

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Algoritmo Genético Multiobjetivo para Roteamento com Caminhões e Drones

Maria Luísa Riolino Guimarães

JUIZ DE FORA
AGOSTO, 2025

Algoritmo Genético Multiobjetivo para Roteamento com Caminhões e Drones

MARIA LUÍSA RIOLINO GUIMARÃES

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Luciana Brugiolo Gonçalves
Coorientador: Lorenza Leão Oliveira Moreno

JUIZ DE FORA
AGOSTO, 2025

ALGORITMO GENÉTICO MULTIOBJETIVO PARA ROTEAMENTO COM CAMINHÕES E DRONES

Maria Luísa Riolino Guimarães

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Luciana Brugiolo Gonçalves
D.Sc. em Ciência da Computação

Lorenza Leão Oliveira Moreno
D.Sc. em Informática

Carlos Cristiano Hasenclever Borges
D.Sc. em Engenharia Civil

Marcos de Mendonça Passini
D.Sc. em Engenharia de Sistemas e Computação

JUIZ DE FORA
13 DE AGOSTO, 2025

À minha família, pelo apoio e incentivo constantes e por não medirem esforços para que eu chegasse até aqui.

Resumo

No campo da logística verde, um dos principais desafios é desenvolver rotas de entrega que minimizem os custos econômicos e o impacto ambiental associado ao consumo energético dos veículos utilizados. Neste contexto, este trabalho explora um problema de entrega híbrida que envolve o uso de caminhões e drones. A metodologia proposta combina um novo algoritmo construtivo randomizado com uma adaptação do clássico NSGA-II (*Non-dominated Sorting Genetic Algorithm*), incluindo um novo módulo de busca local, permitindo a geração de soluções multiobjetivo que buscam minimizar a energia total consumida pelos caminhões, o custo total de entrega e o tempo total de entrega. Finalmente, os resultados obtidos demonstram que a nova estratégia construtiva, quando associada a um novo módulo de busca local, é competitiva em relação ao modelo proposto pelo artigo de referência do problema.

Palavras-chave: Algoritmo genético, logística verde, roteamento de veículos, caminhões, drones, otimização multiobjetivo.

Abstract

In green logistics, a primary challenge is creating delivery routes that minimize economic costs and the environmental impact of vehicle energy consumption. This work explores a hybrid delivery problem using trucks and drones. Our proposed methodology merges a new randomized constructive algorithm with an adapted NSGA-II (Non-dominated Sorting Genetic Algorithm), including a novel local search module, to produce multi-objective solutions. These solutions aim to minimize the trucks' total energy consumption, total delivery cost, and total delivery time. Results demonstrate that our new constructive strategy, combined with a new local search module, is competitive with the benchmark model proposed in the problem's reference article.

Keywords: Genetic algorithm, green logistics, vehicle routing, trucks, drones, multi-objective optimization.

Agradecimentos

A toda a minha família, em especial aos meus pais Liandyr Júnior e Christiane, e à minha irmã Júlia, por serem minhas principais referências, pela paciência nos momentos mais atarefados e pela fé inesgotável que sempre depositam em mim.

Às professoras Luciana Brugiolo e Lorenza Moreno pelo incentivo, orientação e sabedoria, sem os quais este trabalho não seria concluído. À Universidade Federal de Juiz de Fora, pelo apoio financeiro ao início desta pesquisa, através da Bolsa de Iniciação Científica. Ao professor Stênio, que me convidou para o projeto e cuja orientação também foi essencial para que ele fosse concluído.

Às professoras Bárbara Quintela e Alessandreia Oliveira, e a todas as voluntárias do Projeto Meninas Digitais UFJF, pelo trabalho inspirador e pelos anos de amizade e aprendizado.

Aos amigos que fiz durante a Graduação, pelo apoio diário e por tornarem mais leves os momentos difíceis.

A todos os professores e funcionários do Departamento de Ciência da Computação, do Instituto de Ciências Exatas e da UFJF em geral, que contribuem diariamente para que a Universidade siga sendo referência em ensino público, gratuito e de qualidade.

Sumário

Lista de Figuras	6
Lista de Tabelas	7
Lista de Abreviações	8
1 Introdução	9
2 Fundamentação Teórica	11
2.1 Otimização Multiobjetivo em VRP	11
2.2 Problemas de Roteamento Verde de Veículos	12
2.3 Abordagens Multiobjetivo em Roteamento com Caminhões e Drones	13
3 Definição formal do MOVRPDD	15
3.1 Funções objetivo	17
3.2 Restrições	18
4 NSGA-II para o MOVRPDD	19
4.1 Representação da solução	21
4.2 Pré-processamento	22
4.3 Inicialização da População	24
4.4 Algoritmo Construtivo	25
4.5 Operadores Genéticos e atualização da população	29
4.5.1 Operadores de Cruzamento	31
4.5.2 Operadores de Mutação	33
4.6 Busca Local Multiobjetiva	33
5 Resultados Computacionais	35
5.1 Instâncias e Ambiente Experimental	35
5.2 Resultados Experimentais	36
5.3 Análise Experimental	40
6 Considerações finais	45
Referências Bibliográficas	46

Lista de Figuras

3.1	Ilustração do MOVRPDD. Adaptado de (ZHANG et al., 2022).	16
4.1	Fluxograma das abordagens implementadas.	20
4.2	Representação da Solução	22
4.3	Ilustração do procedimento <i>tour splitting</i>	25
4.4	Estrutura da Lista de Candidatos	26
4.5	Processo de construção das rotas de caminhão.	27
4.6	Processo de codificação das soluções geradas pelo novo Construtivo.	29
4.7	Procedimento do NSGA-II. Adaptado de (DEB et al., 2002).	31
4.8	Operadores de Cruzamento.	32
4.9	Operadores de Mutação.	33
5.1	Comparativo de uma solução inicial para a instância 40-20-0.3	43
5.2	Comparativo da evolução das soluções para a instância 40-20-0.3	44

Lista de Tabelas

2.1	Classificação dos artigos quanto à variante do problema abordado.	13
5.1	Parâmetros associados ao MOVRPDD	36
5.2	Média dos resultados de IGD, HV e Tempo em instâncias com 40 clientes.	38
5.3	Média dos resultados de IGD, HV e Tempo em instâncias com 80 clientes.	38
5.4	Média dos resultados de C-metric em instâncias com 40 clientes.	39
5.5	Média dos resultados de C-metric em instâncias com 80 clientes.	39
5.6	Média dos resultados de IGD, HV e Tempo em instâncias com 120 clientes.	40
5.7	Média dos resultados de IGD, HV e Tempo em instâncias com 160 clientes.	40
5.8	Média dos resultados de C-metric em instâncias com 120 clientes.	41
5.9	Média dos resultados de C-metric em instâncias com 160 clientes.	41
5.10	Resultados comparativos do teste de Wilcoxon.	42
5.11	Ranking médio dos algoritmos analisados, segundo o teste de Friedman. . .	42

Lista de Abreviações

DCC	Departamento de Ciência da Computação
ENSGA-II	Extended Non-dominated Sorting Genetic Algorithm
GA	Genetic Algorithm (algoritmo genético)
LM	Last Mile (última milha)
MOVRPDD	Multi-objective vehicle routing problem with drone delivery and dynamic flight endurance
NSGA-II	Non-dominated Sorting Genetic Algorithm
TW	Time windows (janelas de tempo)
UFJF	Universidade Federal de Juiz de Fora
VANTs	Veículos aéreos não tripulados
VRP	Vehicle Routing Problem

1 Introdução

Em 2015, a Organização das Nações Unidas lançou a Agenda 2030, um plano de ações global que estabeleceu objetivos e metas a serem cumpridos de forma coletiva por governos, empresas e pela sociedade civil, em busca de um futuro mais sustentável, resiliente e justo (Organização das Nações Unidas, 2015). Entre os 17 Objetivos de Desenvolvimento Sustentável (ODS) definidos, estão a promoção de padrões responsáveis de produção e consumo, o incentivo à inovação em infraestrutura e a redução dos impactos ambientais causados por atividades humanas. A crescente emissão de gases de efeito estufa é uma das principais preocupações nesse sentido, por intensificar o aquecimento da atmosfera terrestre e contribuir para as mudanças climáticas, com impactos diretos sobre ecossistemas, economia e saúde humana (IPCC, 2023).

Dados do Instituto de Energia e Meio Ambiente e Observatório do Clima (2024) mostram que o setor de transportes atingiu um recorde histórico no consumo de combustíveis fósseis no Brasil, sendo responsável por mais de 40% das emissões de gases de efeito estufa no ano de 2023, o que reforça a importância da busca por soluções inovadoras e sustentáveis para os processos logísticos tradicionais. Neste contexto, a logística verde (NIKSERESHT; GOLMOHAMMADI; ZANDIEH, 2024) surge como uma abordagem que não se limita apenas à escolha de veículos mais eficientes, mas também envolve a otimização de rotas de transporte para reduzir o impacto ambiental, melhorar a eficiência no uso de recursos e diminuir os custos operacionais.

Diversas abordagens de roteamento vêm sendo exploradas nos últimos anos para alcançar esses objetivos, como é o caso da utilização de drones no transporte de alimentos e mercadorias, segmento que se popularizou na China e está sendo explorado por diversas empresas mundialmente (WOODWORTH, 2024; YANG, 2023; iFood, 2022), uma vez que oferece flexibilidade nas rotas e potencial para reduzir o consumo de combustível e a pegada de carbono. Por outro lado, os drones apresentam alguns desafios logísticos, em especial quanto à baixa autonomia de voo e capacidade de carga, o que pode ser compensado, em contextos mais complexos de entrega, com a implementação de sistemas

de transporte híbridos, que combinam veículos tradicionais com novas tecnologias.

Diante desse cenário, este trabalho trata de uma abordagem híbrida que visa encontrar soluções otimizadas para um problema de roteamento de veículos com entregas realizadas por pares de caminhões e drones, no qual são considerados os aspectos econômicos das entregas — neste caso, o custo e o tempo —, e também a sustentabilidade, por meio da redução do consumo de energia dos caminhões, contribuindo assim para um modelo logístico mais verde e eficiente.

O algoritmo genético NSGA-II foi utilizado como base da implementação, devido à sua baixa demanda por parâmetros e ao seu caráter elitista que favorece a obtenção de soluções de qualidade em problemas multiobjetivo. O procedimento padrão foi estendido com operadores genéticos de cruzamento e mutação, além de um método de codificação das soluções que considera as especificidades das rotas de caminhão e drone. Além disso, um novo módulo de inicialização da população foi desenvolvido, inspirado na fase construtiva do GRASP, com o objetivo de acelerar a convergência algorítmica. Finalmente, uma busca local multiobjetivo foi incluída no procedimento, a fim de que o espaço de solução fosse melhor explorado ao longo das gerações.

Os próximos capítulos estão divididos da seguinte forma. O Capítulo 2 apresenta uma revisão bibliográfica da literatura. Em seguida, a definição formal do problema estudado é feita no Capítulo 3. O Capítulo 4 descreve as abordagens propostas, detalhando a representação da solução e os algoritmos desenvolvidos. No Capítulo 5 os experimentos computacionais realizados são descritos e os resultados obtidos são analisados. Finalmente, o Capítulo 6 traz as considerações finais e sugestões de trabalhos futuros.

2 Fundamentação Teórica

Um dos tópicos mais estudados no campo da otimização combinatória é o problema de roteamento de veículos (VRP) e suas variantes. O VRP clássico, introduzido por Dantzig e Ramser (1959) e posteriormente detalhado por Clarke e Wright (1964), pode ser descrito como uma generalização do Problema do Caixeiro Viajante (FLOOD, 1956), onde o conjunto de rotas de menor custo deve ser determinado para uma frota homogênea de veículos que atende a um conjunto de clientes a partir de um depósito central, sem que a capacidade de carga dos veículos utilizados seja ultrapassada.

A partir da definição clássica do problema, que já é classificado como NP-difícil (LENSTRA; KAN, 1981), inúmeras variações foram exploradas ao longo dos anos, a fim de incorporar os desafios trazidos pelas novas aplicações práticas que surgem continuamente em cenários reais. Em geral, VRPs costumam envolver janelas de tempo, que delimitam um período específico do dia para que as tarefas sejam realizadas; a presença de múltiplos depósitos; a utilização de frotas heterogêneas, ou seja, contendo tipos variados de veículos ou, ainda, restrições relacionadas ao tráfego e à autonomia dos veículos (TAN; YEH, 2021).

2.1 Otimização Multiobjetivo em VRP

Para além das variações nas restrições, o Problema de Roteamento de Veículos (VRP) pode ainda ser formulado como um problema multiobjetivo, no qual diferentes critérios, frequentemente conflitantes, devem ser otimizados simultaneamente (JOZEFOWIEZ; SEMET; TALBI, 2008).

Duan, He e Yen (2022), por exemplo, apresentam um VRP no qual a distância percorrida pelos veículos deve ser minimizada, sendo que, concomitantemente, o número de veículos utilizados também deve ser mínimo.

Também é possível que, em problemas multiobjetivo, alguns objetivos sejam de minimização, enquanto outros são de maximização. É o caso do VRP para logística de

cadeia fria abordado por Yang e Tao (2023), no qual os custos operacionais devem ser minimizados e, adicionalmente, a satisfação dos consumidores deve ser a maior possível.

Nesses casos em que há múltiplos objetivos, a otimização visa encontrar um conjunto de soluções, conhecidas como fronteira de Pareto, que se caracterizam por não serem dominantes entre si, ou seja, nenhuma solução pode ser melhorada em relação a um objetivo sem que ao menos um outro objetivo seja modificado (DEB, 2001).

2.2 Problemas de Roteamento Verde de Veículos

No últimos anos, a comunidade acadêmica tem considerado cada vez mais as questões ambientais ao tratar de problemas de otimização. Os Problemas de Roteamento Verde de Veículos (GVRPs) exploram maneiras de reduzir o impacto ambiental dos processos logísticos através de esquemas de roteamento que utilizem abordagens menos prejudiciais ao meio ambiente (ERDOĞAN; MILLER-HOOKS, 2012).

