

Otimização de Rotas de Transporte com Múltiplos Drones: Proposta utilizando *Machine Learning*, Meta-Heurísticas e Solução Exata

Ricardo Vieira Chagas de Oliveira, Alexandre Ribeiro, Maria José Pereira Dantas

Escola Politécnica e de Artes – Pontifícia Universidade Católica de Goiás (PUC-GO)
Goiânia – GO – Brasil

ricardovieiracdo@gmail.com, {alexribeiro, mjdantas}@pucgoias.edu.br

Abstract. *The increasing utilization of drones for last-mile deliveries, motivated by the reduction of delivery time and maintenance costs, as well as by the mitigation of CO₂ emissions, demands robust solutions for the multiple drone transport route optimization problem (Covering Salesman Problem with Multiple Drones – CSPMD). This article proposes a multilayer solution to this problem, in an approach that integrates Machine Learning (ML), metaheuristic, and dynamic programming. The obtained results indicate that the use of specific metaheuristics, in the context of this multilayer framework, constitutes a promising strategy for solving the CSPMD. The performance evaluation of these metaheuristics indicated that the Genetic Algorithm outperformed in small instances, achieving a cost up to 0.49% lower, while Simulated Annealing proved superior in large instances, with a cost up to 1.93% lower.*

Resumo. *A crescente utilização de drones para entregas de última milha, motivada pela redução do tempo de entrega e dos custos de manutenção, bem como pela mitigação das emissões de CO₂, demanda soluções robustas para o problema de otimização de rotas de transporte com múltiplos drones (Covering Salesman Problem with Multiple Drones – CSPMD). O presente artigo propõe uma solução multifásica para este desafio, em uma abordagem que integra técnicas de aprendizado de máquina (Machine Learning - ML), meta-heurística e programação dinâmica. Os resultados obtidos apontam que o emprego de meta-heurísticas específicas, no contexto desta estrutura multifásica, constitui uma estratégia promissora para a resolução do CSPMD. A avaliação de desempenho dessas meta-heurísticas indicou que o Algoritmo Genético apresentou custo até 0.49% menor em instâncias pequenas, ao passo que o Simulated Annealing demonstrou custo até 1.93% menor em instâncias grandes.*

1. Introdução

Atualmente, as principais empresas especializadas na comercialização e distribuição de mercadorias, como a Amazon, Google e Walmart, estão direcionando recursos substanciais para o desenvolvimento de novas modalidades de transporte de produtos, especialmente para a última milha de entrega. Destaca-se, sobretudo, o investimento em veículos aéreos não tripulados, conhecidos como drones, para a realização dessas entregas. Neste contexto, a introdução de drones no transporte de mercadorias emerge como uma alternativa vantajosa. Esta medida não apenas contribui para a redução do tempo de entrega

e dos custos de manutenção, mas também para a mitigação das emissões de CO₂ associadas ao transporte terrestre [Yoo et al. 2018]. A importância desse aspecto é reforçada pela Organização das Nações Unidas [ONU 2018], que destaca que o setor de transporte é responsável por aproximadamente 25% das emissões globais de CO₂.

O problema de otimização de rotas de transporte com múltiplos drones (*Covering Salesman Problem with Multiple Drones – CSPMD*) pode ser formalmente modelado como um *Covering Salesman Problem* (CSP), o qual está classificado na categoria de problemas NP-Difícil [Song et al. 2022]. Nesse contexto, o problema consiste em um drone-mãe que segue uma rota e, ao longo do percurso, distribui drones menores que farão a cobertura de regiões específicas a fim de realizar a entrega a todos os clientes (pontos de interesse) dentro de suas respectivas áreas e, ao final de suas entregas, retornarão a pontos de recolhimento predefinidos para serem armazenados. O problema é modelado sobre um grafo completo $G = (V, E)$, onde $V = \{v_1, v_2, \dots, v_n\}$ é o conjunto de vértices, e E é o conjunto de arestas. Considerando que a cada aresta (v_i, v_j) do grafo está associado um custo c_{ij} , o objetivo é determinar um circuito sobre um subconjunto de vértices $V' \subseteq V$ que minimize uma função de custo composta. Essa função de custo é a combinação de duas partes: o custo da rota principal, definida por V' e percorrida pelo drone-mãe, e o custo agregado da cobertura de cada região específica [Wen and Wu 2022].

