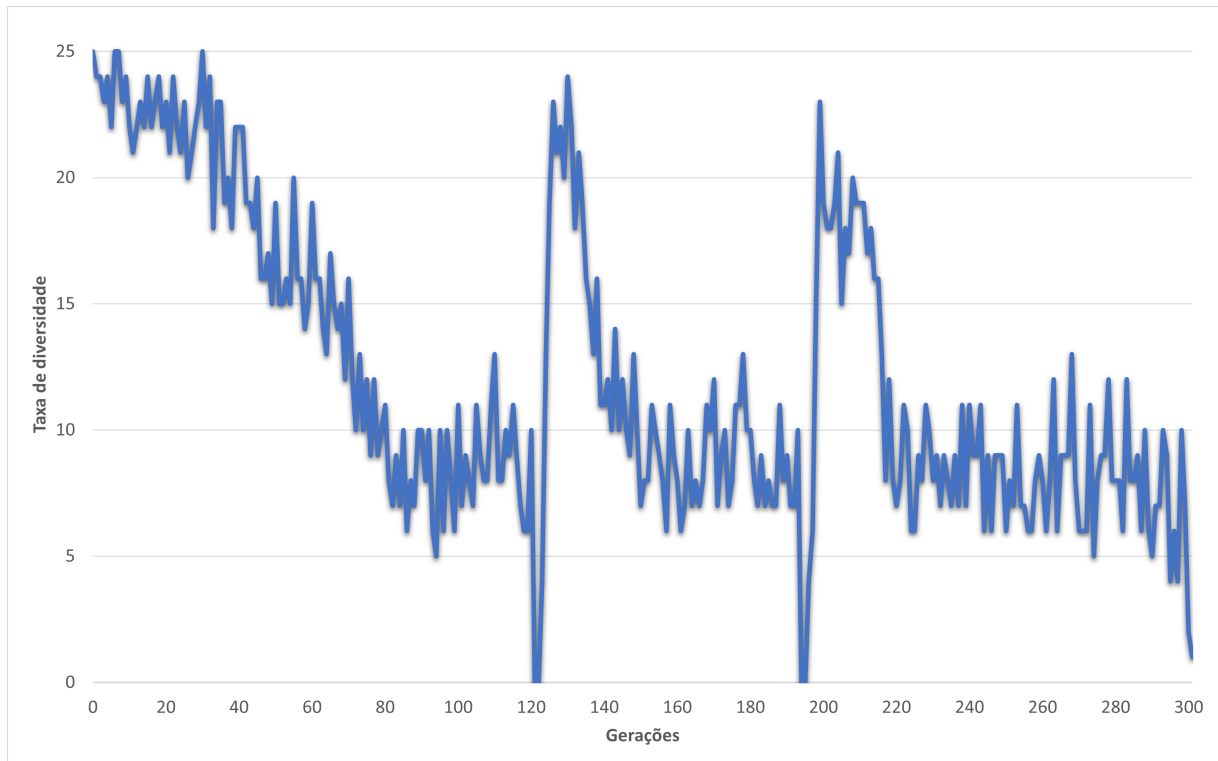
Tabela 11 – *Gaps* médios em porcentagens de acordo com a estrutura populacional e mecanismos de seleção adotados (critério de parada: 300 segundos).

| | Conjuntos | | | | Média |
|---------------------------|-----------|------|------|------|-------------|
| | 5C | 10C | 20C | 30C | |
| População não estruturada | 1,31 | 3,63 | 3,74 | 3,31 | 3,00 |
| População estruturada | 0,68 | 2,31 | 3,51 | 2,79 | 2,32 |

Fonte: Autoria própria

Na Tabela 11, os *gaps* médios evidenciam o desempenho superior obtido pelo MA utilizando a população estruturada e, hierarquicamente, organizada em uma árvore ternária.

Figura 13 – Evolução da taxa de diversidade discreta em uma população ao longo das gerações (critério de parada: 300 gerações).



Fonte: Autoria própria

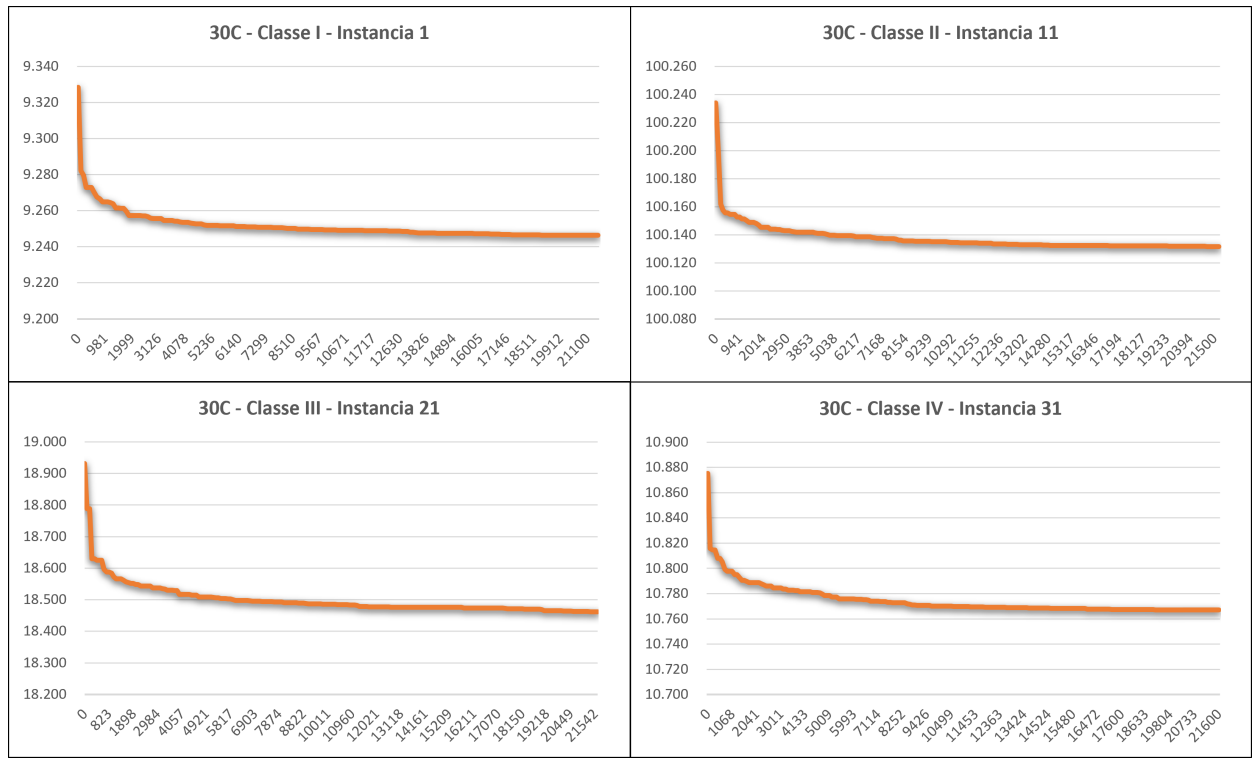
Como aconteceu em outras situações o desempenho mais próximo entre as abordagens ocorreu no grupo com 20 clientes.

Embora haja ainda um vasto espectro de possibilidades a serem exploradas a fim de se alcançar o máximo desempenho dos algoritmos evolutivos implementados, a partir desses testes preliminares foi estabelecer um ajuste mais preciso dos parâmetros garantindo assim a aplicação mais eficiente das técnicas desenvolvidas nesse trabalho. Além disso, é importante ressaltar que todos os testes executados para ajuste dos parâmetros utilizando as instâncias geradas e, descritos anteriormente, foram replicados para as instâncias de Archetti *et al.* (2011) com apenas um produto e resultados similares foram observados. Outro aspecto relevante que pôde ser observado ao longo do processo de ajuste dos parâmetros foi a inexistência de resultados muito discrepantes frente aos diversos cenários que foram explorados, evidenciando a robustez da técnica proposta neste trabalho.

5.3 Estratégia de decomposição

Com base no modelo de decomposição descrito na Seção 3.4, implementamos uma estratégia para resolver o problema de dimensionamento de lotes e o de roteamento

Figura 14 – Evolução da incumbente em instâncias com 30 clientes (critério de parada: 21600 segundos).