Sabet e Farooq (2022) classificam os GVRPs em três principais categorias: GVRPs com veículos tradicionais; GVRPs com veículos que utilizam combustíveis alternativos (e.g. veículos elétricos ou movidos a biocombustíveis) e, finalmente, GVRPs com frotas heterogêneas, ou seja, que associam veículos com características distintas.

Em especial, o uso de veículos aéreos não tripulados (VANTs), popularmente conhecidos como drones, tem sido explorado como alternativa inovadora no contexto da logística verde (MADANI; NDIAYE, 2022), principalmente quando utilizados em conjunto com caminhões tradicionais, por ser uma possível estratégia para a redução de custos, tempos de entrega e impactos ambientais. Em comparação com caminhões, os drones apresentam baixo impacto ambiental e alta velocidade, mas possuem baixa capacidade de carga e autonomia de voo limitada, o que justifica a associação dos veículos em processos de entrega.

2.3 Abordagens Multiobjetivo em Roteamento com Caminhões e Drones

Recentemente, um número crescente de trabalhos na literatura tem explorado a associação de caminhões e drones em VRPs, considerando configurações logísticas variadas e, em muitos casos, múltiplos objetivos de otimização (ZHANG et al., 2024). A Tabela 2.1 apresenta algumas das variações desses problemas abordados por artigos publicados nos últimos cinco anos. Na tabela, “Dep.” refere-se à quantidade de depósitos envolvidos no problema; “Obj.” diz respeito à quantidade de objetivos; “TW” são os problemas que consideram janelas de tempo e “LM” são aqueles que tratam apenas da última milha nos processos de entrega.

Tabela 2.1: Classificação dos artigos quanto à variante do problema abordado.

Referência	Quantidade			TW	LM
	Caminhões/Drones	Dep.	Obj.		
Bektur (2024)	Múltiplos caminhões,	1	2		✓
Momeni et al. (2022)	Múltiplos drones	N	2		
Das et al. (2021)		1	2	✓	
Han et al. (2020)	Múltiplos pares	N	2	✓	
Lu, Yang e Yang (2023)	caminhão-drone	1	2		
Zhang et al. (2022)		1	3		
Gonzalez-R et al. (2024)		1	2		✓
Kong et al. (2024)	Um caminhão equipado	1	2		✓
Luo et al. (2022)	com múltiplos drones	1	2	✓	✓
Luo et al. (2023)		1	3		
Xiong, Lei e Li (2022)	Um caminhão equipado com um drone	1	2		

As abordagens multiobjetivo adotadas ao longo dos últimos anos são diversas, mas observa-se uma prevalência de estratégias evolucionistas, em especial algoritmos genéticos, além da aprendizagem por reforço (*Reinforcement Learning*) e da utilização de *solvers*. Dentre os algoritmos genéticos, o mais popular entre os autores é o NSGA-II (*Non-dominated Sorting Genetic Algorithm*), introduzido por DEB et al. em 2002.

Entre os VRPs, destaca-se a utilização de drones na última milha (LM), que representa a etapa final do processo de entrega de mercadorias, ocorrendo entre o ponto mais próximo do destino final e o local de entrega.

Bektur (2024) utiliza uma estratégia que envolve programação matemática, um

algoritmo genético e aprendizagem por reforço para resolver o problema da última milha, em uma variação que considera múltiplos caminhões e múltiplos drones, visando minimizar o custo econômico das entregas, enquanto a satisfação dos clientes atendidos é maximizada. Já Kong et al. (2024) utilizam aprendizagem por reforço para minimizar o tempo e o custo econômico das entregas, com uma frota de veículos que é composta por múltiplos drones associados a apenas um caminhão.

Da mesma forma, outros autores também abordam variações que associam vários drones a um único caminhão para realizar entregas na última milha: enquanto Gonzalez-R, Sanchez-Wells e Andrade-Pineda (2024) implementam a técnica de *Simulated Annealing* para minimizar o tempo e a emissão de gás carbônico, Luo et al. (2022) utilizam um algoritmo genético adaptado para minimizar custos e maximizar a satisfação dos clientes, enquanto consideram, também, janelas de tempo (TW) de entrega.

Em problemas de roteamento, janelas de tempo são intervalos específicos nos quais os clientes podem ser atendidos. Trata-se de outra variação muito comum de VRPs, discutida, por exemplo, em Das et al. (2021) e Han et al. (2020). Em ambos os trabalhos, múltiplos pares de caminhões e drones são utilizados. O primeiro utiliza uma abordagem adaptada do *NSGA-II* associada a outro algoritmo evolutivo, o *Ant Colony Optimization*. O segundo, por sua vez, implementa um algoritmo de colônia de abelhas (*Bee Colony Optimization*) e considera a utilização de múltiplos depósitos.

Há outros trabalhos na literatura que tratam de frotas com múltiplos pares caminhão-drone, como a variação apresentada por Lu, Yang e Yang (2023), na qual uma combinação de programação linear inteira e computação evolucionista foi utilizada para um problema que inclui a sincronização entre os voos dos drones, e a versão proposta por Zhang et al. (2022), que busca minimizar três objetivos, dois econômicos e um ambiental, e será melhor explorada no próximo capítulo.

3 Definição formal do MOVRPDD

O Problema Multiobjetivos de Roteamento de Veículos com Entrega por Drones e Autonomia de Voo Dinâmica (MOVRPDD, do inglês “Multi-objective vehicle routing problem with drone delivery and dynamic flight endurance”) foi introduzido e modelado por Zhang et al. (2022) e consiste em realizar entregas a um conjunto de clientes, usando uma frota de caminhões que sai de um único depósito, cada caminhão equipado com um drone, sendo que:

- Todos os clientes devem ter suas demandas atendidas em uma única visita;
- O tamanho da frota (número de pares caminhão-drone) é definido a partir da razão entre a demanda total dos clientes e a capacidade máxima de carga de cada caminhão;
- Todas as rotas iniciam-se e terminam no depósito;
- Os caminhões podem realizar as entregas diretamente aos clientes ou servir como pontos de decolagem e pouso para os drones, que, por sua vez, realizam entregas de acordo com sua autonomia de voo;
- Um drone decola do caminhão a partir de determinado cliente (atendido pelo caminhão), atende um único cliente e retorna para o mesmo caminhão em um dos clientes seguintes da rota.
- Um drone não pode retornar para o mesmo cliente de onde partiu, ou seja, enquanto um voo é executado, o caminhão deve seguir para um dos próximos clientes da rota.

A Figura 3.1 ilustra uma solução do problema, na qual 13 clientes são atendidos por três pares caminhão-drone. A primeira rota atende aos clientes $\{4, 7, 8, 10\}$, sendo que o cliente 8 é atendido por um drone, que decola do cliente 7 e retorna ao caminhão no cliente 10. Os clientes $\{6, 9, 5, 3, 11, 2\}$ são atendidos na segunda rota, na qual o drone

sai do cliente 5, realiza a entrega da mercadoria ao cliente 3 e reencontra o caminhão no cliente 2. O terceiro par caminhão-drone, por fim, atende aos clientes $\{13, 1, 12\}$.

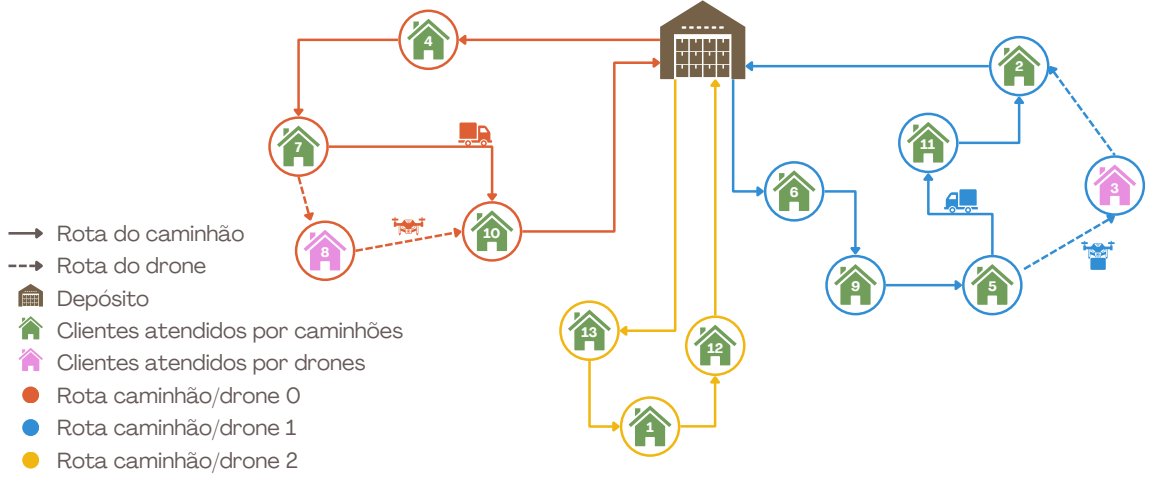


Figura 3.1: Ilustração do MOVRPDD. Adaptado de (ZHANG et al., 2022).

O MOVRPDD pode ser modelado como um multigrafo $G = (N_0, A)$, ponderado nas arestas, onde $N_0 = \{d\} \cup N$ é o conjunto de vértices, sendo d o depósito e N o conjunto de clientes a serem atendidos. Para cada cliente $i \in N$, são conhecidas a demanda $q_i \geq 0$, as coordenadas de localização (x_i, y_i) e a classificação quanto ao tipo de atendimento (apenas caminhões ou caminhões e drones). $V = \{1, 2, \dots, v\}$ representa o conjunto de pares caminhão-drone disponíveis na frota. Para cada aresta $(i, j) \in A$, são calculadas duas distâncias: a distância de Manhattan, d_{ij}^M , utilizada quando um caminhão percorre o trajeto entre os pontos i e j , e a distância Euclidiana, d_{ij}^E , correspondente ao percurso realizado por um drone entre os mesmos pontos. Além disso, são conhecidas a constante E , que representa a autonomia máxima de voo de um drone sem carga, o custo base por utilização de um par caminhão-drone C_B e, sobre ambos os veículos: a capacidade máxima de carga (Q_T e Q_D), o peso sem carga (W_T e W_D), o custo de viagem associado a uma unidade de distância (C_T e C_D) e a velocidade média de viagem (S_T e S_D). Estes valores são utilizados na definição das funções-objetivo e das restrições, conforme apresentado a seguir.

3.1 Funções objetivo

Este problema de otimização envolve a minimização de três funções objetivo. O **primeiro objetivo**, f_1 , relaciona-se ao aspecto sustentável do problema, no qual busca-se minimizar o consumo de energia dos caminhões. Em f_1 , o consumo dos drones não é considerado, uma vez que sua contribuição é insignificante quando comparada ao impacto associado ao consumo dos caminhões. A função de minimização associada a esse objetivo é descrita na Equação 3.1, onde d_{ij}^M é a distância de Manhattan entre os pontos i e j ; W_{ij}^v é o peso do caminhão carregado com a mercadoria ainda não entregue quando o trecho entre i e j é percorrido; x_{ij}^v é uma variável binária que indica se o caminhão v percorreu o trecho (i, j) .

$$Min f_1 = \sum_{v \in V} \sum_{i \in N_0} \sum_{j \in N} d_{ij}^M W_{ij}^v x_{ij}^v \quad (3.1)$$

Os outros **dois objetivos** são econômicos e referem-se ao custo total das rotas (f_2) e ao tempo decorrido até a finalização de todas as entregas (f_3).

Na Equação 3.2, soma-se o custo total da rota associada a cada par caminhão-drone. A constante C_T representa o custo de viagem de caminhões por unidade de distância; C_D é o custo de viagem de drones por unidade de distância; $(d_{ij}^E + d_{jk}^E)$ é a distância Euclidiana de cada voo (i, j, k) , onde i é o ponto de onde o drone decola, j é o cliente atendido e k é o ponto de retorno. A variável binária y_{ijk} indica se o voo (i, j, k) ocorreu e, por fim, $x_{0j}^v C_B$ é o custo base associado à utilização de cada par caminhão-drone.

$$Min f_2 = \sum_{v \in V} \left(\sum_{i \in N_0} \sum_{j \in N_0} C_T d_{ij}^M x_{ij}^v + \sum_{i \in N} \sum_{j \in N} \sum_{k \in N} C_D (d_{ij}^E + d_{jk}^E) y_{ijk} \right) + \sum_{v \in V} \sum_{j \in N} x_{0j}^v C_B \quad (3.2)$$

A Equação 3.3, por fim, refere-se ao tempo máximo entre as rotas, ou seja, o tempo despendido até que o último par caminhão-drone retorne ao depósito, após todos os clientes serem atendidos.

$$Min f_3 = \max_{v \in V} \{t_{n+1}^v\} \quad (3.3)$$

3.2 Restrições

Em linhas gerais, o modelo considera que a autonomia de voo de um drone é dinâmica, ou seja, depende diretamente de sua carga. Além disso, as rotas dos caminhões são calculadas considerando a distância de Manhattan, enquanto as rotas dos drones são calculadas com a distância Euclidiana. O tempo de decolagem e pouso dos drones é irrelevante, assim como o tempo de carregamento dos mesmos. Se um drone decola de um cliente i e retorna para um cliente k , o seu respectivo caminhão deve passar por ambos os clientes. Por fim, os tempos de viagem dos drones e caminhões devem ser sincronizados, de forma que o caminhão sempre chegue no cliente de retorno em tempo de receber o drone. O modelo completo do problema está detalhado em (ZHANG et al., 2022).

4 NSGA-II para o MOVRPDD

Diferente de outros problemas de otimização nos quais o foco é obter uma solução ótima, ou seja, que apresente o melhor valor possível para determinada função-objetivo, os problemas multiobjetivo buscam obter um conjunto de soluções não-dominadas, denominadas Pareto-ótimas, nas quais um dos objetivos não pode ser melhorado sem que pelo menos um dos outros seja afetado.

Os algoritmos evolutivos constituem uma das abordagens mais proeminentes para a resolução destes problemas (SHARMA; KUMAR, 2022). Dentre eles, destaca-se o NSGA-II (DEB et al., 2002), algoritmo que foi utilizado por Zhang et al. (2022) como base para o ENSGA-II, uma versão adaptada do algoritmo genético clássico com extensões desenvolvidas para o contexto específico do MOVRPDD.