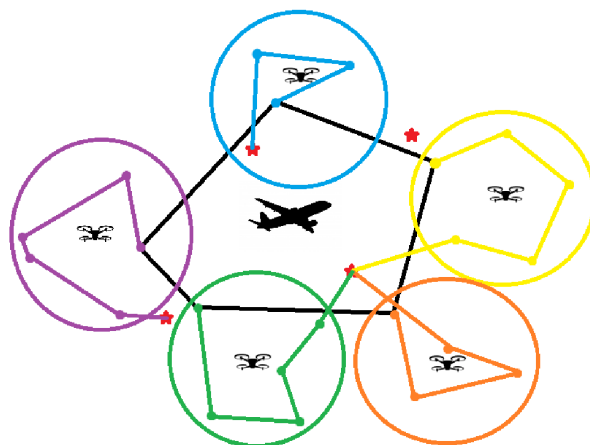


Figura 1. Ilustração do CSPMD.

A Figura 1 ilustra a estrutura do problema, onde as linhas pretas representam a trajetória percorrida pelo drone-mãe, as áreas coloridas indicam as regiões a serem cobertas por cada um dos drones menores, as linhas coloridas determinam a trajetória de cada drone menor, e as estrelas vermelhas designam os pontos de recolhimento.

Dentre as abordagens para a resolução desse problema, destaca-se o algoritmo IO-3S, proposto por [Wen and Wu 2022], que apresenta uma solução multifásica em três estágios que integra *Machine Learning* (ML), meta-heurística e programação dinâmica. A primeira etapa do processo utiliza ML para realizar a clusterização das regiões de interesse. Esse procedimento agrupa os pontos de interesse em conjuntos geograficamente coesos. O objetivo é que cada *cluster* seja eficientemente atendido por um drone menor, lançado durante a rota do drone-mãe. O segundo estágio, por sua vez, recorre a uma meta-heurística para a otimização da rota principal. Este é um desafio de natureza dual.

Ele envolve tanto a determinação da sequência de visitação dos *clusters*, um problema análogo ao do Caixeiro Viajante (*Traveling Salesman Problem* – TSP), quanto a seleção do ponto de interesse ótimo dentro de cada um deles. O objetivo final de todo o processo é minimizar o custo total do percurso. Por fim, o terceiro estágio recorre à programação dinâmica para calcular os custos exatos das sub-rotas, determinando, para cada *cluster*, o custo do percurso que um drone menor deve realizar para cobrir todos os pontos de interesse contidos em sua respectiva região designada.

Ademais, este trabalho se diferencia da abordagem de [Wen and Wu 2022], que emprega meta-heurísticas para selecionar os pontos de interesse em cada *cluster* e, posteriormente, para encontrar a rota entre eles. A presente pesquisa propõe uma abordagem metodológica centrada no uso de medoides [Kaufman and Rousseeuw 1987] — definidos como os objetos mais centrais de um *cluster* — para determinar os pontos de interesse que compõem o percurso do drone-mãe. Esta estratégia cumpre um duplo objetivo: primeiramente, reduz a complexidade de implementação da segunda fase do algoritmo, ao isolar a tarefa de seleção de pontos da otimização da sequência; em segundo lugar, provê pontos robustos que poderão ser utilizados no percurso do drone-mãe.

O presente artigo está estruturado da seguinte forma: a Seção 2 apresenta a fundamentação teórica e os trabalhos relacionados. A Seção 3 detalha a metodologia de pesquisa empregada, seguida pela apresentação e discussão dos resultados na Seção 4. Finalmente, a Seção 5 expõe as conclusões deste estudo.