Fonte: Autoria própria

separadamente. Utilizamos o solver para obter uma solução inicial para o problema de produção. Como as quantidades de entregas ainda não foram definidas, atualizamos as demandas originais subtraindo os estoques iniciais da fábrica e dos clientes. Em seguida, esse plano de produção foi fixado para todos os indivíduos da população e o MA foi aplicado com o objetivo de obter uma solução para o problema de roteamento dos veículos. Importante salientar que apenas os operadores *crossOPR* e *crossOX1* foram utilizados nesta etapa por se tratar de operadores que não envolvem o plano de produção. A partir da solução obtida pelo MA para o roteamento, o solver foi executado novamente para otimizar o plano de produção considerando agora que o planejamento das entregas foi estabelecido. Ao combinar as soluções obtidas nestas etapas teremos uma solução completa para problema integrado.

Alguns experimentos foram realizados com o intuito de balancear os tempos de execução dedicados para cada uma das técnicas, adotando-se o tempo total como sendo de 300 segundos. A partir desses resultados, fixamos o tempo total do solver em 60 segundos e o do MA em 240 segundos.

Na Tabela 12, temos os *gaps* médios da estratégia de decomposição (Solver + MA)

Figura 15 – Evolução da incumbente em instâncias com 20 clientes (critério de parada: 21600 segundos).



Fonte: Autoria própria

comparados aos resultados obtidos pelo MA para o problema integrado. Podemos observar que o desempenho da estratégia integrada foi superior, especialmente nos grupos com 5, 10 e 30 clientes. As estratégias apresentaram desempenho mais próximo no grupo com 20 clientes, ainda que com boa vantagem para a estratégia integrada.

Tabela 12 – *Gaps* médios em porcentagens usando a estratégia de decomposição (critério de parada: 300 segundos).

| | Conjuntos | | | | Média |
|------------------|-----------|------|------|------|-------------|
| | 5C | 10C | 20C | 30C | |
| LSP+VRP | 1,72 | 3,72 | 3,90 | 3,21 | 3,14 |
| Integrado | 0,68 | 2,31 | 3,51 | 2,79 | 2,32 |

Fonte: Autoria própria

5.4 Comparação entre os algoritmos genético e memético

Alguns experimentos foram realizados com o objetivo de analisar o impacto da busca local no processo evolutivo, pois sem a incorporação dessas estratégias teríamos na

verdade um algoritmo genético. Além do algoritmo genético e do memético, avaliamos também o desempenho da busca local aplicada isoladamente para resolver o problema. Desse modo, temos na Tabela 13 a comparação entre as três estratégias para todas as instâncias com 5 clientes e limitando o tempo máximo de execução em 5 minutos (300 segundos). Nos demais grupos não foi possível comparar a busca local com as outras estratégias devido ao grande número de soluções ineficazes. A busca local aplicada de modo isolado apresentou o pior desempenho médio, exceto para a Classe I na qual foi ligeiramente superior ao GA. De maneira geral, estes resultados evidenciam o impacto da busca local, uma vez que o *gap* médio obtido pelo GA é praticamente o dobro do MA. Assim, ao analisarmos o desempenho em cada classe, temos que o resultado mais próximo entre as técnicas ocorre na classe IV, mas ainda assim com o GA alcançando resultados em média 40% piores.

Tabela 13 – Efeito da Busca Local: *Gaps* médios em porcentagens para as instâncias com 5 clientes (critério de parada: 300 segundos).

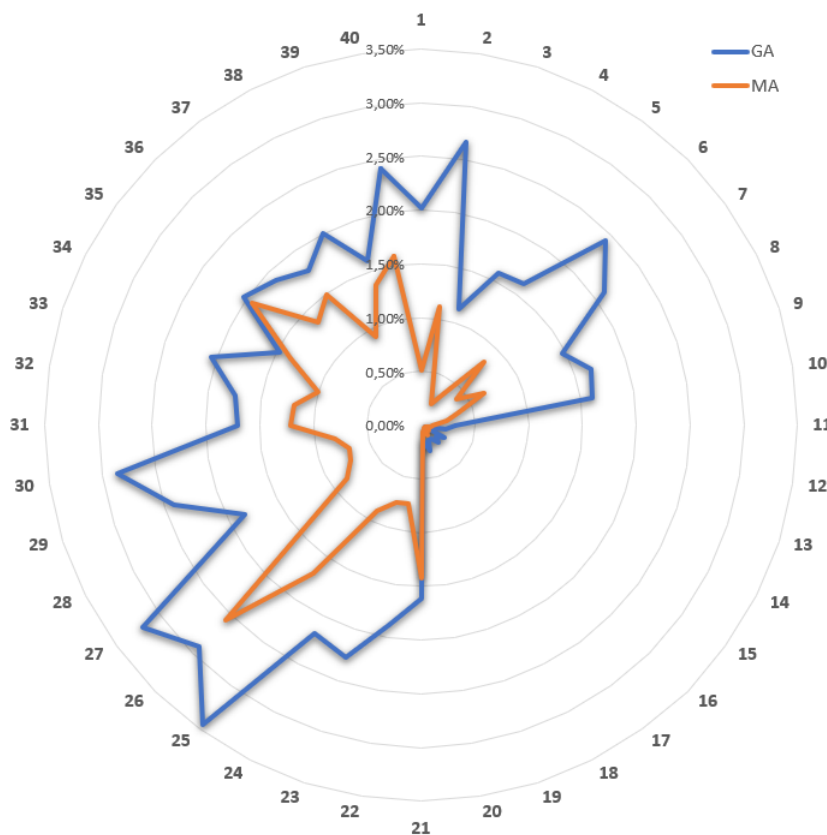
| | Classes | | | | Média |
|----|---------|------|------|------|-------------|
| | I | II | III | IV | |
| BL | 1,55 | 2,19 | 5,20 | 4,22 | 3,29 |
| GA | 1,83 | 0,20 | 2,46 | 1,87 | 1,59 |
| MA | 0,50 | 0,07 | 1,12 | 1,34 | 0,76 |

Fonte: Autoria própria

Na Figura 16 podemos observar o comportamento médio das 5 sementes por instância e, desta forma, é possível concluir que o desempenho mais próximo entre as técnicas ocorre em instâncias da Classe II (11 a 20) e na instância 35. Nas demais instâncias, contudo, a diferença de desempenho entre as técnicas mostra-se bem acentuada. É importante ressaltar que a Classe II possui as instâncias com altos custos de produção, o que reduz fortemente o impacto das melhorias obtidas no roteamento por meio da busca local, ou seja, o comportamento mais próximo entre os métodos nessas instâncias era esperado.

Para ilustrar graficamente o efeito causado pela busca local no método proposto, escolhemos, ao acaso, um indivíduo (filho) da população logo após a aplicação do operador de reprodução na primeira geração. Para isso, selecionamos a instância 1 do conjunto com 14 clientes proposto por Archetti *et al.* (2011) para facilitar a visualização das rotas envolvendo apenas um veículo e um produto. Após 10 iterações, o custo da solução inicial que era de 46907 foi reduzido para 43369, e essa melhoria pode ser facilmente observada nas Figuras 17 e 18, exibindo as rotas antes e depois da aplicação da busca local. De forma mais específica, podemos observar na Figura 18 que após a aplicação da busca local as rotas ficaram livres de interseções nos períodos 2, 3, 4 e 5.

Figura 16 – MA versus GA: *Gaps* médios em porcentagens para as instâncias com 5 clientes (critério de parada: 300 segundos).