O NSGA-II básico consiste em aplicar operadores de cruzamento e mutação em indivíduos (soluções) de determinada população inicial P_0 , que tem tamanho N , até que uma nova população G seja formada com N novos indivíduos gerados. P_0 e G são mescladas e os indivíduos são ordenados em fronteiras de Pareto. Cada fronteira F_i constitui-se de um grupo de soluções não dominadas entre si. Os N melhores indivíduos são mantidos para a próxima geração, e o procedimento se repete até que o número definido de gerações seja alcançado.

Algumas adaptações foram desenvolvidas por Zhang et al. (2022) no NSGA-II original para que ele pudesse ser utilizado para o problema abordado neste trabalho. Dentre elas, destacam-se um método de codificação e decodificação das soluções, haja vista a necessidade de considerar as especificidades das rotas de caminhão e drone; os operadores de cruzamento e mutação, e um módulo de busca local denominado “*multi-dimensional local search*”. Da mesma forma, este trabalho introduziu um novo método construtivo para a criação da população inicial e uma nova busca local.

A Figura 4.1 sintetiza o procedimento completo implementado para o MOVRPDD, incluindo as duas abordagens de construção da população inicial e um novo módulo opcional de busca local. Ao longo deste capítulo, as principais etapas envolvidas no

procedimento serão apresentadas e discutidas, em especial quanto às diferenças entre a implementação da literatura e aquela adotada por este trabalho.

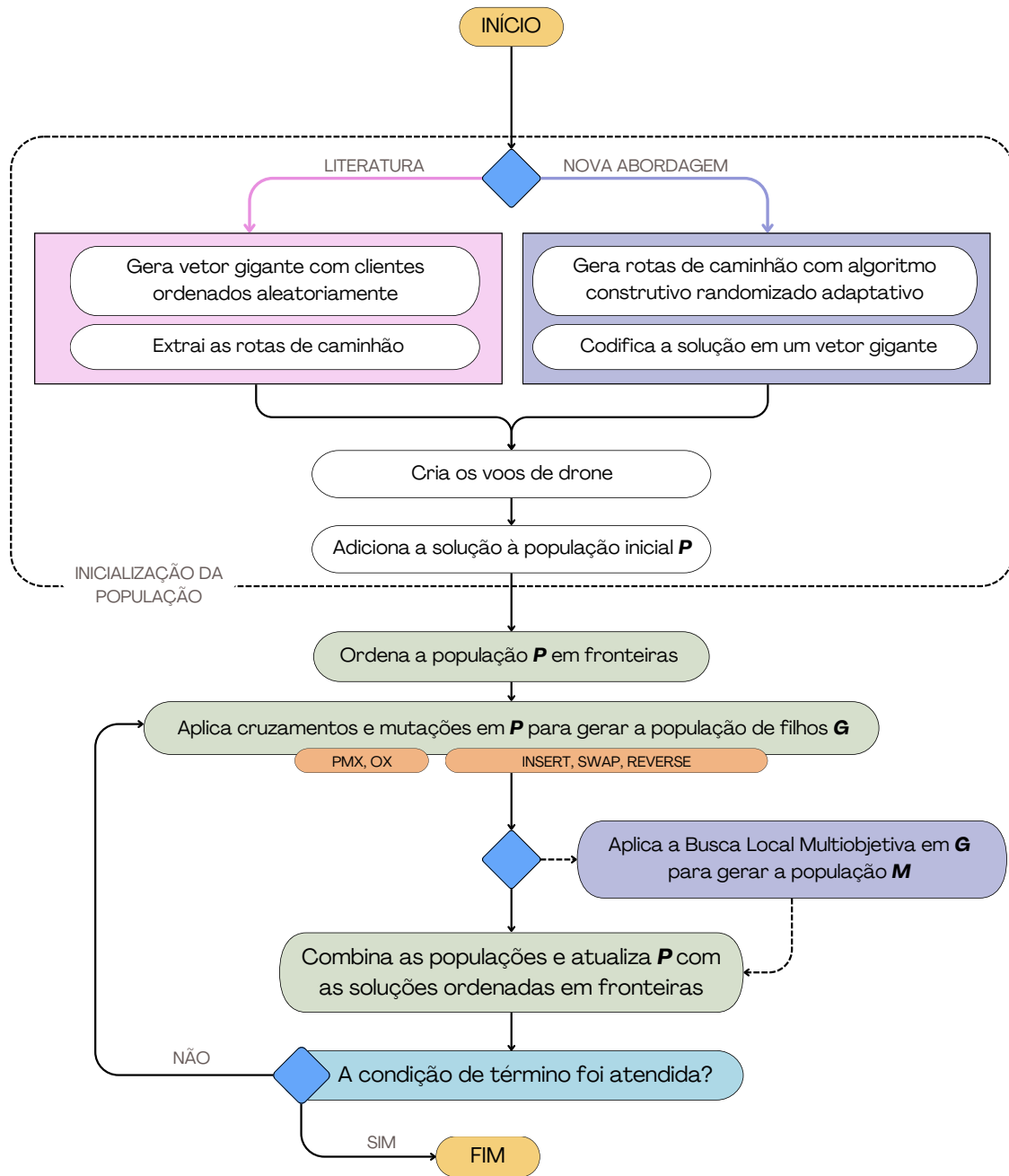


Figura 4.1: Fluxograma das abordagens implementadas.

4.1 Representação da solução

Em um algoritmo genético, cada solução deve ser representada por um cromossomo. Trata-se de uma estrutura que permite a aplicação de operadores genéticos. No MOVRPDD, cada gene do cromossomo corresponde a um cliente presente na rota de cada par caminhão-drone.

O processo de transformar as rotas em cromossomos é chamado de **codificação**. Neste trabalho, elas são representadas por um vetor de inteiros (identificadores), aqui referido como “rota gigante”. Da mesma forma, a **decodificação** é o processo de transformar essa sequência de identificadores em rotas específicas. O processo de construção dessas rotas é detalhado na Seção 4.3.

Na representação adotada para o MOVRPDD, utiliza-se, além da “rota gigante”: um conjunto de rotas caminhão-drone; as três funções-objetivo (f_1, f_2, f_3) e outros indicadores utilizados para identificar a dominância na população. Em uma estrutura de rota caminhão-drone, são armazenados o percurso de cada caminhão e os voos executados por seu respectivo drone.

A solução apresentada na Figura 4.2 mostra uma possibilidade de entrega para uma instância com 13 clientes com 3 pares caminhão-drone. Para cada uma das rotas apresenta-se o conjunto de clientes atendidos pelo caminhão (*truckRoute*) e os trechos das rotas realizadas pelos drones (*droneRoute*). Cada voo (i, j, k) é constituído pelo cliente de decolagem i , o cliente j atendido pelo drone e o cliente de retorno k . Enquanto nas duas primeiras rotas há entregas por caminhão e drone, na terceira todas as entregas são realizadas apenas pelo caminhão.

A “rota gigante” é constituída por todos os clientes da rota, e sua construção depende da estratégia construtiva adotada. Na abordagem apresentada por este trabalho, as rotas de caminhão são inicialmente construídas por um algoritmo guloso e, em seguida, concatenadas para formá-la. O vetor auxiliar de primeiros clientes é utilizado para demarcar o índice de início de cada uma das rotas, facilitando o processo de decodificação das soluções. As rotas dos drones são extraídas a partir dessa estrutura, através de um algoritmo determinístico.

O início e o final das rotas de cada par caminhão-drone são demarcados com o

índice '0', que refere-se ao depósito. Conforme a especificação do problema, os clientes de decolagem e retorno devem estar incluídos sequencialmente no percurso do caminhão, mas não são obrigatoriamente consecutivos.

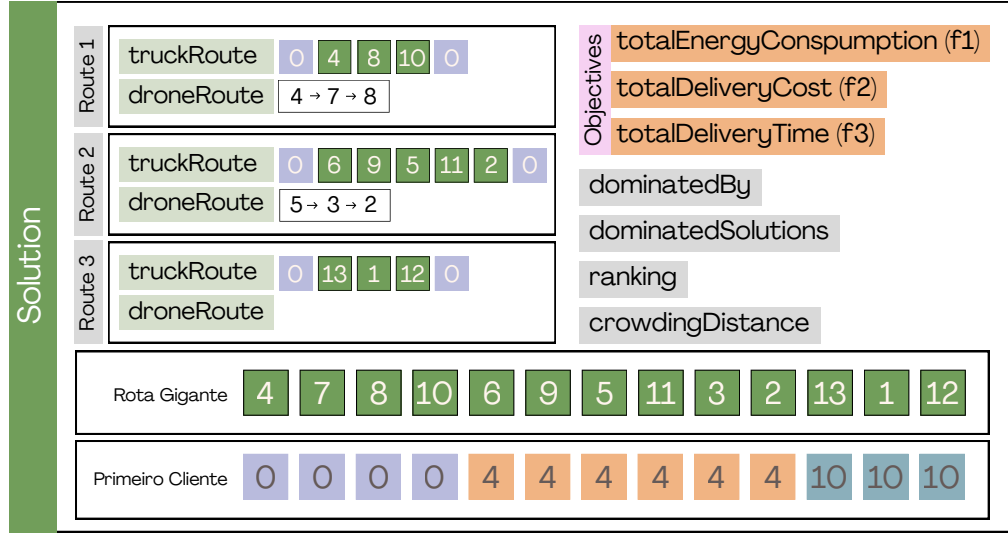


Figura 4.2: Representação da Solução

A instância do problema é modelada como um grafo, no qual cada cliente ou depósito é representado por um nó que contém informações quanto à sua **classificação** (depósito ou cliente); um **identificador numérico** (o depósito é sempre 0); sua **localização**, através de coordenadas (x, y) ; sua **demand**a, ou seja, o peso dos itens a serem entregues e, por fim, o **tipo de serviço**, que determina se um cliente pode ou não ser atendido por um drone.

4.2 Pré-processamento

Uma etapa de pré-processamento é implementada para calcular e sistematizar informações que serão utilizadas ao longo da execução. Para tanto, cinco matrizes $N \times N$ são criadas:

- **Matrizes de Distância:** métodos diferentes são utilizados para calcular a distância percorrida por caminhões e drones entre os nós i e j . Portanto, são criadas matrizes de distância para cada modo de transporte. O percurso de um caminhão é calculado em termos de distância de Manhattan, conforme a Equação 4.1, onde (x_i, y_i) são as

coordenadas do cliente i e (x_j, y_j) são as coordenadas do cliente j . Já o percurso de um drone (uma perna do voo) é calculado pela distância Euclidiana, conforme a Equação 4.2.

$$d_{ij}^M = |x_i - x_j| + |y_i - y_j| \quad (4.1)$$

$$d_{ij}^E = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (4.2)$$

Esta distinção deve ser feita pois, enquanto os caminhões respeitam as vias terrestres para locomoção, a trajetória de voo dos drones é feita em linhas retas

- **Matrizes de Tempo:** o tempo de deslocamento entre dois nós é utilizado no cálculo da função-objetivo f_3 . Por isso, são criadas matrizes para o tempo de deslocamento de ambos os veículos entre os nós i e j . O cálculo do tempo de viagem dos caminhões e drones se dá, respectivamente, pelas Equações 4.3 e 4.4, onde S_T é a velocidade média do caminhão e S_D , a do drone.

$$t_{ij}^T = d_{ij}^M \times S_T \quad (4.3)$$

$$t_{ij}^D = d_{ij}^E \times S_D \quad (4.4)$$

- **Fator de escala dinâmico:** é utilizado para determinar a autonomia máxima de voo dos drones em cada cenário. Um voo (i, j, k) é viável quando seu tempo de execução é menor ou igual ao resultado da multiplicação entre a distância a ser percorrida pelo drone e o fator de escala dinâmico. Esta grandeza é calculada segundo a Equação 4.5 apenas quando um drone sai do cliente i para atender o cliente j , onde q_j é a demanda do cliente j e Q_D é a capacidade máxima de carga do drone. Quando o drone retorna de j para k , por estar vazio, o fator de escala é igual a 1, haja vista que não há impacto na velocidade de voo devido ao peso da encomenda transportada.

$$\theta_{ij}^v = 100\% - 0.2 \times \frac{q_j}{Q_D} \quad (4.5)$$

4.3 Inicialização da População

No artigo de referência, ZHANG et al. inicializam cada solução da população por meio de um vetor de N inteiros que contém os índices de todos os clientes dispostos de maneira aleatória. A partir deste vetor, ou “rota gigante”, as rotas de caminhão são extraídas conforme o algoritmo de *tour splitting* proposto por Prins, Labadi e Reghioui (2009).

Essencialmente, o procedimento de *tour splitting* consiste em duas fases:

- ***Split***: a “rota gigante” é percorrida a fim de que todas as rotas possíveis formadas por fragmentos sequenciais de clientes sejam avaliadas. O índice do início de cada rota escolhida é registrado em uma lista.
- ***Extraction***: a lista de índices é percorrida e as rotas de caminhão são extraídas.

A implementação eficiente dos algoritmos é disponibilizada em (PRINS; LABADI; REGHIOUI, 2009). Por se tratar de um procedimento determinístico, assim como o algoritmo de extração de voos que será apresentado adiante, a “rota gigante” é utilizada no ENSGA-II e também no presente trabalho como a codificação da solução, ou seja, a estrutura que representa um cromossomo no algoritmo genético. Consequentemente, o procedimento de decodificação das soluções consiste na extração das rotas de caminhão e drone a partir da “rota gigante”.

A Figura 4.3 ilustra o processo de construção das rotas de caminhão e a posterior extração dos voos de drone no ENSGA-II. A etapa de *Split* quebra a “rota gigante” em sub-rotas com melhor custo, demarcando o primeiro cliente de cada sub-rota na lista auxiliar de identificadores:

- A primeira sub-rota é formada por $\{4, 7, 8, 10\}$. O primeiro cliente é o 4, e seu índice na “rota gigante” é 0. Dessa forma, como há quatro clientes na sub-rota, as quatro primeiras posições do vetor auxiliar são preenchidas com o índice 0;
- A segunda sub-rota é formada por $\{6, 9, 5, 11, 3, 2\}$. O primeiro cliente é o 6, que está na posição 4 na “rota gigante”. Assim, as próximas seis posições do vetor auxiliar são preenchidas com o índice 4;

- Por fim, $\{13, 1, 12\}$ compõem a última sub-rotas. O primeiro cliente é 13, localizado no índice 10 da “rota gigante”. Logo, as três últimas posições do vetor auxiliar são preenchidas com o índice 10.