2. Fundamentação Teórica e Trabalhos Relacionados

Meta-heurísticas como o Algoritmo Genético (*Genetic Algorithm* – GA) e o *Simulated Annealing* (SA) são amplamente utilizadas em publicações científicas para resolver problemas relacionados ao TSP. O GA é a abordagem com o maior número de publicações sobre o tema, e o SA ocupa a terceira posição nesse *ranking*. Tal fato evidencia a proeminência de ambas as técnicas na literatura especializada [Toaza and Esztergár-Kiss 2023].

O trabalho de [Bezdek et al. 1984] apresenta a fundamentação do algoritmo de clusterização *Fuzzy C-Means* (FCM). Métodos de análise tradicionais geram *clusters* rígidos (*hard clusters*). Neles, cada ponto de dado pertence inequivocamente a um único *cluster*. Em oposição a isso, o FCM produz *clusters fuzzy*. A essência desta abordagem é o uso de uma função de pertinência (*membership function*). Essa função atribui a cada objeto (ponto de interesse) um grau de associação parcial com múltiplos *clusters*. Isso viabiliza a otimização de uma função objetivo de mínimos quadrados. Este processo de otimização é executado de forma iterativa, continuando até que um critério de convergência seja atingido. Essa flexibilidade no pertencimento torna o algoritmo mais robusto a ruídos e *outliers*. Isso ocorre porque objetos (pontos de interesse) anômalos recebem baixos graus de pertinência. Consequentemente, eles exercem menor influência no resultado da clusterização.

O trabalho de [Kaufman and Rousseeuw 1987] introduz o conceito de medoide, que é central para algoritmos de clusterização robustos. A característica fundamental de um medoide é que ele seja um membro real do conjunto de dados. Tal propriedade torna o algoritmo de clusterização por medoides inerentemente mais robusto a *outliers*. Esse princípio é a base do algoritmo *Partitioning Around Medoids* (PAM). O objetivo do PAM

é encontrar um conjunto de k medoides. Estes medoides devem minimizar a soma das dissimilaridades de cada objeto (ponto de interesse) ao seu representante mais próximo. No presente trabalho, este conceito é aplicado a um problema de roteamento. Aqui, os medoides são usados como critério para selecionar os pontos de interesse que irão compor a rota do drone-mãe.

A programação dinâmica, embora computacionalmente intensiva, revela-se uma abordagem viável para a otimização de instâncias menores [Wen and Wu 2022]. Neste artigo, sua aplicação é direcionada à obtenção da solução exata para as sub-rotas nas regiões de interesse, cuja estrutura é visualmente representada pelas áreas coloridas na Figura 1.

3. Metodologia de Pesquisa

3.1. Base de Dados

A Tabela 1, desenvolvida com base em [Wen and Wu 2022], especifica os parâmetros para a construção das instâncias de teste. São detalhados o número de clientes (pontos de interesse) e as dimensões do plano cartesiano que definem o espaço do problema, uma vez que este é modelado como um conjunto de pontos em um ambiente bidimensional. Cabe ressaltar que, para todas as instâncias, foram definidos aleatoriamente dois atributos: a localização de cada cliente (ponto de interesse) e o peso do respectivo pacote, cujos valores foram gerados dentro do intervalo de 0.4 kg a 2.0 kg. Adicionalmente, cada cliente possui uma demanda unitária e é atendido uma única vez.

Tabela 1. Descrição das instâncias de teste.

Nome da Instância	Número de Clientes	Área de Distribuição
Pequena	40 clientes	6 km x 6 km
Média 1	60 clientes	8 km x 8 km
Média 2	80 clientes	10 km x 10 km
Grande	100 clientes	12 km x 12 km