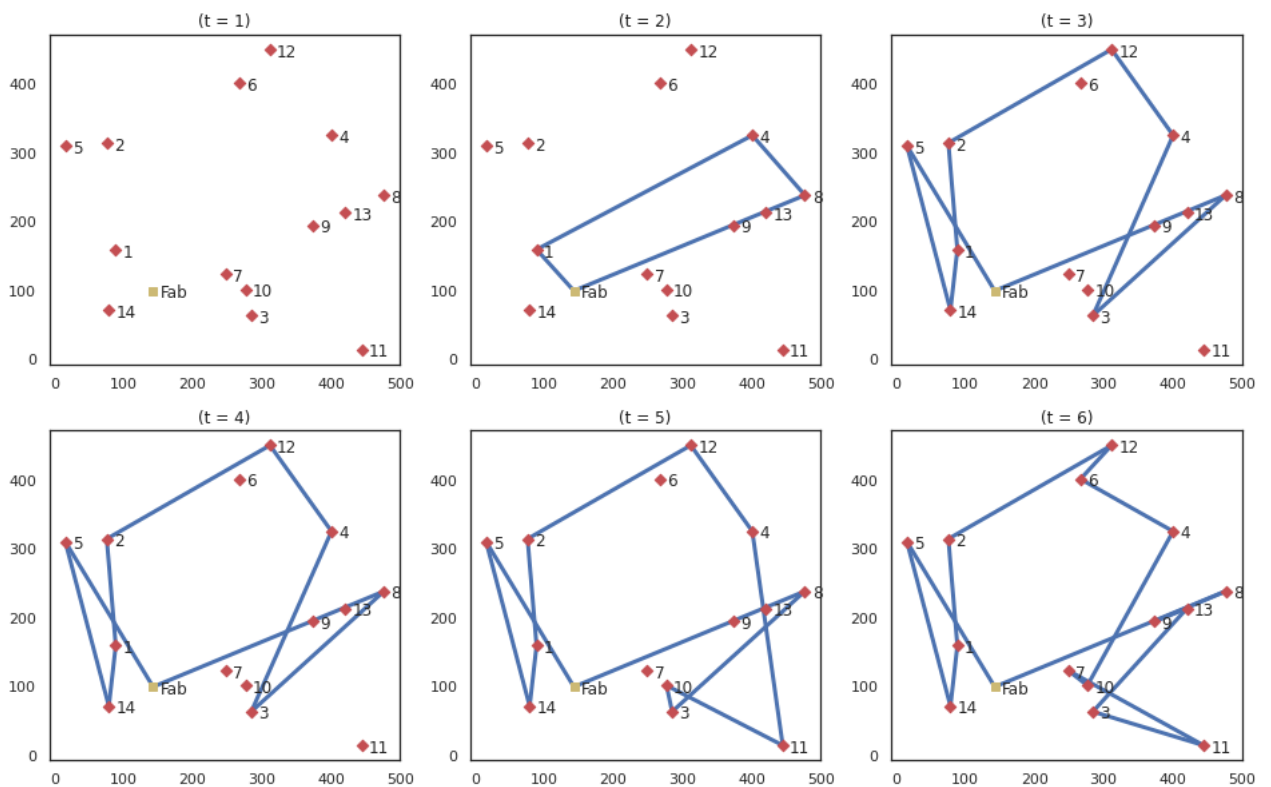


Fonte: Autoria própria

5.5 Resultados para as instâncias com múltiplos produtos

Na Tabela 14 apresentamos um resumo geral do desempenho do MA em comparação aos valores obtidos pelo *branch-and-cut* do solver Gurobi no tempo limite de 3600 segundos. Nessa tabela, para cada classe de instâncias, o *gap* médio em porcentagens representa a diferença entre o valor da função objetivo obtido pelo método e o limitante inferior obtido pelo solver, dividido pelo limitante inferior.

Para o grupo de instâncias com 5 clientes (5C), os dois métodos obtiveram excelentes resultados em todas as instâncias. Nas Classes I e III ambos encontraram as soluções ótimas em todas as instâncias, enquanto na Classe II o *gap* médio foi igual a zero tanto no solver quanto no MA, mas em duas sementes o *gap* médio foi igual a 0.01% para ambos. Na Classe IV, os dois métodos apresentaram o pior desempenho, mas ainda assim, os valores médios não foram superiores a 0.05%. Conforme a Tabela 16, o solver não determinou a solução ótima em apenas 3 instâncias da Classe IV, ao passo que o MA encontrou a solução ótima em 188 instâncias, o que equivale a 94% do total deste grupo. Em relação aos tempos computacionais exibidos na Tabela 15, o solver resolveu, em média, todas as

Figura 17 – Rotas de indivíduo **antes** da aplicação da Busca Local.

Fonte: Autoria própria

instâncias em 224 segundos. De forma mais específica, as duas primeiras classes foram resolvidas em menos de 15 segundos, enquanto que a Classe II levou em torno de 2 minutos e, na Classe IV, os tempos médios ficaram em torno de 12 minutos. Convém ressaltar que o MA exauriu o tempo limite estabelecido como critério de parada em todas as instâncias desse grupo.

No grupo com 10 clientes (10C), podemos observar que o MA obteve resultados muito similares ao solver em todas as classes, a diferença entre as duas técnicas foi de apenas 0.01% na média geral. Diferentemente do grupo anterior, a classe que apresentou pior *gap* médio foi a Classe III com média em torno de 3% e o melhor desempenho foi obtido na Classe II, com altos custos de produção, atingindo o *gap* médio de 0.10%. As Classes I e IV também apresentaram resultados muito bons, com os *gaps* médios inferiores a 1%, assim, de maneira geral, os resultados médios ficaram próximos das soluções ótimas nesse grupo. Com relação aos tempos computacionais destacamos que as técnicas também se aproximaram bastante, pois o Gurobi determinou apenas uma solução ótima da Classe I, porém, nas demais, utilizou todo o tempo de processamento disponível.

Como podemos ver na Tabela 14, para os grupos de instâncias maiores, com 20 e

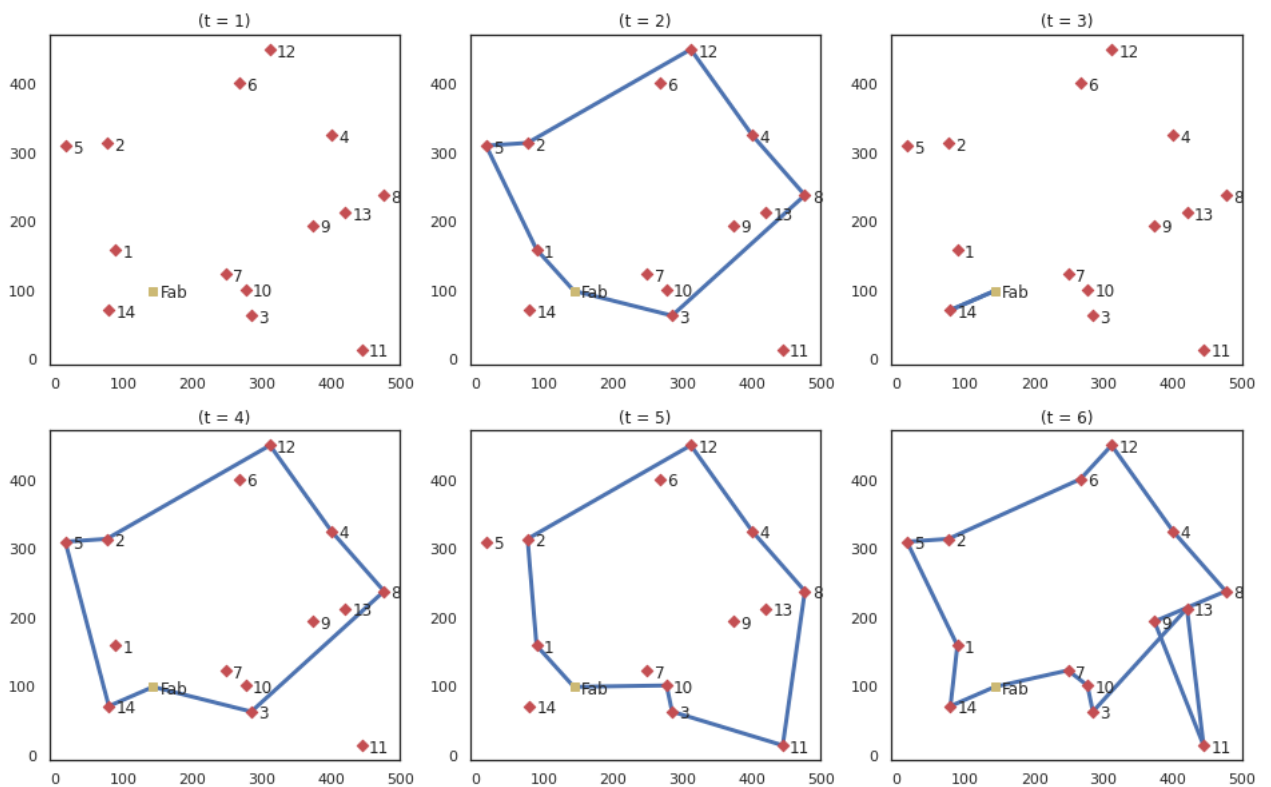
Tabela 14 – *Gaps* médios em percentagens para as instâncias com múltiplos produtos.