Depois desta etapa, a “rota gigante” é dividida (*extract routes*) e, em seguida, os voos de drone são extraídos (*create drone routes*). O procedimento para a criação dos voos de drone é detalhado por Zhang et al. (2022), e envolve a exploração sequencial das rotas, na tentativa de utilizar os veículos aéreos para atender à maior quantidade possível de clientes, sem que as restrições do problema sejam desrespeitadas.

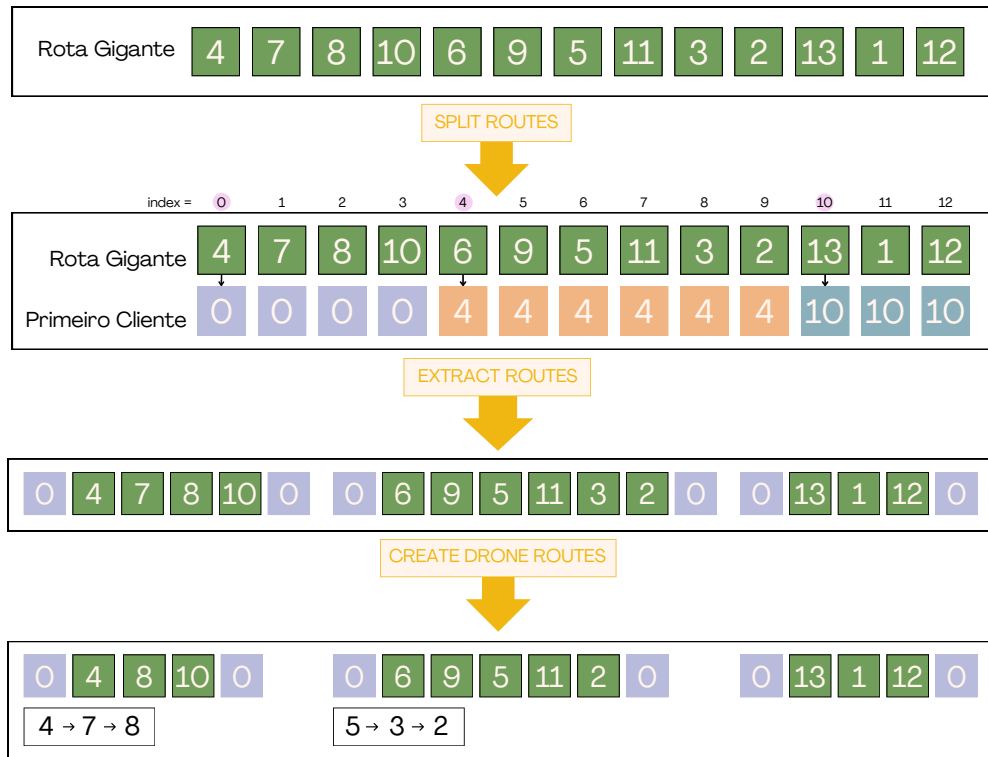


Figura 4.3: Ilustração do procedimento *tour splitting*.

Com o objetivo de introduzir soluções de melhor qualidade na população inicial, e potencialmente acelerar a convergência algorítmica, neste trabalho foi desenvolvido um novo algoritmo construtivo, descrito na próxima seção.

4.4 Algoritmo Construtivo

O algoritmo construtivo proposto foi inspirado na fase construtiva do GRASP (*Greedy Randomized Adaptive Search Procedure*), introduzido por Feo e Resende (1995), tratando-

se de uma abordagem randomizada capaz de gerar diferentes soluções a cada execução.

A quantidade de rotas geradas na solução é determinada pela Equação 4.6, onde q_i é a demanda de cada cliente e Q_T é a capacidade de carga máxima de um caminhão. Inicialmente, um cliente selecionado aleatoriamente é inserido em cada uma das rotas criadas para dar início à construção da solução.

$$n_r = \frac{\sum_{i \in N} q_i}{Q_T} + 1 \quad (4.6)$$

A partir disso, as rotas são preenchidas com base na Heurística da Inserção Mais Barata (ROSENKRANTZ; STEARNS; LEWIS II, 1977). Uma Lista de Candidatos (LC) é criada para cada uma das rotas da solução, possibilitando que a inserção de cada cliente k restante seja avaliada para cada segmento (i, j) de cada rota. A Figura 4.4 apresenta a estrutura da lista: o identificador do cliente a ser incluído; o identificador da rota na qual o cliente será incluído; o custo da inserção do cliente naquele segmento; os identificadores dos nós i e j entre os quais o novo cliente k será, potencialmente, inserido.

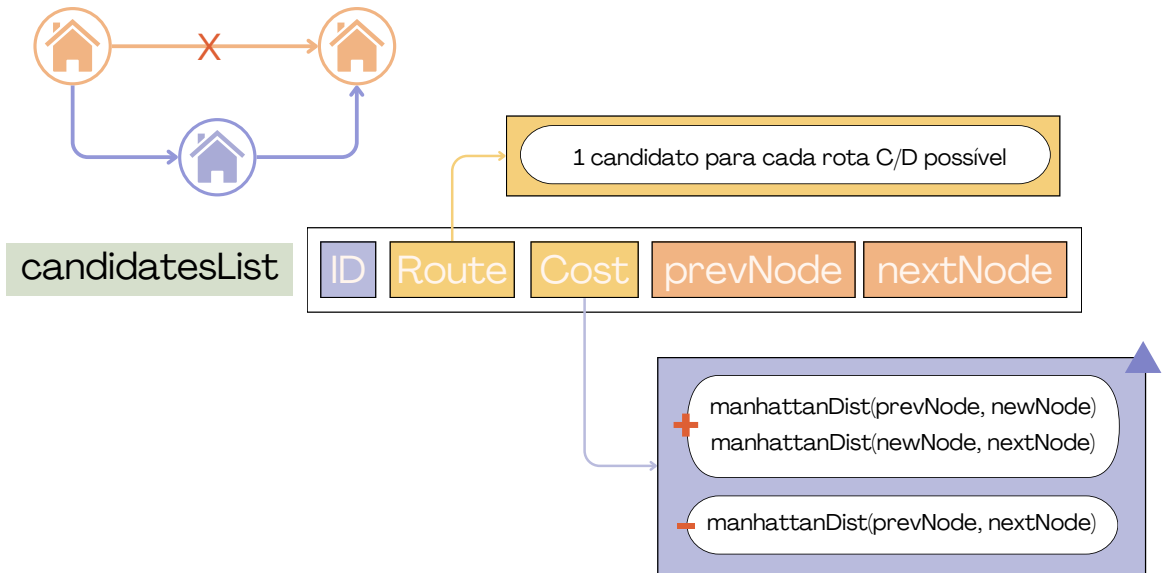


Figura 4.4: Estrutura da Lista de Candidatos

Para calcular o custo de inserção do cliente k entre i e j , avalia-se o custo da remoção da aresta (i, j) em detrimento do acréscimo das arestas (i, k) e (k, j) , sendo que

cada aresta representa a distância de Manhattan entre os nós.

Com os candidatos iniciais estabelecidos, a LC é ordenada de acordo com o custo, e uma Lista de Candidatos Restrita (LCR) é formada com os α primeiros candidatos. Um candidato aleatório é escolhido a partir da LCR, e a LC é, então, atualizada com o novo cenário da solução atual. O procedimento continua até que todos os candidatos sejam atendidos. A Figura 4.5 exemplifica este processo para um cenário com 13 clientes, onde, devido à demanda, 3 rotas de caminhão são criadas.

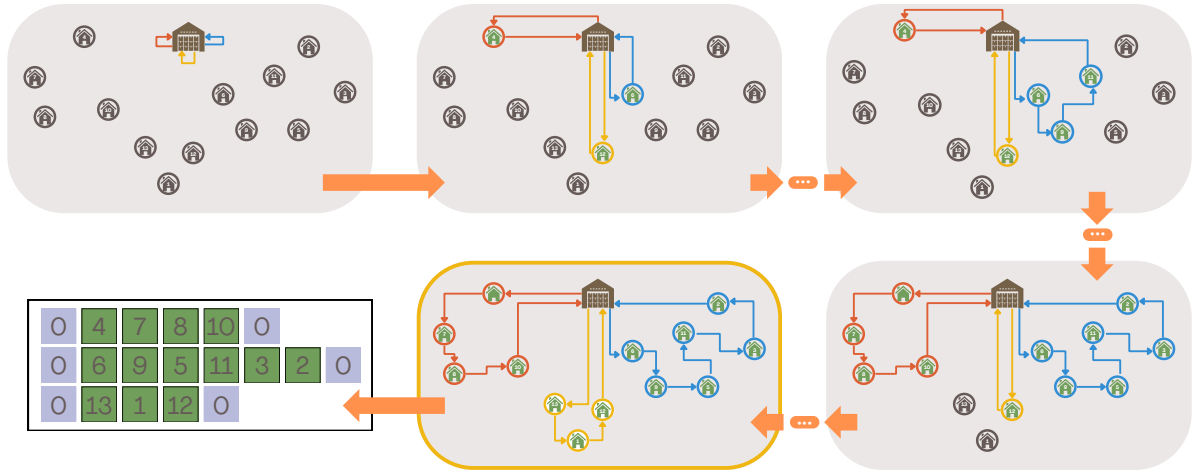


Figura 4.5: Processo de construção das rotas de caminhão.

O novo procedimento para construção da população inicial é apresentado no Algoritmo 1. A função recebe como parâmetros de entrada uma lista com valores α para a criação da LCR, uma lista C com os identificadores de todos os clientes a serem atendidos, a quantidade de rotas n_r , conforme definido pela Equação 4.6, e o tamanho da população n_p .

A lista de valores L_α é utilizada para definir o tamanho da LCR em cada iteração. Se em determinado momento a LC contém 30 candidatos e o $\alpha \in L_\alpha$ escolhido é igual a 0.5, por exemplo, os 15 melhores candidatos da LC serão considerados para a LCR. Se $\alpha = 0.25$ e a LC tem tamanho 30, os 7 melhores candidatos serão considerados.

Após a construção das rotas de caminhão, a solução deve ser codificada de modo que as próximas etapas do procedimento ocorram corretamente. A Figura 4.6 ilustra o processo de codificação destas soluções e posterior extração dos voos de drone. Em síntese,

Algorithm 1 Algoritmo Construtivo Randomizado para Rotas de Caminhão

Entrada: L_α Porcentagens para definição da LCR = $\{0.25, 0.5, 0.75, 1\}$ C Lista com os clientes a serem atendidos n_r Quantidade de rotas n_p Tamanho da população

```

1:  $P_0 := \emptyset$  ▷ Inicializa população
2: for  $n_{sol} := 0$  to  $n_p - 1$  do
3:    $LC := \emptyset$  ▷ Inicializa lista de candidatos
4:    $sol := \emptyset$  ▷ Inicializa uma nova solução
5:    $n_c := C.size$  ▷ Registra a quantidade de clientes a serem atendidos
6:   for  $r := 0$  to  $n_r$  do ▷ Para cada rota na solução
7:     for  $i := 0$  to  $n_c$  do ▷ Para cada cliente a ser atendido
8:        $custo := dist_M(0, C[i]) + dist_M(C[i], 0)$  ▷ Calcula custo de inserção
9:        $LC = LC \cup \langle r, C[i], custo, 0, 0 \rangle$  ▷ Inclui candidato na LC
10:    end for
11:  end for
12:  for  $r := 0$  to  $n_r$  do ▷ Para cada rota na solução
13:     $cliente := escolheAleatoriamente(LC)$  ▷ Escolhe um cliente aleatoriamente
14:     $sol \rightarrow incluiCliente(cliente)$  ▷ Inclui o cliente na rota
15:     $LC = atualizaLC(cliente)$  ▷ Atualiza LC com a remoção do candidato
16:  end for
17:  while há clientes não atendidos do
18:     $\alpha := escolheAleatoriamente(L_\alpha)$ 
19:     $N := \alpha \times LC.size$ 
20:     $LCR := selecionaCandidatos(LC, 0, N)$  ▷ LCR: primeiros N candidatos da LC
21:     $k := escolheAleatoriamente(LCR)$  ▷ Escolhe um candidato da LCR
22:     $sol \rightarrow incluiCliente(k)$  ▷ Inclui o cliente na rota escolhida
23:     $LC = atualizaLC(k)$  ▷ Atualiza LC com a remoção do candidato
24:  end while
25:   $sol \rightarrow codificaSolucao()$  ▷ Registra rota gigante
26:   $P_0 := P_0 \cup sol$  ▷ Adiciona solução à população inicial
27:   $n_{sol} = n_{sol} + 1$ 
28: end for

```

Saída: P_0 Conjunto de soluções da população inicial.

uma lista de identificadores é utilizada para registrar os primeiros clientes de cada uma das rotas criadas. Com isso, os clientes são inseridos de forma ordenada na “rota gigante” (que é o cromossomo do GA). Tendo a “rota gigante” corretamente formada, com a lista de primeiros clientes associada, o processo de decodificação das soluções é, essencialmente, a etapa de *Extraction* do procedimento introduzido por Prins, Labadi e Reghioui (2009).

Assim como na versão original, os voos de drone são posteriormente extraídos a partir da “rota gigante”, considerando a viabilidade da solução. A implementação do algoritmo segue o proposto por Zhang et al. (2022).