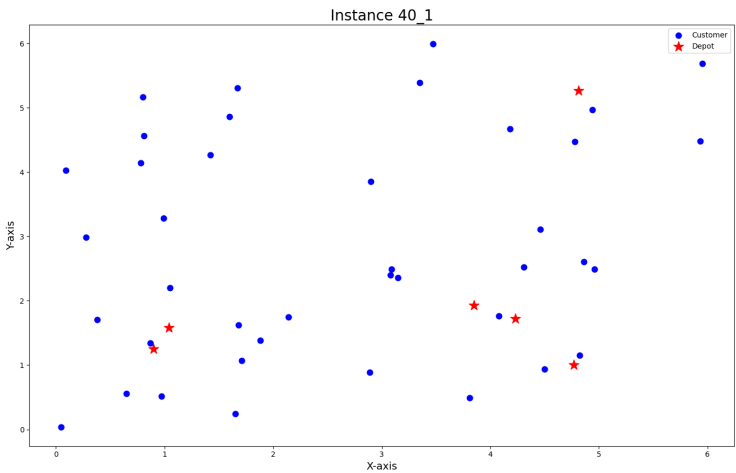


Figura 2. Instância 40_1.

Na Figura 2, é exibido um exemplo de uma instância pequena, detalhando a disposição dos pontos no plano.

3.2. Algoritmo

Esta seção detalha o algoritmo proposto, cuja estrutura em três fases é análoga à de [Wen and Wu 2022]. A primeira fase consiste na clusterização das regiões de interesse. A segunda fase é dedicada à determinação da rota do drone-mãe, utilizando medoides para selecionar um ponto de interesse em cada *cluster* e um algoritmo meta-heurístico (GA/SA) para definir a sequência ótima de visitação. Por fim, a terceira fase emprega programação dinâmica para obter a solução exata da rota dentro de cada região de interesse.

Ademais, as restrições para um drone menor foram definidas, especificando um número máximo de 6 clientes (pontos de interesse) atendíveis, uma capacidade máxima de carga de 10 kg e uma autonomia de voo (distância máxima percorrível) de 18 km.

3.2.1. Clusterização

A parametrização do algoritmo FCM foi projetada para uma exploração adaptativa do espaço de soluções. Para tanto, o número de *clusters* foi determinado dinamicamente por meio de um processo iterativo, assegurando que a solução resultante satisfizesse integralmente as restrições operacionais definidas para os drones menores.

Para os demais hiperparâmetros do FCM, adotaram-se valores padrões consolidados na literatura. O expoente de *fuzzificação* foi fixado em $m = 2$. O critério de convergência do algoritmo foi definido por $\epsilon = 0.005$ e um número máximo de 1000 iterações. A fim de assegurar a exploração de diferentes inicializações, uma semente pseudoaleatória foi empregada em cada execução.

A Figura 3 ilustra a partição em *clusters* resultante da aplicação do algoritmo sobre uma instância de teste, representada na Figura 2, em que cada cor distinta representa um *cluster*.

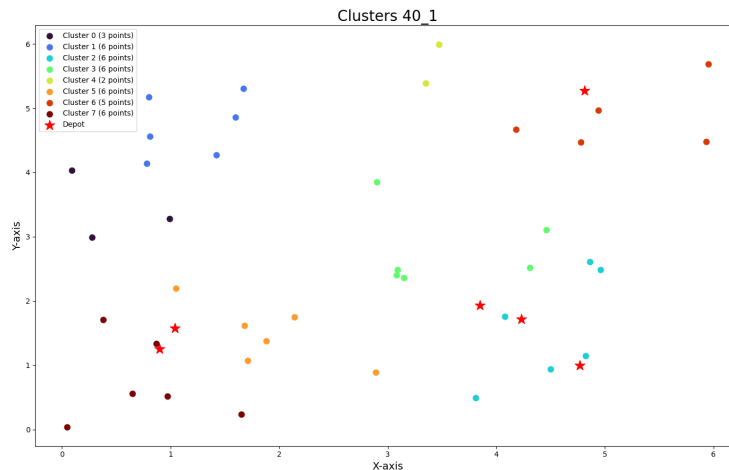


Figura 3. Clusterização da instância 40_1.

3.2.2. Rota do drone-mãe

A Figura 4 apresenta o caminho do drone-mãe em uma instância pequena, ilustrando uma solução obtida.