| Conjuntos | Métodos | Classes | Sementes | | | | | Média |
|-----------|---------|---------|-------------|-------------|-------------|-------------|-------------|-------------|
| | | | 1 | 2 | 3 | 4 | 5 | |
| 5C | Gurobi | I | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | | II | 0,00 | 0,00 | 0,00 | 0,01 | 0,01 | 0,00 |
| | | III | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | | IV | 0,00 | 0,05 | 0,03 | 0,03 | 0,00 | 0,02 |
| | | Média | 0,00 | 0,01 | 0,01 | 0,01 | 0,00 | 0,01 |
| | MA | I | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | | II | 0,00 | 0,00 | 0,00 | 0,01 | 0,01 | 0,00 |
| | | III | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | | IV | 0,00 | 0,05 | 0,05 | 0,05 | 0,01 | 0,03 |
| | | Média | 0,00 | 0,01 | 0,01 | 0,01 | 0,00 | 0,01 |
| 10C | Gurobi | I | 0,72 | 0,77 | 0,89 | 0,59 | 0,78 | 0,75 |
| | | II | 0,09 | 0,10 | 0,09 | 0,09 | 0,10 | 0,09 |
| | | III | 2,83 | 3,25 | 2,87 | 2,55 | 3,55 | 3,01 |
| | | IV | 0,94 | 1,04 | 0,76 | 0,87 | 1,19 | 0,96 |
| | | Média | 1,15 | 1,29 | 1,15 | 1,03 | 1,40 | 1,20 |
| | MA | I | 0,72 | 0,77 | 0,89 | 0,59 | 0,78 | 0,75 |
| | | II | 0,09 | 0,11 | 0,09 | 0,10 | 0,10 | 0,10 |
| | | III | 2,84 | 3,31 | 2,89 | 2,57 | 3,54 | 3,03 |
| | | IV | 0,97 | 1,07 | 0,80 | 0,89 | 1,15 | 0,98 |
| | | Média | 1,15 | 1,31 | 1,17 | 1,04 | 1,39 | 1,21 |
| 20C | Gurobi* | | | | | | | |
| | MA | I | 3,05 | 2,73 | 2,69 | 2,47 | 2,87 | 2,76 |
| | | II | 0,32 | 0,37 | 0,31 | 0,34 | 0,33 | 0,33 |
| | | III | 5,33 | 5,48 | 4,91 | 6,02 | 5,33 | 5,42 |
| | | IV | 3,26 | 3,29 | 3,57 | 3,74 | 3,25 | 3,42 |
| | | Média | 2,99 | 2,97 | 2,87 | 3,14 | 2,95 | 2,98 |
| 30C | Gurobi* | | | | | | | |
| | MA | I | 1,92 | 2,05 | 2,16 | 1,80 | 2,33 | 2,05 |
| | | II | 0,22 | 0,27 | 0,23 | 0,22 | 0,24 | 0,23 |
| | | III | 5,88 | 5,79 | 6,02 | 6,03 | 5,30 | 5,80 |
| | | IV | 2,69 | 2,56 | 2,54 | 2,67 | 2,63 | 2,62 |
| | | Média | 2,68 | 2,67 | 2,74 | 2,68 | 2,62 | 2,68 |

* Não encontrou nenhuma solução factível após 1 h de execução.

Fonte: Autoria própria

Figura 18 – Rotas de indivíduo **depois** da aplicação da Busca Local.



Fonte: Autoria própria

30 clientes, o desempenho do MA foi bastante superior, enquanto o Gurobi não encontrou nenhuma solução factível dentro do limite de tempo estabelecido. Por outro lado, o MA obteve soluções factíveis com excelentes *gaps* médios, especialmente na Classe II que apresentou valores médios inferiores 0.35%. No grupo 20C, as Classes I e IV tiveram resultados próximos de 3% e 3.5%, respectivamente, ao passo que a Classe III ficou em torno de 5%. De maneira geral, em relação às classes, o desempenho do MA foi similar aos dois grupos de instâncias menores. Os resultados do grupo 30C foram ainda melhores o que evidencia a robustez do método proposto. Os valores obtidos para a Classe II indicam que as soluções determinadas pelo MA ficaram muito próximas das soluções ótimas com os *gaps* inferiores a 0.25%. Nas Classes I e IV, os *gaps* ficaram em torno de 2% e 2.7%, respectivamente, por sua vez, na Classe III o *gap* médio foi de 5.8% enquanto a média geral foi de 2.7%. Nenhuma solução ótima foi obtida pelo MA dentro do limite de tempo fixado.

5.5.1 Estratégia híbrida usando o solver

Diante da dificuldade do solver em obter soluções até mesmo factíveis para as instâncias com 20 e 30 clientes, decidimos explorar com maior profundidade sua capacidade

Tabela 15 – Tempos de execução médios (em segundos) para as instâncias com múltiplos produtos.

| Conjuntos | Métodos | Classes | | | | Média |
|-----------|---------|---------|-------|-------|-------|--------------|
| | | I | II | III | IV | |
| 5C | Gurobi | 4 | 13 | 122 | 756 | 224 |
| | MA | 3.600 | 3.600 | 3.600 | 3.600 | 3.600 |
| 10C | Gurobi | 3.555 | 3.600 | 3.600 | 3.600 | 3.589 |
| | MA | 3.600 | 3.600 | 3.600 | 3.600 | 3.600 |
| 20C | Gurobi | 3.600 | 3.600 | 3.600 | 3.600 | 3.600 |
| | MA | 3.600 | 3.600 | 3.600 | 3.600 | 3.600 |
| 30C | Gurobi | 3.600 | 3.600 | 3.600 | 3.600 | 3.600 |
| | MA | 3.600 | 3.600 | 3.600 | 3.600 | 3.600 |

Fonte: Autoria própria

Tabela 16 – Número de soluções ótimas obtidas nas instâncias com múltiplos produtos.

| Conjuntos | Métodos | Classes | | | | Total |
|-----------|---------|---------|----|-----|----|------------|
| | | I | II | III | IV | |
| 5C | Gurobi | 50 | 50 | 50 | 47 | 197 |
| | MA | 50 | 48 | 49 | 41 | 188 |
| 10C | Gurobi | 1 | - | - | - | 1 |
| | MA | - | - | - | - | - |
| 20C | Gurobi | - | - | - | - | - |
| | MA | - | - | - | - | - |
| 30C | Gurobi | - | - | - | - | - |
| | MA | - | - | - | - | - |

Fonte: Autoria própria

para melhorar uma solução inicial obtida pelo MA.

Na Tabela 17, apresentamos os resultados desses experimentos utilizando o mesmo critério de parada dos testes anteriores, ou seja, o tempo total de execução foi de 1 hora. Fixado o tempo total, alguns experimentos para obter o balanceamento ideal entre os tempos de cada uma das técnicas foram realizados, especialmente considerando o desempenho inversamente proporcional das mesmas a medida que as instâncias aumentam de tamanho. A partir desse resultados, o MA foi executado durante 5 minutos e o solver 55 minutos, ou seja, ao final dos primeiros 5 minutos a melhor solução obtida pelo MA foi fornecida como solução inicial para o solver que, em seguida, foi executado por mais 55 minutos.

Como podemos observar, exceto para o grupo de instâncias com 10 clientes em que

Tabela 17 – *Gaps* médios em porcentagens das estratégias híbridas.

| Conjuntos | Métodos | Classes | | | | Média |
|-----------|---------|---------|------|------|------|-------------|
| | | I | II | III | IV | |
| 10C | Gurobi | 0,75 | 0,09 | 3,01 | 0,96 | 1,20 |
| | MA+GRB | 0,75 | 0,10 | 3,02 | 0,97 | 1,21 |
| | MA | 0,75 | 0,10 | 3,03 | 0,98 | 1,21 |
| 20C | Gurobi | - | - | - | - | - |
| | MA+GRB | 2,21 | 0,25 | 4,48 | 2,08 | 2,25 |
| | MA | 2,76 | 0,33 | 5,42 | 3,42 | 2,98 |
| 30C | Gurobi | - | - | - | - | - |
| | MA+GRB | 2,08 | 0,23 | 5,52 | 1,95 | 2,44 |
| | MA | 2,05 | 0,23 | 5,80 | 2,62 | 2,68 |