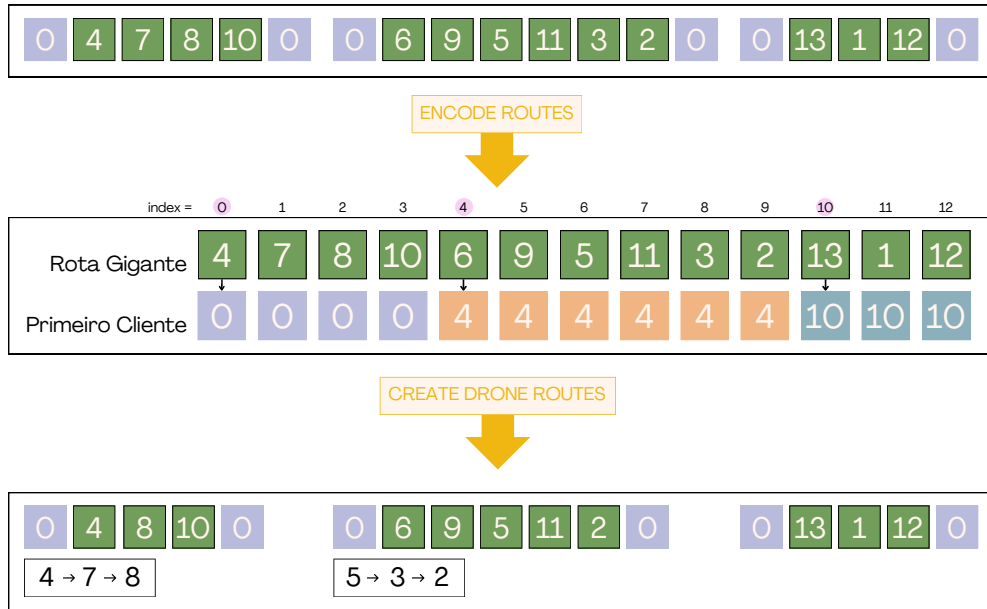


Figura 4.6: Processo de codificação das soluções geradas pelo novo Construtivo.

4.5 Operadores Genéticos e atualização da população

No procedimento padrão do NSGA-II, parte das soluções presentes na população P são submetidas a operadores genéticos, mais especificamente, cruzamentos e mutações. A intenção é que os melhores indivíduos sejam selecionados e combinados para que a população evolua, preservando as boas características das gerações anteriores.

No ENSGA-II e também neste trabalho, foram escolhidos dois operadores de

cruzamento e três operadores de mutação. As próximas seções deste capítulo descrevem em detalhes o comportamento de cada operador.

O processo de evolução da população segue da seguinte forma:

1. A população atual P é ordenada em fronteiras de Pareto, utilizando o *Fast Non-dominated Sorting Approach* (FNDS) de Deb et al. (2002). O algoritmo envolve classificar as soluções presentes na população em termos de dominância, utilizando um sistema de *ranking*. Para tanto, cada solução possui um contador que registra quantas soluções a dominam, e também uma lista de soluções dominadas. O algoritmo percorre e ordena as soluções a partir destes indicadores. Ao final da execução, soluções com *ranking* 0 compõem a primeira fronteira, ou seja, não são dominadas entre si, mas dominam todas as outras soluções. Quando não é possível comparar duas soluções quanto a seu *ranking*, por pertencerem à mesma fronteira, utiliza-se uma métrica de comparação denominada *crowding-distance*, que prioriza as soluções que se encontram em regiões menos populadas. A implementação eficiente dos algoritmos é fornecida pelos autores.
2. Para selecionar as soluções para a etapa de cruzamento, três indivíduos são selecionados aleatoriamente de P e participam de um torneio ternário. O melhor entre eles (com menor *ranking* ou, caso estejam na mesma fronteira, aquele com menor *crowding-distance* (DEB et al., 2002)) é escolhido como PAI #1. O mesmo se repete para a seleção do PAI #2. Estes indivíduos são então submetidos a cruzamentos, de acordo com a taxa de cruzamento definida. O operador de cruzamento é escolhido aleatoriamente, e cada par de pais gera dois filhos a partir desse processo.
3. Cada filho gerado é submetido a um operador de mutação, a depender da taxa de mutação definida. Assim como ocorre no cruzamento, a escolha do operador de mutação é aleatória.
4. Contanto que sejam viáveis, isto é, não violem as restrições do problema, os indivíduos gerados a partir dos operadores genéticos são incluídos em uma nova população G . O procedimento se repete até que G tenha o mesmo número de indivíduos de P .

5. As populações P e G são agrupadas em um grande conjunto de tamanho $2N$ (sendo N o tamanho da população original). Nesta etapa do procedimento, um novo módulo opcional de busca local foi introduzido neste trabalho, conforme apresentado na Seção 4.6. Caso esse módulo seja utilizado na execução, um novo grupo $\mathcal{M}$ com até 6 novas soluções mutantes, obtidas pela busca local multiobjetiva, é também acrescentado ao agrupamento.
6. As soluções agrupadas são submetidas novamente ao FNDS, e as N melhores soluções são mantidas para a próxima geração, como ilustra a Figura 4.7. Estes passos se repetem até que a última geração seja alcançada.

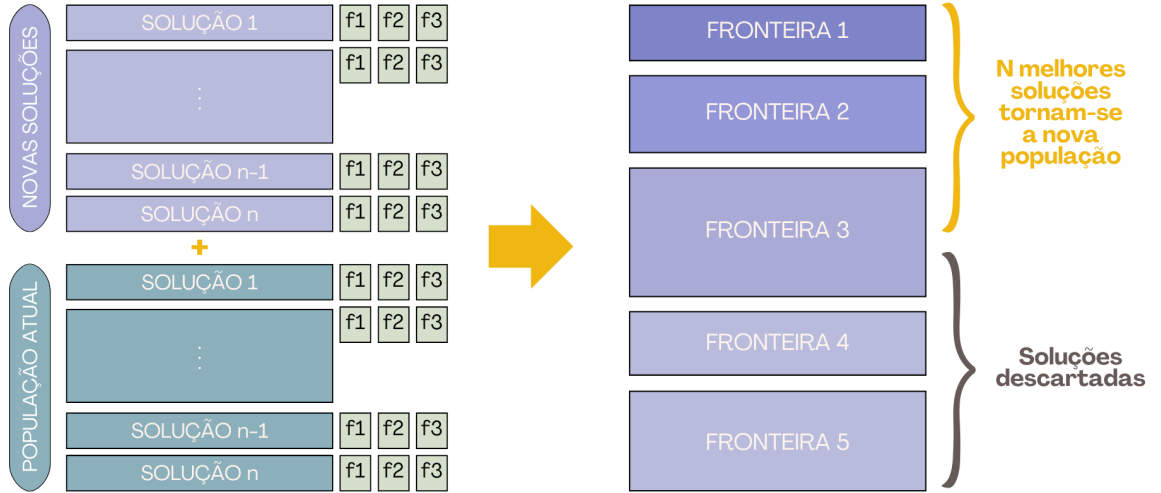


Figura 4.7: Procedimento do NSGA-II. Adaptado de (DEB et al., 2002).

4.5.1 Operadores de Cruzamento

Os operadores de cruzamento são utilizados para gerar novas soluções. Neste trabalho, utiliza-se o *Partially-Mapped Crossover* (PMX) e o *Order Crossover* (OX) (POTVIN, 1996). Para cada par de soluções selecionadas para cruzamento, uma taxa de cruzamento pré-definida indica a probabilidade de aplicar o operador de cruzamento, que é escolhido aleatoriamente entre os dois operadores utilizados.

Em ambos os cruzamentos, dois pontos de corte são determinados aleatoriamente, dividindo o cromossomo em três partes. O fragmento intermediário é inteiramente copiado de um dos pais para o filho.

Em seguida, no OX, as posições ainda não ocupadas no filho são preenchidas com os genes restantes, de acordo com a sequência em que aparecem no outro pai. A Figura 4.8(a) ilustra este processo. O fragmento que contém os clientes $\{10, 6, 9\}$ é copiado do PAI #1 para o FILHO. Em seguida, os clientes restantes são copiados na sequência em que aparecem no PAI #2.

Já no PMX, um mapeamento de correspondência entre os genes é utilizado para determinar o cruzamento, como exemplifica a Figura 4.8(b). Após o fragmento do PAI #1 ser copiado para o FILHO, é feito o mapeamento dos genes que ocupam cada posição desse fragmento no PAI #2: $10 \leftrightarrow 5, 6 \leftrightarrow 7, 9 \leftrightarrow 6$. A partir da primeira posição na sequência, a cópia direta de cada gene do PAI #2 para o FILHO é testada. O cliente 8 ainda não pertence ao FILHO, então é adicionado na primeira posição. O mesmo ocorre para os clientes 1 e 2. A verificação continua, agora, após a posição de corte. O cliente 10 já existe no FILHO, então o mapeamento deve ser consultado. O cliente 10 ocupa a quarta posição no PAI #1. No PAI #2, quem ocupa esta posição é o cliente 5. Logo, o 5 é copiado para o FILHO. Esse procedimento se repete até que todo o FILHO seja gerado.

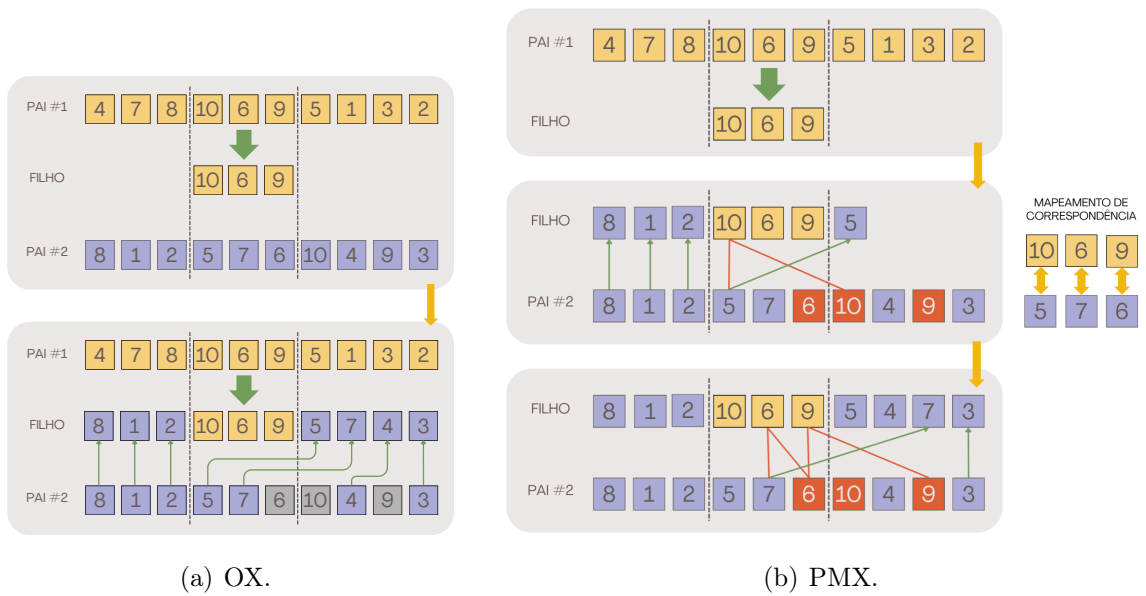


Figura 4.8: Operadores de Cruzamento.

4.5.2 Operadores de Mutação

Os operadores de mutação são aplicados em 20% dos indivíduos gerados pelos cruzamentos. A escolha entre *insert*, *swap* ou *reverse* é aleatória, e a função destes movimentos é introduzir maior diversidade à nova população, por meio de pequenas perturbações causadas aos indivíduos escolhidos. O operador *insert* elege um gene (cliente) para ser inserido logo após outro gene no cromossomo. O operador *swap*, por sua vez, troca dois genes de lugar. O operador *reverse*, por fim, elege dois pontos no cromossomo e inverte todos os genes contidos naquele fragmento. A Figura 4.9 exemplifica o funcionamento dos três operadores.

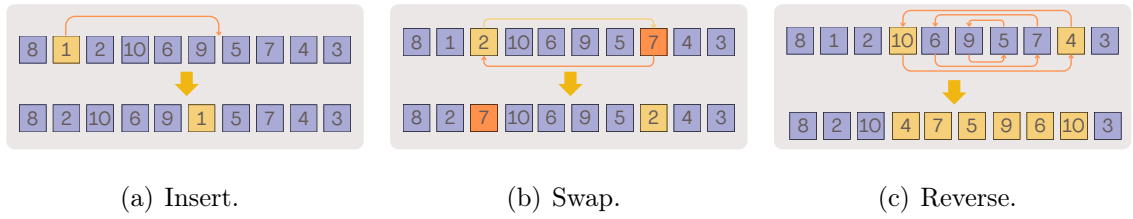


Figura 4.9: Operadores de Mutação.

4.6 Busca Local Multiobjetiva

Um novo módulo de busca local foi desenvolvido com base no *Massive Local Search Operator*, introduzido por Viana, Junior e Contreras (2020), com modificações para que a natureza multiobjetiva do MOVRPDD fosse considerada.

A cada nova geração, os melhores indivíduos da população atual P são selecionados para gerar novos indivíduos — através dos operadores de cruzamento e mutação apresentados na Seção 4.5 — que compõem uma nova população G . Esta população G é, então, submetida ao procedimento de Busca Local Multiobjetiva descrito no Algoritmo 2.

São escolhidas em G as duas melhores soluções quanto a cada um dos três objetivos e incluídas no conjunto B_{sol} . A opção por este método se deu pois, dessa forma, acredita-se que uma suposta estagnação em um mínimo local pode ser evitada. Além disso, evita-se que um objetivo seja privilegiado em relação aos outros.

Para a busca local, os mesmos movimentos utilizados na mutação são considerados

(*swap*, *insert* ou *reverse*). A cada chamada do algoritmo, um dos movimentos é escolhido aleatoriamente. Uma estrutura de laços aninhados itera sobre todos os pares de clientes da solução $sol \in B_{sol}$, aplicando o operador de mutação escolhido e gerando uma nova solução $novaSol$. Se $novaSol$ é melhor que sol (em relação ao objetivo), então $novaSol$ passa a ser utilizada como base para os movimentos seguintes. Ao fim da iteração, se houve melhora em relação ao objetivo atual, a solução gerada é incluída em uma nova população de soluções $\mathcal{M}$.

Por fim, as populações P , G e $\mathcal{M}$ são agrupadas e ordenadas em fronteiras segundo o FNDS padrão do NSGA-II, e as N melhores soluções deste agrupamento são utilizadas na próxima geração.