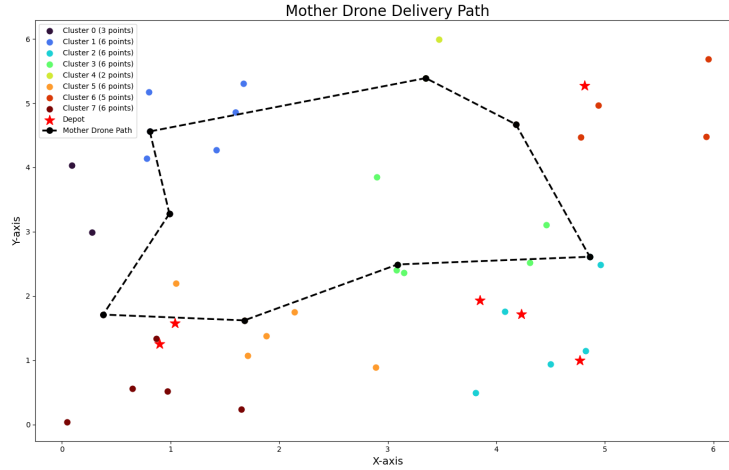


Figura 4. Rota do drone-mãe da instância 40_1.

A principal contribuição e o ponto central de diferenciação deste trabalho residem na abordagem meta-heurística adotada para a segunda fase do algoritmo. De forma análoga ao estudo de [Wen and Wu 2022], meta-heurísticas são empregadas para a otimização da rota do drone-mãe. Contudo, a metodologia aqui proposta diverge na etapa de seleção dos pontos de parada em cada *cluster*. Enquanto o trabalho de referência utiliza uma solução meta-heurística para ambas as tarefas (seleção de pontos e roteirização), a presente abordagem propõe o uso de medoides para a seleção dos pontos, reservando as meta-heurísticas apenas para a otimização da sequência de visitação.

Para a construção da rota, o método proposto utiliza, inicialmente, os medoides de cada *cluster*. A fim de promover a diversidade e explorar um espaço de busca mais amplo, foi introduzido um componente estocástico: para cada *cluster*, há uma probabilidade de 50% de que seu medoide seja substituído por outro ponto selecionado aleatoriamente dentro do mesmo *cluster*. Para garantir a formação de um conjunto diversificado de soluções candidatas, o procedimento de geração foi iterado 200 vezes. As soluções resultantes constituíram a população inicial para a fase de otimização da rota, conduzida pelos algoritmos GA e SA.

Para o GA, foi utilizada uma população de 100 indivíduos que evolui ao longo de 500 gerações. A seleção combina um elitismo que preserva as 20 melhores rotas com o método da roleta viciada para os demais indivíduos. A criação de novas soluções é feita pelo operador *Ordered Crossover* (OX), e a diversidade é mantida por uma taxa de mutação de 1%, que realiza trocas aleatórias de cidades (mutação por *swap*), buscando um equilíbrio entre a exploração do espaço de busca e a convergência para boas soluções.

Para o SA, a geração de rotas vizinhas foi realizada pelo método de perturbação *two-opt*. O algoritmo partiu de uma temperatura inicial calculada dinamicamente para se adequar ao problema e aplicou um fator de resfriamento (*alfa*) de 0.9 para reduzir gradu-

almente a probabilidade de aceitar soluções piores. O critério de parada principal foi a ausência de melhorias na melhor rota encontrada por 3 ciclos consecutivos de temperatura, permitindo uma busca ampla no início e um refinamento focado ao final do processo.

3.2.3. Rota do drone menor

A solução exata para o subproblema é obtida por meio de um algoritmo de programação dinâmica. A construção da solução ótima é realizada de forma iterativa, utilizando uma relação de recorrência que calcula o custo de um caminho maior a partir de soluções de subproblemas já computados. A relação é definida como

$$C(S, j) = \min_{i \in S, i \neq j} \{C(S \setminus \{j\}, i) + d(i, j)\}$$

Onde $C(S, j)$ representa o custo do caminho mais curto que visita todos os pontos do conjunto S e termina no ponto j . Cabe ressaltar que, no contexto desse artigo, o ponto de origem para esse cálculo corresponde ao medoide determinado para o respectivo *cluster*.