Fonte: Autoria própria

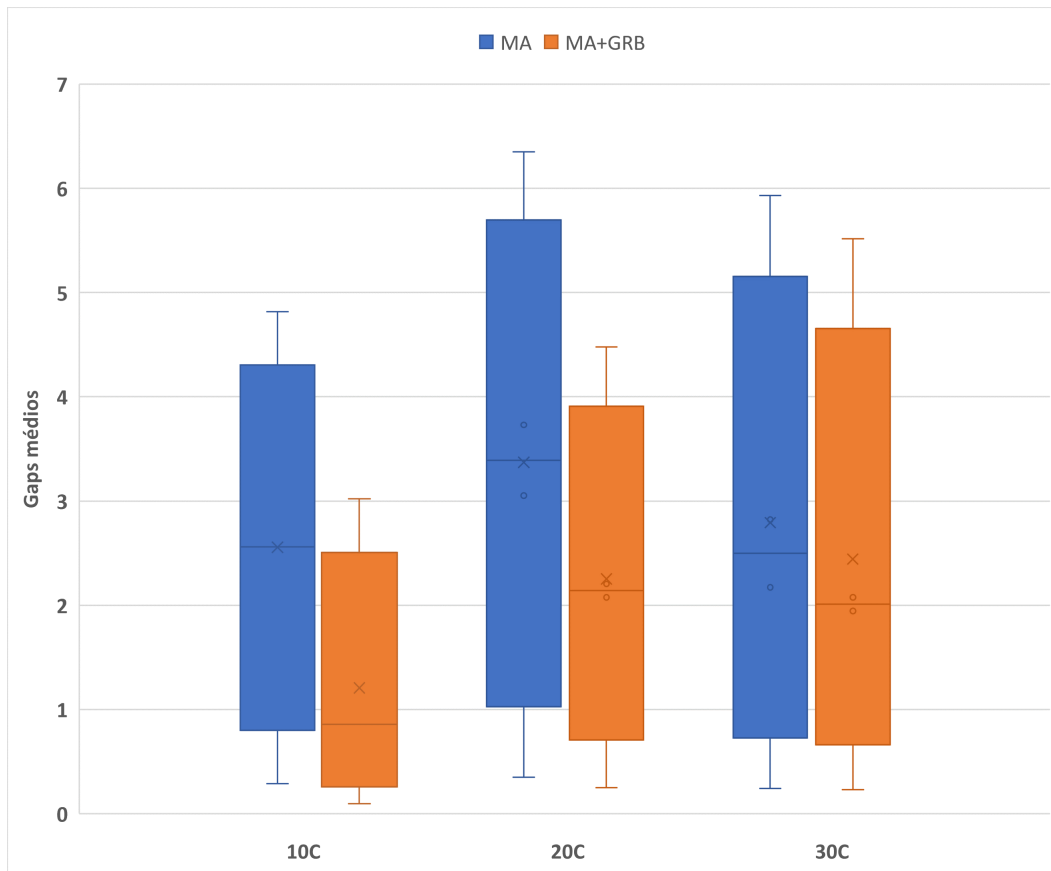
as três estratégias alcançaram praticamente os mesmos resultados, nos demais grupos com 20 e 30 clientes a estratégia híbrida conseguiu atingir melhores soluções do que o MA e o solver isoladamente aplicados. Em média, conseguiu melhorar em 0.74 pontos percentuais as soluções do grupo com 20 clientes e em 0.26 pontos percentuais, em média, as soluções do grupo com 30 clientes. Analisando os resultados por classes, observamos que no grupo com 20 clientes a estratégia híbrida conseguiu melhorar os resultados para todas as classes, enquanto que, no grupo com 30 clientes, essa melhora ocorreu apenas nas classes III e IV.

Na Figura 19, apresentamos uma comparação entre a solução inicial obtida pelo MA após 5 minutos de execução e o resultado final após 55 minutos de execução do Gurobi. Pelos *boxplots* de cada grupo de instâncias, podemos comprovar que quanto maiores as instâncias maiores foram as dificuldades encontradas pelo solver para melhorar as soluções obtidas pelo MA. Ainda assim, essa estratégia híbrida mostrou-se vantajosa ao obter melhores resultados do que os alcançados pelo MA ou pelo Gurobi isoladamente. Esses resultados também evidenciam que um caminho promissor poderia ser a combinação de métodos exatos com a meta-heurística proposta nesse trabalho.

5.6 Descrição das instâncias com apenas um produto

Para efeito de comparação dos métodos desenvolvidos neste trabalho com outras técnicas da literatura, vamos utilizar os dois grupos de instâncias propostas por Archetti *et al.* (2011) com 14 e 50 clientes, conforme a Tabela 18. O grupo A1 tem apenas um veículo enquanto o grupo A2 tem um número ilimitado de veículos. Além disso, as demandas são constantes e os estoques iniciais dos clientes são variáveis ao passo que a capacidade de estoque dos clientes é limitada. As capacidades de produção e de estoque da planta são ilimitadas, mas os estoques iniciais são nulos. Além disso, as instâncias estão divididas em

Figura 19 – Melhoria obtida pela estratégia híbrida (*gaps* médios em porcentagens).



Fonte: Autoria própria

quatro classes, de acordo com a Tabela 19. A Classe I (1 a 24) tem a configuração básica de custos, a Classe II (25 a 48) tem altos custos de produção à medida que a Classe III (49 a 72) tem altos custos de transportes e, por fim, a Classe IV (73 a 96) não tem custos de estoques dos clientes. Desse modo, cada classe tem 24 instâncias com 5 sementes cada e, assim, temos 480 instâncias para cada um dos grupos, totalizando 960 instâncias nesse conjunto.

5.7 Resultados para as instâncias com apenas um produto

Embora todos os métodos implementados nesse trabalho tenham como foco o PRP com múltiplos produtos, optamos por comparar a performance do MA com outras técnicas, utilizando as instâncias clássicas do PRP com um único item.

Diante do grande número de instâncias no conjunto proposto por Archetti *et al.* (2011) e no sentido de viabilizar as comparações com outros trabalhos publicados mais recentemente, selecionamos apenas os conjuntos com 14 e 50 clientes para comparação nesse estudo, conforme a Tabela 18.

Tabela 18 – Visão geral das instâncias de Archetti *et al.* (2011).

| instâncias | A1 | A2 |
|------------------------------------|-----------|-----------|
| Número de instâncias | 480 | 480 |
| Número de períodos | 6 | 6 |
| Número de clientes | 14 | 50 |
| Número de veículos | 1 | Ilimitado |
| Demanda | Constante | Constante |
| Capacidade de produção | Ilimitado | Ilimitado |
| Capacidade de estoque da fábrica | Ilimitado | Ilimitado |
| Capacidade de estoque dos clientes | Constante | Constante |
| Estoque inicial da fábrica | 0 | 0 |
| Estoque inicial dos clientes | Variável | Variável |
| Capacidade dos veículos | Constante | Constante |

Fonte: Adaptado de Adulyasak, Cordeau e Jans (2014)

Tabela 19 – Descrição das classes de Archetti *et al.* (2011).

| Classe | Tipo | Descrições |
|---------------|-------------|--|
| Classe I | 1-24 | instâncias padrão |
| Classe II | 25-48 | Custos de produção elevados (Classe I x 10) |
| Classe III | 49-72 | Custos de transporte elevados (Classe I x 5) |
| Classe IV | 73-96 | Custos de estoques dos clientes zerados |

Fonte: Adaptado de Adulyasak, Cordeau e Jans (2014)

Tabela 20 – Estado-da-arte das técnicas para o PRP clássico.

| Referência | Abreviação | Técnica | Solver | Conjunto |
|--------------------------------------|-------------------|----------------------------|---------------|-----------------|
| Adulyasak, Cordeau e Jans (2014) | ACJ-ALNS | MIP-based ALNS | CPLEX 12.2 | A |
| Absi <i>et al.</i> (2015) | AADF-IM | Iterative MIP heuristic | CPLEX 12.1 | A2, A3 |
| Solyali e Süral (2017) | SS-M | MIP-based heuristic | CPLEX 12.5 | A |
| Qiu <i>et al.</i> (2018b) | QWXFP-VNS | MIP-based VNS | CPLEX 12.6 | A |
| Chitsaz, Cordeau e Jans (2019) | CCJ-M | Decomposition Matheuristic | CPLEX 12.6 | A |
| Li <i>et al.</i> (2019) | LCCZ-M | MIP-based heuristic | CPLEX 12.6 | A |
| Avci e Yildiz (2019) | AY-M | MIP-based Local Search | CPLEX 12.6 | A |
| Manousakis <i>et al.</i> (2022) | MKKZ-M | MIP-based Local Search | Gurobi 9.0.2 | A |
| Vadseth, Andersson e Stålhane (2022) | VACS-M | Matheuristic | Gurobi 9.1 | A2, A3 |
| Esta tese | RSAL-MA | Memetic Algorithm | Gurobi 9.1.2 | A1, A2 |

Fonte: Autoria própria

O $gap(\%)$ para cada instância é calculado como sendo a diferença entre a solução obtida pelo método e a melhor solução conhecida até o momento, dividida pela melhor solução. Para determinar a melhor solução conhecida em cada instância, desconsiderando as soluções obtidas neste trabalho, utilizamos o material suplementar disponibilizado pelos autores cujos trabalhos estão listados na Tabela 20. Convém salientar que utilizamos os trabalhos mais relevantes da última década, com destaque para os recentes trabalhos de Manousakis *et al.* (2022) e Vadseth, Andersson e Stålhane (2022) que apresentaram

resultados excelentes para esse conjunto de instâncias. Outro destaque importante acerca desses trabalhos é o fato de todos utilizarem métodos exatos combinados com os heurísticos.