Algorithm 2 Procedimento de Busca Local Multiobjetiva

Entrada: G Grupo de indivíduos gerados pelos cruzamentos e mutações $\mathcal{O}_{mut}$ Conjunto de operadores de mutação = {swap, insert, reverse} $\mathcal{F}_{obj}$ Conjunto de funções-objetivo = $\{f_1, f_2, f_3\}$

```

1:  $\mathcal{M} := \emptyset$  ▷ Inicializa conjunto de soluções
2: for função objetivo  $f \in Fobj$  do ▷ Para cada função-objetivo  $f$ 
3:    $B_{sol} := \text{escolheSolucoes}(G, f, 2)$  ▷ Melhores duas soluções em  $G$  considerando  $f$ 
4:   for  $sol \in B_{sol}$  do ▷ Para cada solução
5:      $op := \text{escolheAleatoriamente}(\mathcal{M}_{mut})$  ▷ Escolhe um operador de mutação
6:      $melhorSol := sol$ 
7:     for  $i := 1$  to  $N - 1$  do ▷ Para cada combinação de clientes  $(i, j)$ 
8:       for  $j := i + 1$  to  $N$  do
9:          $novaSol := \text{aplicaMutacao}(melhorSol, i, j, op)$ 
10:        if  $novaSol \rightarrow f < melhorSol \rightarrow f$  then ▷ Se a solução foi melhorada
11:           $melhorSol := novaSol$  ▷ novaSol passa a ser utilizada
12:        end if
13:      end for
14:    end for
15:    if  $melhorSol \neq sol$  then ▷ Se uma solução melhor foi encontrada
16:       $\mathcal{M} = \mathcal{M} \cup melhorSol$  ▷ Adiciona novaSol ao conjunto de soluções mutantes
17:    end if
18:  end for
19: end for

```

Saída: $\mathcal{M}$ Conjunto de novas soluções mutantes.

5 Resultados Computacionais

Este trabalho introduz um novo método construtivo para as soluções que compõem a população inicial do GA. Além disso, a adição de um módulo de busca local multiobjetiva após a etapa de aplicação dos operadores genéticos é sugerida. Para avaliar o efeito destas adições ao procedimento tradicional do NSGA-II, foram realizados experimentos computacionais que comparam o impacto dessas abordagens quanto ao tempo de processamento e à qualidade das soluções obtidas.

A primeira seção deste capítulo descreve brevemente as instâncias e o ambiente computacional utilizado. Em seguida, os resultados dos experimentos conduzidos são apresentados e analisados.

5.1 Instâncias e Ambiente Experimental

Para avaliar as contribuições propostas são utilizadas 36 instâncias, geradas por Zhang et al. (2022) com o objetivo de refletir diversos cenários reais que podem ser encontrados em processos de entrega. Estas instâncias representam todos os tipos de combinações possíveis entre as características consideradas, sendo elas:

- Número de clientes atendidos: 40, 80, 120 ou 160;
- Raio da região de localização dos clientes: 20km, 30km ou 40km;
- Porcentagem de clientes que podem ser atendidos por drones: 30%, 50% ou 70%;

Dessa forma, uma instância que contém 40 clientes localizados em uma área com raio de 30km, dos quais 70% podem ser atendidos por drones, é nomeada “40 – 30 – 0.7”.

Os parâmetros envolvidos na execução também são os mesmos utilizados por Zhang et al. (2022), conforme apresentado na Tabela 5.1. Para todos os algoritmos avaliados, foram utilizadas 500 iterações, uma população de tamanho 80, e taxas de 80% e 20%, respectivamente, para cruzamentos e mutações.

Tabela 5.1: Parâmetros associados ao MOVRPDD

Parâmetro	Notação	Valor numérico
Peso do caminhão (vazio)	Wt	1500 kg
Capacidade máxima de carga do caminhão	Qt	1000 kg
Peso do drone (vazio)	Wd	25 kg
Capacidade máxima de carga do drone	Qd	5 kg
Custo de viagem do caminhão por un. de distância	Ct	25 un. monetárias / km
Custo de viagem do drone por un. de distância	Cd	1 un. monetária / km
Custo base de uso de um par caminhão-drone	Cb	500 un. monetárias
Autonomia máxima de voo de um drone vazio	E	0.5 h
Velocidade média de viagem do caminhão	St	60 km/h
Velocidade média de voo do drone	Sd	65 km/h

Os experimentos foram executados em um computador pessoal com processador Intel Core i5-7200U com uma unidade gráfica integrada Intel HD Graphics 620. A máquina possui 8 GB de memória RAM. O ambiente de software consistiu no sistema operacional Microsoft Windows 11 Home Single Language, que hospeda a distribuição Ubuntu 22.04.5 LTS através do Windows Subsystem for Linux (WSL2). Os algoritmos foram totalmente implementados em C++ e compilados com o G++ (versão 11.4.0), e os scripts de análise utilizaram Python 3.9.12.

5.2 Resultados Experimentais

Avaliar algoritmos que buscam resolver problemas de otimização multiobjetivo é uma tarefa complexa, pois não há uma única solução ótima a ser encontrada. Dessa forma, para além do comparativo entre o tempo de processamento das abordagens estudadas, a convergência e a diversidade das soluções no espaço dos objetivos devem ser examinadas.

Neste trabalho, três indicadores de performance consolidados pela literatura foram utilizados. São eles:

- *Inverted generational distance (IGD)*, introduzido por Zhang e Li (2007). Cada uma das abordagens é executada 15 vezes para a mesma instância, e as soluções não-dominadas obtidas após todas as execuções formam o conjunto adotado como fronteira verdadeira. IGD é o cálculo da distância média entre as soluções na fronteira verdadeira e cada solução não-dominada obtida pela abordagem. Um menor

IGD indica melhor desempenho, pois as soluções encontradas estão próximas e bem distribuídas ao longo da fronteira de referência.

- **Hypervolume (HV)**, introduzido por Zitzler e Thiele (2002). Esta métrica verifica o espaço dominado pelas soluções encontradas em relação a um ponto de referência, que deve ser dominado por todas as soluções no conjunto. Como os valores de cada função-objetivo foram normalizados entre 0 e 1, devido às suas grandezas diferentes, este trabalho adota o ponto $[1.1, 1.1, 1.1]$ como referência. Um valor de HV maior indica um conjunto de soluções melhor, pois representa um maior volume dominado no espaço dos objetivos.
- **C-metric**, ou *comparative metric*, também introduzida por Zitzler e Thiele (2002), é utilizada para comparar duas abordagens quanto à proporção de soluções dominadas. O valor de $C(A, B)$ indica a proporção de soluções encontradas por B que são dominadas por pelo menos uma solução encontrada por A . Dessa forma, se $C(A, B) = 1$, todas as soluções obtidas pelo algoritmo B são dominadas por alguma solução obtida pelo algoritmo A .

As tabelas 5.2 e 5.3 apresentam as médias dos resultados de IGD, HV e tempo de CPU para as instâncias com 40 e 80 clientes. Já as tabelas 5.4 e 5.5 comparam as abordagens estudadas quanto à *C-metric*. Cada abordagem foi executada 15 vezes para cada instância, de forma que a influência de *outliers* nos resultados fosse reduzida. São apresentados, respectivamente, os resultados obtidos pelas seguintes abordagens:

- CNSGA-II (C): ENSGA-II alterado para incluir a nova estratégia de construção da população inicial. Não utiliza o novo módulo de busca local;
- HNSGA-II (H): abordagem híbrida, que altera o ENSGA-II para incluir o novo algoritmo construtivo da população inicial e também o novo módulo de busca local multiobjetiva ao final de cada geração;
- ENSGA-II + BL (E): utiliza a abordagem construtiva da literatura (ENSGA-II) para determinar a população inicial e inclui o novo módulo de busca local multiobjetiva ao final de cada geração.

Tabela 5.2: Média dos resultados de IGD, HV e Tempo em instâncias com 40 clientes.

Instância	IGD			HV			Tempo de CPU (s)		
	C	H	E	C	H	E	C	H	E
40-20-0.3	0,0486	0,0082	0,0305	1,1883	1,2682	1,2425	11,9443	30,7513	28,8759
40-20-0.5	0,0569	0,0240	0,0373	1,1486	1,1766	1,2529	13,1856	38,9737	39,0886
40-20-0.7	0,0348	0,0407	0,0593	1,0649	0,9315	1,0235	11,9427	50,3697	50,4364
40-30-0.3	0,1375	0,0235	0,0652	1,0390	1,2379	1,1691	10,5499	43,5816	28,4826
40-30-0.5	0,0465	0,0292	0,0460	1,2200	1,2261	1,1740	11,5770	41,2884	37,1514
40-30-0.7	0,0363	0,0162	0,0742	1,0881	1,1219	1,0337	10,5184	57,5472	49,0762
40-40-0.3	0,0759	0,0183	0,0299	1,0452	1,1581	1,1177	2,4830	34,0583	30,3240
40-40-0.5	0,0124	0,0039	0,0003	1,1568	1,1915	1,2252	2,9589	48,7651	44,2472
40-40-0.7	0,0792	0,0462	0,0405	0,9176	1,0791	1,0922	3,1682	54,5447	52,4777

Tabela 5.3: Média dos resultados de IGD, HV e Tempo em instâncias com 80 clientes.

Instância	IGD			HV			Tempo de CPU (s)		
	C	H	E	C	H	E	C	H	E
80-20-0.3	0,3533	0,0020	0,0955	0,8118	1,3123	1,2042	17,6593	52,0328	434,3909
80-20-0.5	0,1759	0,0300	0,0489	0,9637	1,2782	1,2771	16,5698	401,3890	338,8976
80-20-0.7	0,0680	0,0049	0,0146	1,1342	1,2889	1,2903	16,3955	640,1835	577,1996
80-30-0.3	0,3493	0,0431	0,0068	0,6393	1,2190	1,2856	68,4578	275,1023	245,0094
80-30-0.5	0,1252	0,0330	0,0071	0,9592	1,2172	1,2855	16,4680	379,4828	334,7712
80-30-0.7	0,1773	0,0064	0,0466	0,9845	1,2983	1,2431	15,8272	667,8960	622,9857
80-40-0.3	0,3394	0,0188	0,0182	0,5478	1,1908	1,2239	16,1697	289,4405	250,3136
80-40-0.5	0,2099	0,0232	0,0254	0,8939	1,2551	1,1780	16,7715	374,1634	322,0368
80-40-0.7	0,0409	0,0089	0,0154	0,3629	0,4646	1,0385	16,7032	834,3381	713,7107

Os resultados apresentados nas Tabelas 5.2 e 5.3 demonstram que a busca local é efetiva em encontrar soluções melhores para praticamente todas as instâncias, visto que as duas abordagens híbridas (H e E) apresentam, em geral, melhor HV e IGD.

As soluções encontradas pelas abordagens que utilizam busca local (H e E) também são superiores em termos de dominância, quando comparadas às soluções da versão construtiva (C), como demonstram os resultados das Tabelas 5.4 e 5.5. Em especial, para as instâncias com 80 clientes, a busca local associada ao novo construtivo (H) superou 100% das soluções obtidas pelo construtivo sem busca local (C), assim como grande parte das soluções obtidas pela abordagem da literatura (E), o que indica que um bom conjunto de soluções iniciais também impacta positivamente a evolução das soluções em GAs.

Quanto ao tempo de execução, a utilização da busca local acrescenta, inevitavelmente, uma complexidade considerável aos algoritmos. Para instâncias menores, nota-se ainda que a inclusão do novo algoritmo construtivo não ocasionou impacto significativo no tempo de execução, quando comparamos ambas as abordagens híbridas (H e E), embora tenha aprimorado os resultados obtidos para grande parte das instâncias.

Tabela 5.4: Média dos resultados de C-metric em instâncias com 40 clientes.

Instância	CNSGA-II vs. HNSGA-II		ENSGA-II+BL vs. HNSGA-II		CNSGA-II vs. ENSGA-II+BL	
	C(C,H)	C(H,C)	C(E,H)	C(H,E)	C(C,E)	C(E,C)
40-20-0.3	0,4204	0,9990	0,5239	0,9244	0,5837	0,8570
40-20-0.5	0,4225	0,9477	0,6908	0,8935	0,5244	0,8645
40-20-0.7	0,7550	0,7683	0,6283	0,8595	0,5124	0,4333
40-30-0.3	0,1979	0,8925	0,6401	0,8436	0,1383	0,8722
40-30-0.5	0,3185	0,8591	0,8052	0,8983	0,4658	0,8591
40-30-0.7	0,7279	0,8583	0,7793	0,8967	0,8933	0,8667
40-40-0.3	0,3946	0,9050	0,5520	0,9210	0,4650	0,8546
40-40-0.5	0,6120	0,8037	0,6129	0,5508	0,5541	0,8037
40-40-0.7	0,7425	0,9217	0,8483	0,6217	0,5392	0,9542

Tabela 5.5: Média dos resultados de C-metric em instâncias com 80 clientes.

Instância	CNSGA-II vs. HNSGA-II		ENSGA-II+BL vs. HNSGA-II		CNSGA-II vs. ENSGA-II+BL	
	C(C,H)	C(H,C)	C(E,H)	C(H,E)	C(C,E)	C(E,C)
80-20-0.3	0,1297	1,0000	0,6907	0,9980	0,2317	1,0000
80-20-0.5	0,0520	1,0000	0,3839	0,9355	0,0998	0,9988
80-20-0.7	0,0373	1,0000	0,7027	0,9009	0,2243	0,9937
80-30-0.3	0,0136	1,0000	0,9129	0,7125	0,0999	1,0000
80-30-0.5	0,0605	1,0000	0,7968	0,5886	0,0657	1,0000
80-30-0.7	0,1462	1,0000	0,6315	0,9525	0,2623	0,9968
80-40-0.3	0,0518	1,0000	0,9159	0,7940	0,0123	1,0000
80-40-0.5	0,0800	1,0000	0,8911	0,8547	0,1901	1,0000
80-40-0.7	0,5950	0,9983	0,8000	0,9942	0,8608	1,0000

Para as instâncias de 120 e 160 clientes, para viabilizar a apresentação dos resultados, foram realizadas 5 execuções para cada abordagem, com número de gerações igual a 300.

Os resultados dos cálculos de HV e IGD para essas instâncias são apresentados, respectivamente, nas Tabelas 5.6 e 5.7, e indicam que a nova abordagem híbrida proposta (H) produz as melhores soluções para todas as instâncias grandes (com 120 ou 160 clientes), com exceção da instância “120-40-0.3”.

Conforme indica a Tabela 5.7, o HNSGA-II encontrou todas as soluções que compõem a fronteira verdadeira adotada em cinco das instâncias com 160 clientes, haja vista o IGD igual a zero. No entanto, é importante ressaltar que este resultado só foi obtido devido ao processo de construção da fronteira de referência que, para as instâncias grandes, foi formada pela união das soluções não dominadas entre aquelas obtidas ao longo das 15 execuções (5 para cada uma das 3 abordagens). Portanto, como o número

de execuções foi bastante limitado, a fronteira de Pareto de referência é relativamente pequena, o que aumenta a chance de coincidência exata entre soluções obtidas e soluções da referência.

Tabela 5.6: Média dos resultados de IGD, HV e Tempo em instâncias com 120 clientes.

Instâncias	IGD			HV			Tempo de CPU (s)		
	C	H	E	C	H	E	C	H	E
120-20-0.3	0,4915	0,0073	0,1191	0,2738	1,1580	1,0325	3,5757	398,3133	393,6770
120-20-0.5	0,2154	0,0091	0,1360	0,7962	1,2207	0,9877	4,2466	561,0986	543,0542
120-20-0.7	0,4706	0,0693	0,1710	0,4299	1,1442	0,5094	6,2418	1492,1901	1564,5842
120-30-0.3	0,5797	0,0119	0,1140	0,2402	1,1570	0,8716	3,9202	392,4895	394,2881
120-30-0.5	0,2361	0,0004	0,0823	0,6502	1,2134	1,0496	4,4021	534,6899	527,2181
120-30-0.7	0,0694	0,0024	0,0829	1,0274	1,2563	1,0654	5,4103	988,5507	1008,7280
120-40-0.3	0,8821	0,0454	0,0307	0,0939	1,1626	1,1557	4,1313	377,6650	376,5440
120-40-0.5	0,2760	0,0009	0,1351	0,7405	1,2702	1,0338	4,5433	583,1969	600,6802
120-40-0.7	0,1765	0,0019	0,0881	0,9179	1,2670	1,1154	5,6695	801,5799	847,6285

Tabela 5.7: Média dos resultados de IGD, HV e Tempo em instâncias com 160 clientes.

Instâncias	IGD			HV			Tempo de CPU (s)		
	C	H	E	C	H	E	C	H	E
160-20-0.3	0,2435	0,0388	0,0550	0,5631	1,0630	0,9611	10,2312	1696,8359	1374,3286
160-20-0.5	0,2111	0,0000	0,1904	0,9352	1,2817	1,0050	10,0456	2377,4734	2133,5714
160-20-0.7	0,0547	0,0032	0,1825	1,1388	1,2786	0,9718	12,2314	4068,9269	4087,6174
160-30-0.3	0,7218	0,0000	0,1434	0,1729	1,1825	0,8583	9,0896	1391,6327	1360,7231
160-30-0.5	0,2125	0,0000	0,1134	0,7037	1,2441	1,0309	9,9607	2202,0069	1713,4954
160-30-0.7	0,1149	0,0464	0,2535	1,0267	1,2326	0,7623	22,2143	3776,7015	3322,4633
160-40-0.3	0,6769	0,0000	0,1499	0,1632	1,1366	0,8825	13,7961	1341,4194	1287,3156
160-40-0.5	0,3380	0,0000	0,3006	0,5798	1,2025	0,6928	13,1929	1460,2905	1608,7799
160-40-0.7	0,0562	0,0051	0,1928	1,0410	1,2292	0,9468	13,0006	2942,9549	2640,4399

Do mesmo modo, o comparativo de dominância entre as soluções não-dominadas obtidas por cada algoritmo é apresentado na Tabela 5.8 para instâncias com 120 clientes, e na Tabela 5.9 para instâncias com 160 clientes. Assim como nas instâncias menores, observa-se que as abordagens híbridas (H e E) obtêm maior sucesso em encontrar soluções não-dominadas em quase todos os cenários.

5.3 Análise Experimental

Para além da análise das métricas de performance, os testes estatísticos de Friedman (FRIEDMAN, 1940) e Wilcoxon (WILCOXON, 1992) foram conduzidos para avaliar a superioridade da nova abordagem híbrida (HNSGA-II) em relação aos outros algoritmos

Tabela 5.8: Média dos resultados de C-metric em instâncias com 120 clientes.

Instância	CNSGA-II vs. HNSGA-II		ENSGA-II+BL vs. HNSGA-II		CNSGA-II vs. ENSGA-II+BL	
	C(C,H)	C(H,C)	C(E,H)	C(H,E)	C(C,E)	C(E,C)
120-20-0.3	0,0000	1,0000	0,3105	0,8942	0,0000	1,0000
120-20-0.5	0,0000	1,0000	0,0057	0,9762	0,1054	1,0000
120-20-0.7	0,0000	1,0000	0,2350	0,7975	0,2000	0,9975
120-30-0.3	0,0000	1,0000	0,1241	0,9500	0,0000	1,0000
120-30-0.5	0,0623	1,0000	0,5260	0,9916	0,0084	1,0000
120-30-0.7	0,1976	0,9443	0,1121	0,9965	0,6538	0,6690
120-40-0.3	0,0000	1,0000	0,8615	0,7022	0,0000	1,0000
120-40-0.5	0,0841	1,0000	0,4206	0,9840	0,2234	1,0000
120-40-0.7	0,0501	1,0000	0,0952	0,9486	0,3162	0,2745

Tabela 5.9: Média dos resultados de C-metric em instâncias com 160 clientes.

Instância	CNSGA-II vs. HNSGA-II		ENSGA-II+BL vs. HNSGA-II		CNSGA-II vs. ENSGA-II+BL	
	C(C,H)	C(H,C)	C(E,H)	C(H,E)	C(C,E)	C(E,C)
160-20-0.3	0,0000	1,0000	0,0956	0,8800	0,1829	1,0000
160-20-0.5	0,0769	1,0000	0,0115	1,0000	0,6277	0,2923
160-20-0.7	0,3107	0,9027	0,0537	1,0000	0,8969	0,1003
160-30-0.3	0,0000	1,0000	0,4143	1,0000	0,0000	1,0000
160-30-0.5	0,0657	1,0000	0,3045	1,0000	0,0268	1,0000
160-30-0.7	0,0000	1,0000	0,0000	0,9283	0,9013	0,0000
160-40-0.3	0,0000	1,0000	0,4079	1,0000	0,0000	1,0000
160-40-0.5	0,0000	1,0000	0,0000	1,0000	0,3571	0,0551
160-40-0.7	0,2724	0,9013	0,0272	1,0000	0,6741	0,0762

implementados neste trabalho. Uma taxa de significância de 5% foi utilizada, conforme a literatura.

Os resultados comparativos do teste de Wilcoxon entre o HNSGA-II e os outros algoritmos implementados são exibidos na Tabela 5.10 quanto aos três indicadores de performance apresentados na seção anterior. Este teste estatístico realiza comparações par a par, permitindo identificar quais algoritmos possuem uma vantagem estatisticamente significativa em relação aos demais. Nesse sentido, um resultado N_1/N_2 indica que o algoritmo HNSGA-II supera a outra abordagem de forma significativa em N_1 instâncias, mas não em N_2 delas.

A Tabela 5.11 apresenta os resultados estatísticos do teste de Friedman. O objetivo deste teste é classificar a performance das abordagens implementadas quanto às métricas HV e IGD. Conforme demonstra a tabela, o novo algoritmo construtivo associado ao módulo de busca local (HNSGA-II) é classificado em primeiro lugar, ou seja,

Tabela 5.10: Resultados comparativos do teste de Wilcoxon.

Indicador de performance	HNSGA-II vs. CNSGA-II	HNSGA-II vs. ENSGA-II+BL
HV	35 / 1	27 / 9
IGD	35 / 1	30 / 6
C-metric	36 / 0	29 / 7

apresenta a melhor performance. A outra abordagem que utiliza busca local (ENSGA-II + BL) ocupa o segundo lugar no *ranking*, seguida da abordagem que não possui busca local (CNSGA-II).

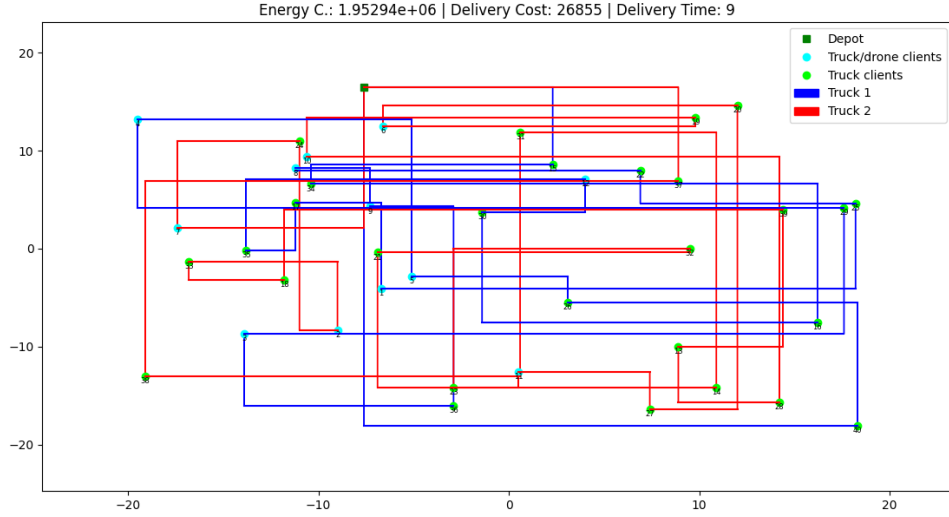
Tabela 5.11: Ranking médio dos algoritmos analisados, segundo o teste de Friedman.

HV	Média	Classif. final	IGD	Média	Classif. final
HNSGA-II	1.277778	1	HNSGA-II	1.194444	1
ENSGA-II	1.916667	2	ENSGA-II	2.000000	2
CNSGA-II	2.805556	3	CNSGA-II	2.805556	3

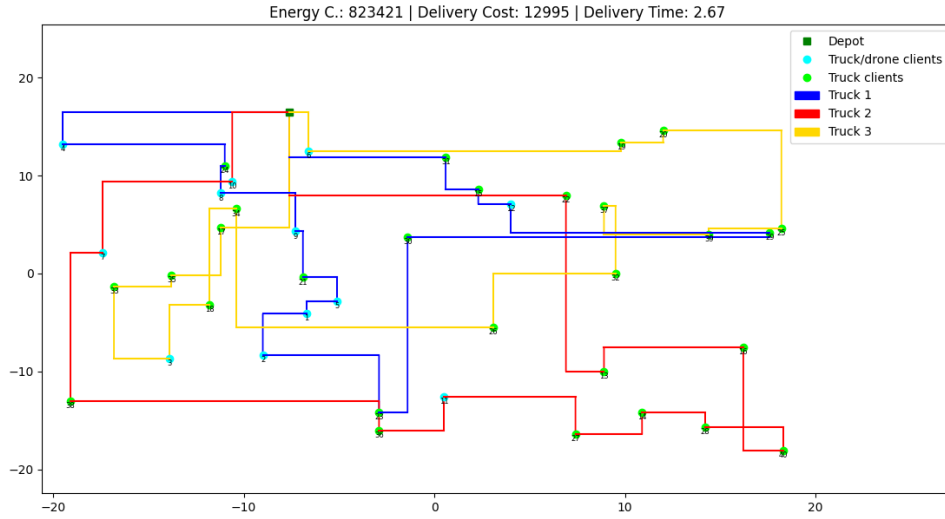
A Figura 5.1 ilustra o impacto da utilização do novo algoritmo construtivo na construção da população inicial. Os clientes estão dispostos no espaço de acordo com suas coordenadas (x, y) , e aqueles que podem ser atendidos por drones são demarcados em azul, enquanto os que só podem ser atendidos por caminhões são demarcados em verde.

Uma solução inicial obtida pelo *tour splitting* para a instância 40 – 20 – 0.3 é retratada em 5.1(a), onde duas rotas de caminhão-drone são definidas. Já na figura 5.1(b), uma solução obtida pelo novo construtivo é apresentada. Três rotas caminhão-drone são criadas e, mesmo assim, todos os objetivos são reduzidos em relação à solução da literatura. Ambas as imagens retratam soluções parciais obtidas após a execução da etapa construtiva da população inicial, ou seja, não há interferência de operadores genéticos ou da busca local nos resultados. A mesma *seed* foi utilizada nas duas execuções.

Um estudo foi conduzido com base em grafos que possibilitam uma análise visual da diversidade das soluções obtidas pelos algoritmos. A Figura 5.2 apresenta esses grafos para a instância 40 – 20 – 0.3. Cada nó no grafo representa um dos clientes da rota, posicionados no espaço de acordo com suas coordenadas (x, y) . Cada aresta (i, j) indica



(a) Versão da literatura.



(b) Novo construtivo.

Figura 5.1: Comparativo de uma solução inicial para a instância 40-20-0.3

se os clientes i e j aparecem de maneira consecutiva em alguma solução. Quanto mais próximo da cor amarela, mais soluções posicionam os clientes i e j para serem atendidos sequencialmente em alguma rota.

Os grafos das soluções obtidas pelo algoritmo que não utiliza busca local são apresentados na Figura 5.2(a), respectivamente, para as gerações 1, 50, 300 e 500. O mesmo ocorre na Figura 5.2(b), quanto às soluções obtidas com o auxílio da busca local.