Podemos observar na Tabela 21 que o RSAL-MA atingiu resultados muito bons para o grupo com 14 clientes (A1), em comparação com os demais métodos, tendo em vista que o método proposto nesse trabalho apresentou melhores resultados do que os métodos ACJ-ALNS, QWAFP-VNS e CCJ-M. Para esse grupo, AY-M atingiu os melhores resultados em todas as classes. Vale ressaltar que, embora Absi *et al.* (2015) (AADF-IM) tenham apresentados os *gaps* obtidos para esse conjunto, os autores não divulgaram os custos, o que inviabilizou a comparação com os demais métodos. Por outro lado, Vadseth, Andersson e Stålhane (2022) (VACS-M) optaram por não utilizar esse conjunto de instâncias. Em relação às instâncias do conjunto A2, com 50 clientes, o MA não superou nenhum dos trabalhos anteriores. Nesse grupo, o método MKKZ-M apresentou melhores resultados para as Classes I, II e III, enquanto VACS-M foi superior na Classe IV.

Na Tabela 22 exibimos os tempos computacionais de todos os métodos. Convém destacar que Vadseth, Andersson e Stålhane (2022) calcularam uma estimativa para os tempos do método MKKZ-M, pois os autores não divulgaram o tempo total de execução da técnica no trabalho original. Utilizamos, no MA, 300 segundos como critério de parada nos dois conjuntos de instâncias. Logo, podemos observar que o método SS-M apresentou os melhores tempos computacionais seguido do método QWAFP-VNS, com tempos em torno de 20 segundos nos dois conjuntos de instâncias.

A seguir apresentamos as conclusões do trabalho e algumas perspectivas para pesquisas futuras.

Tabela 21 – *Gaps* médios em porcentagens das melhores soluções conhecidas para o conjunto A.

| Conjuntos | Classes | ACJ-ALNS | AADF-IM | SS-M | QWXP-P-VNS | CCJ-M | LCCZ-M | AY-M | MKKZ-M | VACS-M | RSAL-MA |
|-----------|---------|----------|---------|-------------|------------|-------|--------|-------------|-------------|-------------|---------|
| A1 | I | 1,70 | - | 0,03 | 0,28 | 0,47 | 0,15 | 0,01 | 0,05 | - | 0,41 |
| | II | 0,37 | - | 0,01 | 0,05 | 0,08 | 0,03 | 0,01 | 0,01 | - | 0,06 |
| | III | 8,42 | - | 0,18 | 1,52 | 2,20 | 0,75 | 0,04 | 0,38 | - | 1,50 |
| | IV | 0,93 | - | 0,03 | 0,56 | 0,23 | 0,08 | 0,01 | 0,00 | - | 0,28 |
| | Média | 2,85 | - | 0,06 | 0,60 | 0,75 | 0,25 | 0,02 | 0,11 | - | 0,56 |
| A2 | I | 1,37 | 0,37 | 0,29 | 0,28 | 0,32 | 0,23 | 0,20 | 0,04 | 0,06 | 3,51 |
| | II | 0,22 | 0,08 | 0,06 | 0,08 | 0,08 | 0,06 | 0,06 | 0,00 | 0,02 | 0,33 |
| | III | 4,47 | 1,64 | 1,00 | 0,95 | 1,24 | 0,90 | 0,86 | 0,14 | 0,23 | 10,90 |
| | IV | 0,39 | 0,25 | 0,26 | 0,18 | 0,19 | 0,22 | 0,21 | 0,10 | 0,03 | 3,24 |
| | Média | 1,61 | 0,59 | 0,40 | 0,37 | 0,46 | 0,35 | 0,33 | 0,07 | 0,09 | 4,50 |

Fonte: Autoria própria

Tabela 22 – Tempos médios de execução em segundos para o conjunto A. Nota: * = tempo estimado (Vadseth, Andersson e Stålhane (2022)).

| Conjuntos | Classes | ACJ-ALNS | AADF-IM | SS-M | QWXP-P-VNS | CCJ-M | LCCZ-M | AY-M | MKKZ-M* | VACS-M | RSAL-MA |
|-----------|---------|----------|---------|------|------------|-------|--------|------|---------|--------|---------|
| A1 | I | 9 | - | 5 | 12 | 18 | 29 | 12 | 86 | - | 300 |
| | II | 9 | - | 5 | 13 | 18 | 28 | 10 | 86 | - | 300 |
| | III | 9 | - | 5 | 12 | 17 | 28 | 9 | 105 | - | 300 |
| | IV | 9 | - | 5 | 13 | 18 | 25 | 9 | 98 | - | 300 |
| | Média | 9 | - | 5 | 12 | 18 | 28 | 10 | 94 | - | 300 |
| A2 | I | 50 | 339 | 16 | 25 | 400 | 43 | 89 | 1513 | 130 | 300 |
| | II | 50 | 236 | 14 | 21 | 344 | 37 | 78 | 1563 | 103 | 300 |
| | III | 43 | 318 | 16 | 25 | 309 | 39 | 78 | 1832 | 156 | 300 |
| | IV | 44 | 376 | 25 | 28 | 434 | 42 | 84 | 1409 | 142 | 300 |
| | Média | 47 | 317 | 18 | 25 | 372 | 40 | 82 | 1579 | 133 | 300 |

Fonte: Autoria própria

6 CONCLUSÕES E PERSPECTIVAS FUTURAS

Esta pesquisa teve como propósito principal desenvolver algoritmos evolutivos para o problema de roteamento e produção com múltiplos produtos, utilizando uma frota homogênea de veículos. O objetivo do problema é minimizar os custos de produção, de estoques na fábrica e nos clientes e os custos de roteamento dos veículos, de modo a atender a demanda dos clientes em um horizonte finito de planejamento. A produção de múltiplos itens ocorre em uma fábrica com capacidade de produção limitada e a distribuição dos produtos aos clientes é realizada por uma frota homogênea e limitada de veículos. Os produtos podem ser armazenados na fábrica e nos clientes com diferentes custos de estoques e o gerenciamento dos estoques é feito pela fábrica.

A partir de uma ampla pesquisa bibliográfica inicial, foi possível vislumbrar algumas lacunas na literatura do PRP que acabaram se tornando o principal foco de desenvolvimento desse trabalho. No Capítulo 2, iniciamos a revisão desses trabalhos com as principais publicações envolvendo o PRP com múltiplos produtos. Em seguida, descrevemos os artigos com os resultados mais relevantes da última década para o problema com apenas um produto, encerrando com os trabalhos que utilizaram os algoritmos evolutivos no contexto do PRP.

No Capítulo 3, definimos as características e o modelo matemático para o PRP com múltiplos produtos e frota homogênea, que foi o tema central desta pesquisa. A partir desse modelo, deduzimos os dois subseqüentes que seriam utilizados para o PRP com apenas um produto e para o problema com frotas heterogêneas e múltiplos produtos. Por fim, apresentamos um procedimento heurístico de decomposição para separar os problemas de dimensionamento de lotes e de roteamento dos veículos. Nesse procedimento, acionamos o solver para obter um plano de produção em uma primeira etapa. Em seguida, com o plano de produção fixado o algoritmo evolutivo determina uma solução para problema de roteamento dos veículos. Ao final desse processo, com as entregas definidas, o plano de produção passa por um novo processo de otimização pelo solver. Portanto, com as soluções combinadas, ao final do procedimento, teremos uma solução completa para o problema integrado.