Percebe-se que, embora o algoritmo híbrido seja capaz de reduzir a quantidade de possibilidades com menos gerações (visto que, já na geração 50, há uma notável diminuição da quantidade de arestas em relação à abordagem sem busca local), o espaço de busca é melhor explorado e as soluções finais apresentam maior diversidade na versão com busca

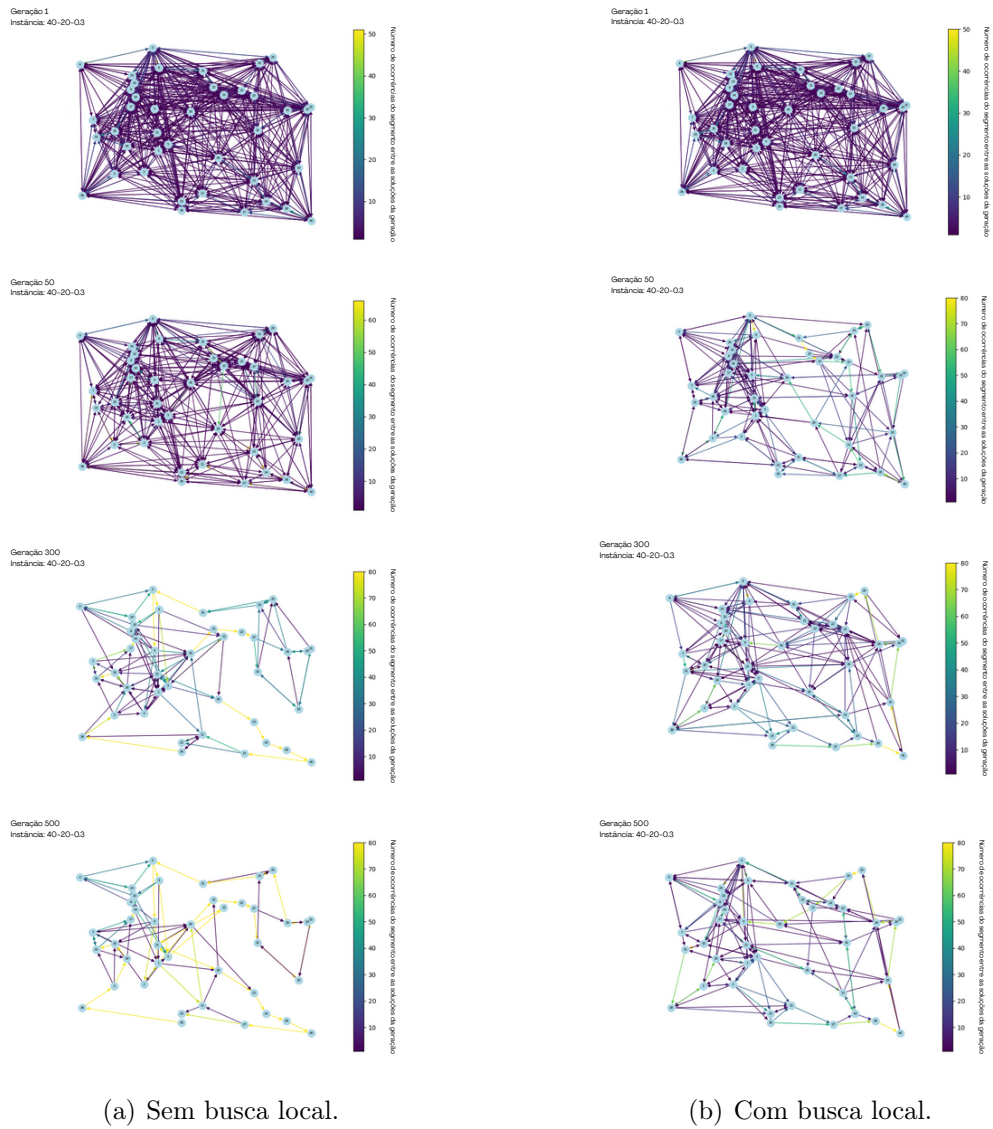


Figura 5.2: Comparativo da evolução das soluções para a instância 40-20-0.3

local, enquanto a versão que não utiliza o novo módulo tende a se estagnar em regiões que apresentam ótimos-locais.

6 Considerações finais

O presente trabalho envolveu a reimplementação completa do algoritmo genético NSGA-II em C++, com a inclusão de adaptações necessárias para o contexto do Problema Multi-objetivo de Roteamento de Veículos com Entrega por Drones (MOVRPDD). Uma nova etapa construtiva foi proposta, na qual as soluções que compõem a população inicial são criadas por meio de um algoritmo construtivo guloso randomizado que utiliza a heurística da Inserção Mais Barata para formar as rotas de caminhão de maneira balanceada. Um novo módulo de busca local foi inserido após a aplicação dos operadores genéticos no procedimento tradicional do NSGA-II, com o objetivo de aprimorar as soluções e prevenir a estagnação da população em mínimos locais.

Experimentos foram realizados para verificar a contribuição dos métodos propostos, por meio de métricas de desempenho de algoritmos multiobjetivos. Os resultados experimentais indicam que, embora computacionalmente cara, a inclusão da busca local é eficiente em encontrar um conjunto de soluções não-dominadas de melhor qualidade, especialmente quando combinada com o novo método construtivo proposto. Conclui-se, ainda, que a utilização de soluções iniciais mais especializadas pode acelerar a convergência algorítmica sem que a complexidade do algoritmo seja afetada.

Trabalhos futuros podem incluir a utilização de técnicas de computação paralela para otimizar o tempo de processamento ou, ainda, a adoção de uma estratégia dinâmica para acionamento do módulo de busca local em pontos estratégicos da execução do algoritmo genético. Dessa forma, ela poderá ser aplicada apenas nas gerações que estiverem se encaminhando para pontos de estagnação, reduzindo o tempo de processamento e, ainda assim, preservando (ou aprimorando) a qualidade das soluções obtidas. Neste sentido, outros operadores genéticos mais especializados para os objetivos do problema também podem ser introduzidos, de forma que as perturbações causadas pelos cruzamentos e mutações sejam ainda mais efetivas e, por consequência, acelerem a convergência algorítmica.

Referências Bibliográficas

- BEKTUR, G. A reinforcement learning-based multiobjective heuristic algorithm for multiple-truck routing problems with heterogeneous drones. *Applied Soft Computing*, v. 167, p. 112290, 2024. ISSN 1568-4946. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1568494624010640>>.
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, *Inform*s, v. 12, n. 4, p. 568–581, 1964.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. *Management science*, *Inform*s, v. 6, n. 1, p. 80–91, 1959.
- DAS, D. N. et al. Synchronized truck and drone routing in package delivery logistics. *IEEE Transactions on Intelligent Transportation Systems*, v. 22, 2021. ISSN 15580016.
- DEB, K. *Multi-Objective Optimization Using Evolutionary Algorithms*. USA: John Wiley & Sons, Inc., 2001. ISBN 047187339X.
- DEB, K. et al. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, 2002. ISSN 1089778X.
- DUAN, J.; HE, Z.; YEN, G. G. Robust multiobjective optimization for vehicle routing problem with time windows. *IEEE Transactions on Cybernetics*, v. 52, n. 8, p. 8300–8314, 2022.
- ERDOĞAN, S.; MILLER-HOOKS, E. A green vehicle routing problem. *Transportation research part E: logistics and transportation review*, Elsevier, v. 48, n. 1, p. 100–114, 2012.
- FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. *Journal of global optimization*, Springer, v. 6, n. 2, p. 109–133, 1995.
- FLOOD, M. M. The traveling-salesman problem. *Operations research*, *INFORMS*, v. 4, n. 1, p. 61–75, 1956.
- FRIEDMAN, M. A comparison of alternative tests of significance for the problem of m rankings. *The annals of mathematical statistics*, JSTOR, v. 11, n. 1, p. 86–92, 1940.
- GONZALEZ-R, P. L.; SANCHEZ-WELLS, D.; ANDRADE-PINEDA, J. L. A bi-criteria approach to the truck-multidrone routing problem. *Expert Systems with Applications*, v. 243, 2024. ISSN 09574174.
- HAN, Y. Q. et al. Metaheuristic algorithm for solving the multi-objective vehicle routing problem with time window and drones. *International Journal of Advanced Robotic Systems*, v. 17, 2020. ISSN 17298814.
- iFood. *Dos testes à autorização de voo: a história do drone do iFood*. 2022. Disponível em: <<https://institucional.ifood.com.br/inovacao/dos-testes-a-autorizacao-de-voo-a-historia-do-drone-do-ifood/>>.

Instituto de Energia e Meio Ambiente e Observatório do Clima. *Análise das emissões de gases de efeito estufa em 2023 e suas implicações para as metas climáticas do Brasil*. [S.l.], 2024. Publicação de novembro de 2024, com análise da evolução das emissões de GEE no Brasil de 1970 a 2023. Disponível em: <<https://energiaeambiente.org.br/produto/emissoesdegasesdeefeitoestufaem2023>>.

IPCC. *Climate Change 2023: Synthesis Report*. Intergovernmental Panel on Climate Change, 2023. Disponível em: <<https://www.ipcc.ch/report/sixth-assessment-report-cycle/>>.

JOZEFOWIEZ, N.; SEMET, F.; TALBI, E.-G. Multi-objective vehicle routing problems. *European journal of operational research*, Elsevier, v. 189, n. 2, p. 293–309, 2008.

KONG, F. et al. Collaborative delivery optimization with multiple drones via constrained hybrid pointer network. *IEEE Internet of Things Journal*, v. 11, 2024. ISSN 23274662.

LENSTRA, J. K.; KAN, A. R. Complexity of vehicle routing and scheduling problems. *Networks*, Wiley Online Library, v. 11, n. 2, p. 221–227, 1981.

LU, Y.; YANG, J.; YANG, C. A humanitarian vehicle routing problem synchronized with drones in time-varying weather conditions. *Computers and Industrial Engineering*, v. 184, 2023. ISSN 03608352.

LUO, Q. et al. Hybrid multi-objective optimization approach with pareto local search for collaborative truck-drone routing problems considering flexible time windows. *IEEE Transactions on Intelligent Transportation Systems*, v. 23, 2022. ISSN 15580016.

LUO, Q. et al. Multi-objective optimization algorithm with adaptive resource allocation for truck-drone collaborative delivery and pick-up services. *IEEE Transactions on Intelligent Transportation Systems*, v. 24, 2023. ISSN 15580016.

MADANI, B.; NDIAYE, M. Hybrid truck-drone delivery systems: A systematic literature review. *IEEE Access*, v. 10, p. 92854–92878, 2022.

MOMENI, M. et al. Coordinated routing system for fire detection by patrolling trucks with drones. *International Journal of Disaster Risk Reduction*, v. 73, p. 102859, 2022. ISSN 2212-4209. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2212420922000784>>.

NIKSERESHT, A.; GOLMOHAMMADI, D.; ZANDIEH, M. Sustainable green logistics and remanufacturing: a bibliometric analysis and future research directions. *The International Journal of Logistics Management*, Emerald Publishing Limited, v. 35, n. 3, p. 755–803, 2024.

Organização das Nações Unidas. *Transformando nosso mundo: a Agenda 2030 para o Desenvolvimento Sustentável*. 2015. Disponível em: <<https://brasil.un.org/pt-br/91863-agenda-2030-para-o-desenvolvimento-sustentavel>>. Acesso em: 05 jun. 2025.

POTVIN, J.-Y. Genetic algorithms for the traveling salesman problem. *Annals of operations Research*, Springer, v. 63, p. 337–370, 1996.

PRINS, C.; LABADI, N.; REGHIOUI, M. Tour splitting algorithms for vehicle routing problems. In: *International Journal of Production Research*. [S.l.: s.n.], 2009. v. 47. ISSN 00207543.

ROSENKRANTZ, D. J.; STEARNS, R. E.; LEWIS II, P. M. An Analysis of Several Heuristics for the Traveling Salesman Problem. *SIAM Journal on Computing*, SIAM, v. 6, n. 3, p. 563–581, 1977.

SABET, S.; FAROOQ, B. Green vehicle routing problem: State of the art and future directions. *IEEE Access*, v. 10, p. 101622–101642, 2022.

SHARMA, S.; KUMAR, V. A comprehensive review on multi-objective optimization techniques: Past, present and future. *Archives of Computational Methods in Engineering*, Springer, v. 29, n. 7, p. 5605–5633, 2022.

TAN, S.-Y.; YEH, W.-C. The vehicle routing problem: State-of-the-art classification and review. *Applied Sciences*, v. 11, n. 21, 2021. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/11/21/10295>>.

VIANA, M. S.; JUNIOR, O. M.; CONTRERAS, R. C. A modified genetic algorithm with local search strategies and multi-crossover operator for job shop scheduling problem. *Sensors*, MDPI, v. 20, n. 18, p. 5440, 2020.

WILCOXON, F. Individual comparisons by ranking methods. In: *Breakthroughs in statistics: Methodology and distribution*. [S.l.]: Springer, 1992. p. 196–202.

WOODWORTH, A. Retail delivery by drone. *Time*, out. 2024. Disponível em: <<https://time.com/7094858/wing/>>.

XIONG, H.; LEI, D.; LI, M. Multi-objective traveling salesman problem with drone: Imperialist competitive algorithm. In: *Proceedings of the 34th Chinese Control and Decision Conference, CCDC 2022*. [S.l.: s.n.], 2022.

YANG, F.; TAO, F. A bi-objective optimization vrp model for cold chain logistics: Enhancing cost efficiency and customer satisfaction. *IEEE Access*, IEEE, v. 11, p. 127043–127056, 2023.

YANG, Z. Food delivery by drone is just part of daily life in Shenzhen. *MIT Technology Review*, maio 2023. Disponível em: <<https://www.technologyreview.com/2023/05/23/1073500/drone-food-delivery-shenzhen-meituan/>>.

ZHANG, Q.; LI, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, IEEE, v. 11, n. 6, p. 712–731, 2007.

ZHANG, R. et al. A review on the truck and drone cooperative delivery problem. *Unmanned Systems*, v. 12, n. 05, p. 823–847, 2024. Disponível em: <<https://doi.org/10.1142/S2301385024300014>>.

ZHANG, S. et al. A novel multi-objective optimization model for the vehicle routing problem with drone delivery and dynamic flight endurance. *Computers and Industrial Engineering*, v. 173, 2022. ISSN 03608352. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0360835222006672>>.

ZITZLER, E.; THIELE, L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, IEEE, v. 3, n. 4, p. 257–271, 2002.