Para resolver o problema adotamos os algoritmos genéticos e meméticos que são técnicas populacionais ainda pouco exploradas no contexto do PRP com múltiplos produtos. Para gerar a quantidade de soluções necessárias para compor a população inicial, modificamos algumas estratégias clássicas de decomposição que resolvem os problemas de dimensionamento de lotes e de roteamento separadamente. Aliado a isso, implementamos quatro novos operadores genéticos e adaptamos um outro operador clássico do TSP. Além disso, desenvolvemos diversos procedimentos de factibilização e adaptamos diversas heurís-

ticas que foram combinadas e utilizadas na busca local dos algoritmos meméticos. Para evitar a estagnação da busca, com a convergência prematura da população, implementamos algumas estratégias de diversificação e reinícios da população sempre que as taxas de diversidade atingiam níveis críticos.

Com o objetivo de avaliar os métodos propostos e considerando a ausência de um conjunto de instâncias que pudesse ser usado como referência para o problema, geramos um novo conjunto aleatoriamente (Seção 5.1), tendo por base alguns trabalhos predecessores. Essas instâncias serão disponibilizadas com o intuito de facilitar futuras análises e comparações com outras técnicas.

Os algoritmos evolutivos são técnicas populacionais robustas que envolvem o ajuste de muitos parâmetros e sua aplicação neste contexto tornou-se um grande desafio. A esse respeito, numerosos experimentos foram especialmente projetados para avaliar e otimizar o desempenho das inúmeras combinações destes parâmetros. Os principais resultados destes testes foram descritos na Seção 5.2. Estes experimentos preliminares possibilitaram o ajuste mais preciso dos algoritmos evolutivos e, conseqüentemente, impactaram positivamente o desempenho global dos métodos.

Na Seção 5.3, foram apresentados os resultados de uma estratégia de decomposição usada para resolver o problema de dimensionamento de lotes e o de roteamento separadamente. O solver foi utilizado para obter uma solução do problema de produção, enquanto o MA foi aplicado para obter uma solução do problema de roteamento dos veículos. As soluções obtidas nestas etapas foram combinadas para obter uma solução completa do problema integrado. Importante observar que, apenas as heurísticas e os operadores genéticos, envolvendo o roteamento de veículos foram usados nesta estratégia.

Na Seção 5.4, ao comparar o desempenho dos algoritmos genéticos e meméticos, por meio de alguns experimentos, evidenciamos o impacto da busca local presente apenas nos algoritmos meméticos. Além disso, avaliamos o desempenho da busca local aplicada isoladamente para resolver o problema. Com base nesses resultados, foi possível observar o desempenho superior obtido pelo algoritmo memético.

Os experimentos executados para as instâncias com múltiplos produtos foram apresentados na Seção 5.5. Assim, foi possível observar que o algoritmo memético superou o solver, encontrando soluções factíveis de alta qualidade, principalmente nas instâncias maiores do problema. O MA proposto atingiu excelentes resultados quando comparados aos do solver nos conjuntos com 5 e 10 clientes. Além disso, a comparação com os limitantes inferiores, obtidos pelo solver nos grupos com 20 e 30 clientes, aponta *gaps* médios bastante reduzidos, o que ressalta a qualidade das soluções obtidas para estas instâncias.

Tendo em vista a dificuldade do solver em obter soluções factíveis para as instâncias com 20 e 30 clientes, realizamos alguns experimentos para explorar sua capacidade em

melhorar uma solução inicial obtida pelo MA. A partir dessas análises, apresentadas na Subseção 5.5.1, foi possível observar que nos grupos de instâncias com 20 e 30 clientes essa estratégia conseguiu atingir melhores soluções do que o MA e o solver quando são isoladamente aplicados.

Finalmente, na Seção 5.7, comparamos o desempenho da técnica com o estado da arte envolvendo um conjunto amplamente utilizado de instâncias do PRP com um único item. Embora todo o desenvolvimento tenha tido como foco o problema com múltiplos itens, as técnicas propostas apresentaram bom desempenho quando comparadas ao estado da arte envolvendo o problema com apenas um item.

Portanto, os experimentos evidenciam a robustez da técnica e apontam perspectivas promissoras à sua aplicação em problemas reais que normalmente envolvem dimensões ainda maiores. A partir destas perspectivas, alguns ajustes, visando otimizar seu desempenho computacional, se fazem necessários para ampliar sua eficácia. Uma alternativa que vale a pena ser explorada trata-se da incorporação de outras meta-heurísticas ou métodos exatos aos algoritmos evolutivos desenvolvidos ao longo desse trabalho. Importante observar que essas abordagens mistas, denominadas de matheurísticas, apresentaram excelentes resultados para o problema com um único item. Neste sentido, acreditamos que seria necessário utilizar outros modelos matemáticos mais ajustados as necessidades desta categoria de métodos. Por outro lado, métodos não populacionais poderiam fazer uso dos operadores e heurísticas que foram desenvolvidas, mas com a vantagem de serem mais leves e eficientes do ponto de vista computacional. Por fim, características adicionais poderiam ser incorporadas ao problema tornando-o mais realista, tais como, sequenciamento da produção, entregas parciais, múltiplos estágios de produção, janelas de tempo, dentre outras.

Ao final deste trabalho, acreditamos ter contribuído com as pesquisas desta área ao abordar uma técnica, ainda pouco explorada, no contexto do PRP dada sua relevância histórica na área de otimização, que mostrou-se eficiente e robusta nos experimentos computacionais realizados neste trabalho.

REFERÊNCIAS

- ABSI, N. *et al.* A Two-Phase Iterative Heuristic Approach for the Production Routing Problem. **Transportation Science**, Elsevier B.V., v. 49, p. 721–1005, 2015. ISSN 18770428.
- ABSI, N. *et al.* Comparing sequential and integrated approaches for the production routing problem. **European Journal of Operational Research**, v. 269, n. 2, p. 633–646, 2018.
- ADULYASAK, Y.; CORDEAU, J.-F.; JANS, R. Optimization-Based Adaptive Large Neighborhood Search for the Production Routing Problem. **Transportation Science**, v. 48, n. 1, p. 20–45, 2014.
- ADULYASAK, Y.; CORDEAU, J. F.; JANS, R. The production routing problem: A review of formulations and solution algorithms. **Computers and Operations Research**, v. 55, p. 141–152, 2015.
- ALVARENGA, J. C.; OLIVEIRA, F. B.; George H. G. Fonseca. Aplicação de técnicas meta-heurísticas em um problema real de otimização de rotas de entregas de um supermercado. *In: Anais do LII Simpósio Brasileiro de Pesquisa Operacional*. João Pessoa, PB: [S.l.: s.n.], 2020.
- ARCHETTI, C. *et al.* Analysis of the maximum level policy in a production-distribution system. **Computers and Operations Research**, Elsevier, v. 38, n. 12, p. 1731–46, 2011.
- ARCHETTI, C.; SPERANZA, M. G. The inventory routing problem: The value of integration. **International Transactions in Operational Research**, v. 23, n. 3, p. 393–407, 2016. ISSN 14753995.
- ARMENTANO, V. A.; SHIGUEMOTO, A. L.; LØKKETANGEN, A. Tabu search with path relinking for an integrated productiondistribution problem. **Computers and Operations Research**, Elsevier, v. 38, n. 8, p. 1199–1209, 2011.
- AVCI, M.; YILDIZ, S. T. A matheuristic solution approach for the production routing problem with visit spacing policy. **European Journal of Operational Research**, Elsevier B.V., v. 279, n. 2, p. 572–588, 2019. ISSN 03772217.
- BERRETTA, R.; MOSCATO, P. The Number Partitioning Problem: An Open Challenge for Evolutionary Computational? *In: Corne, D.; Dorigo, M.; Glover, F.; Dasgupta, D.; Moscato, P.; Poli, R.; Price, K. V. (ed.). New Ideas in Optimisation*. [S.l.: s.n.]: McGraw-Hill, 1999. cap. 14, p. 261–278.
- BERRETTA, R.; RODRIGUES, L. F. A memetic algorithm for a multistage capacitated lot-sizing problem. **International Journal of Production Economics**, v. 87, n. 1, p. 67–81, 2004. ISSN 09255273.
- BOUDIA, M.; LOULY, M. A.; PRINS, C. A reactive GRASP and path relinking for a combined production-distribution problem. **Computers and Operations Research**, v. 34, n. 11, p. 3402–3419, 2007. ISSN 03050548.

BOUDIA, M.; PRINS, C. A memetic algorithm with dynamic population management for an integrated production-distribution problem. **European Journal of Operational Research**, Elsevier B.V., v. 195, n. 3, p. 703–715, 2009.

BRAHIMI, N.; AOUAM, T. Multi-item production routing problem with backordering: A MILP approach. **International Journal of Production Research**, Taylor & Francis, v. 54, n. 4, p. 1076–1093, 2016. ISSN 1366588X.

CHANDRA, P.; FISHER, M. L. Coordination of Production and Distribution Planning. **European Journal of Operational Research**, v. 72, p. 503–517, 1994. ISSN 1943-2631.

CHITSAZ, M.; CORDEAU, J. F.; JANS, R. A unified decomposition matheuristic for assembly, production, and inventory routing. **INFORMS Journal on Computing**, v. 31, n. 1, p. 134–152, 2019. ISSN 15265528.

CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations Research**, Institute for Operations Research and the Management Sciences (INFORMS), v. 12, p. 568–581, 8 1964.

CORDEAU, J. F. *et al.* A guide to vehicle routing heuristics. **Journal of the Operational Research Society**, v. 53, n. 5, p. 512–522, 2002. ISSN 14769360.

DARVISH, M.; ARCHETTI, C.; COELHO, L. C. Trade-offs between environmental and economic performance in production and inventory-routing problems. **International Journal of Production Economics**, Elsevier B.V., v. 217, n. July 2017, p. 269–280, 2019.

DARVISH, M. *et al.* Integrated production-distribution Systems: Trends and Perspectives. **Pesquisa Operacional**, SOBRAPO, v. 41, n. spe, p. 246080–246081, 2021. ISSN 1678-5142.

DAVIS, L. Applying Adaptive Algorithms to Epistatic Domains. *In*: Morgan Kaufmann Publishers Inc. (ed.). **Proceedings of the 9th International Joint Conference on Artificial Intelligence**. San Francisco, CA: [*S.l.: s.n.*], 1985. p. 162–164.

FRANÇA, P. M.; MENDES, A.; MOSCATO, P. A memetic algorithm for the total tardiness single machine scheduling problem. **European Journal of Operational Research**, v. 132, n. 1, p. 224–242, 2001. ISSN 0377-2217.

FUMERO, F.; VERCELLIS, C. Synchronized Development of Production, Inventory, and Distribution Schedules. **Transportation Science**, v. 33, n. 3, p. 330–340, 1999.

IZADI, L.; AHMADIZAR, F.; ARKAT, J. A Hybrid Genetic Algorithm for Integrated Production and Distribution Scheduling Problem with Outsourcing Allowed. **IJE Transactions B**, v. 33, n. 11, p. 2285–2298, 2020.

KAYÉ, B. K. B. *et al.* A memetic algorithm for an external depot production routing problem. **Algorithms**, v. 14, n. 1, p. 1–24, 2021. ISSN 19994893.

LI, Y. *et al.* An efficient three-level heuristic for the large-scaled multi-product production routing problem with outsourcing. **European Journal of Operational Research**, v. 272, n. 3, p. 914–927, 2019. ISSN 03772217.

-
- MANOUSAKIS, E. G. *et al.* An infeasible space exploring matheuristic for the Production Routing Problem. **European Journal of Operational Research**, Elsevier B.V., v. 298, n. 2, p. 478–495, 2022. ISSN 03772217.
- MENDES, A. S.; FRANCA, P. M.; MOSCATO, P. Fitness landscapes for the total tardiness single machine scheduling problem. **Neural Network World**, Citeseer, v. 12, n. 2, p. 165–180, 2002.
- MIRANDA, P. L. *et al.* A decomposition heuristic for a rich production routing problem. **Computers and Operations Research**, v. 98, p. 211–230, 2018. ISSN 03050548.
- MIRANDA, P. L.; MORABITO, R.; FERREIRA, D. Optimization model for a production, inventory, distribution and routing problem in small furniture companies. **Top**, Springer Berlin Heidelberg, v. 26, n. 1, p. 30–67, 2018. ISSN 18638279.
- MOONS, S. *et al.* Integrating production scheduling and vehicle routing decisions at the operational decision level: A review and discussion. **Computers and Industrial Engineering**, v. 104, p. 224–245, 2017.
- MOSCATO, P.; NORMAN, M. G. A Memetic Approach for the Traveling Salesman Problem Implementation of a Computational Ecology for Combinatorial Optimization on Message-Passing Systems. **International Conference on Parallel Computing and Transputer Applications**, n. January, p. 177–186, 1992.
- MOSTAFA, N.; ELTAWIL, A. Using valid inequalities to solve the integrated production-inventory-distribution-routing problem. **International Journal of Operational Research**, v. 35, n. 4, p. 551, 2019. ISSN 1745-7645.
- NEVES-MOREIRA, F. *et al.* Solving a large multi-product production-routing problem with delivery time windows. **Omega (United Kingdom)**, v. 86, p. 154–172, 2019. ISSN 03050483.
- QIU, Y.; QIAO, J.; PARDALOS, P. M. Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory. **Omega (United Kingdom)**, Elsevier Ltd, v. 82, p. 193–204, 2019. ISSN 03050483.
- QIU, Y. *et al.* Formulations and branch-and-cut algorithms for production routing problems with time windows. **Transportmetrica A: Transport Science**, v. 14, n. 8, p. 669–690, 2018. ISSN 23249943.
- QIU, Y. *et al.* A variable neighborhood search heuristic algorithm for production routing problems. **Applied Soft Computing Journal**, Elsevier B.V., v. 66, p. 311–318, 2018. ISSN 15684946.
- QIU, Y. *et al.* Formulations and branch-and-cut algorithms for multi-product multi-vehicle production routing problems with startup cost. **Expert Systems with Applications**, Elsevier Ltd, v. 98, p. 1–10, 2018. ISSN 09574174.
- RAMOS, B.; ALVES, C.; Valério de Carvalho, J. An arc flow formulation to the multitrip production, inventory, distribution, and routing problem with time windows. **International Transactions in Operational Research**, v. 00, p. 1–28, 2020.
- RUOKOKOSKI, M. *et al.* Efficient Formulations and a Branch-and-Cut Algorithm for a Production-Routing Problem. **Transportation**, p. 1–43, 2010.

RUSSELL, R. A. Mathematical programming heuristics for the production routing problem. **International Journal of Production Economics**, Elsevier Ltd, v. 193, n. May, p. 40–49, 2017. ISSN 09255273.

Salehi Sarbijan, M.; BEHNAMIAN, J. Multi-product production routing problem by consideration of outsourcing and carbon emissions: particle swarm optimization. **Engineering Optimization**, Taylor Francis, v. 0, n. 0, p. 1–17, 2020. ISSN 10290273.

SENOUSSI, A. *et al.* Heuristics Based on Genetic Algorithms for the Capacitated Multi Vehicle Production Distribution Problem. **Computers and Operations Research**, Elsevier Ltd, v. 96, p. 108–119, 2018.

SOLYALı, O.; SÜRAL, H. A multi-phase heuristic for the production routing problem. **Computers and Operations Research**, v. 87, p. 114–124, 2017.

TOLEDO, C. *et al.* Multi-population genetic algorithm to solve the synchronized and integrated two-level lot sizing and scheduling problem. **International Journal of Production Research**, Taylor Francis, v. 47, n. 11, p. 3097–3119, 2009.

VADSETH, S. T.; ANDERSSON, H.; STÅLHANE, M. A multi-start route improving matheuristic for the production routing problem. **XX**, 2022.

WAGNER, H. M.; WHITIN, T. M. Dynamic Version of the Economic Lot Size Model. **Management Science**, v. 5, n. 1, p. 89–96, 1958.

YAGMUR, E.; KESEN, S. E. Multi-trip heterogeneous vehicle routing problem coordinated with production scheduling: Memetic algorithm and simulated annealing approaches. **Computers & Industrial Engineering**, v. 161, p. 107649, nov 2021. ISSN 03608352.

ZHANG, Z. *et al.* A benders decomposition approach for the multivehicle production routing problem with order-up-to-level policy. **Transportation Science**, v. 55, n. 1, p. 160–178, 2021. ISSN 15265447.