3.3. Complexidade

A complexidade computacional total é a soma da complexidade das três etapas do algoritmo: a clusterização inicial via FCM; o roteamento do drone-mãe a partir de medoides e de uma meta-heurística; e o roteamento dos drones menores via programação dinâmica, somando o custo de cada *cluster* individualmente.

3.3.1. Definição das Variáveis

- n : Número total de clientes (pontos de interesse).
- c : Número de *clusters*.
- d : Número de dimensões dos dados.
- i : Número de iterações do FCM.
- g : Número de gerações (GA).
- p : Tamanho da população (GA).
- x : Número de etapas de temperatura (SA).
- n_j : Número de clientes (pontos de interesse) no *cluster* j .

3.3.2. Fórmulas de Complexidade

- Clusterização: $O(nc^2di)$ [Pratik et al. 2024]
- Medoides: $\sum_{j=1}^c O(n_j^2)$ [Kaufman and Rousseeuw 1987]
- Rota do drone-mãe (GA): $O(g \cdot (p \cdot c^2 + p^2))$ [Andrei and Pietro 2019]
- Rota do drone-mãe (SA): $O(x \cdot c^2)$ [Avrim et al. 2021]
- Rotas dos drones menores: $\sum_{j=1}^c O(n_j^2 \cdot 2^{n_j})$ [Paul et al. 2018]

4. Resultados e Discussão

A análise indica que o GA foi superior em instâncias menores, enquanto o SA se mostrou superior nas maiores. Para garantir a robustez estatística, cada resultado corresponde à média de cinco execuções independentes.

Nesse contexto, as Tabelas 2 e 3 apresentam os resultados consolidados para o algoritmo proposto, considerando o roteamento do drone-mãe via GA e via SA, respectivamente. As métricas de desempenho analisadas incluem o custo médio da solução, o número médio de *clusters* e o tempo médio de execução computacional.

Tabela 2. Resultados obtidos ao utilizar GA.

Instância	Custo Médio	Média de <i>Clusters</i>	Tempo Médio de Execução
Pequena	49.33 km	8.00	4 minutos e 21.60 segundos
Média 1	80.59 km	13.00	5 minutos e 22.43 segundos
Média 2	112.57 km	18.00	7 minutos e 15.66 segundos
Grande	171.35 km	30.20	9 minutos e 17.80 segundos

Tabela 3. Resultados obtidos ao utilizar SA.

Instância	Custo Médio	Média de <i>Clusters</i>	Tempo Médio de Execução
Pequena	49.57 km	8.00	51.80 segundos
Média 1	80.94 km	13.00	57.41 segundos
Média 2	111.24 km	18.00	1 minuto e 6.30 segundos
Grande	168.04 km	30.20	1 minuto e 40.80 segundos

Cabe ressaltar que, a avaliação de desempenho dos algoritmos foi realizada em um sistema cujas especificações de *hardware* e *software* são: AMD Ryzen 3 7320U (2.4GHz), 8 GB RAM, Windows 11 Home.

No que tange à qualidade da solução, a análise do custo médio revela que o desempenho superior depende da escala da instância. Para cenários de menor porte, o GA apresentou uma ligeira vantagem, com um custo 0.49% inferior na instância “Pequena” e 0.43% inferior na “Média 1”. Contudo, a tendência se inverte a favor do SA em problemas de maior complexidade: na instância “Média 2”, o SA obteve um custo 1.18% menor que o GA, e essa vantagem se tornou mais expressiva na instância “Grande”, onde alcançou um custo 1.93% inferior. Em termos de desempenho, o SA demonstra superioridade em instâncias de alta complexidade, consolidando-se como a abordagem preferencial para otimizações em larga escala, enquanto o GA revela-se mais indicado para instâncias de menor escopo.

Em termos de eficiência, o SA também se destacou ao apresentar tempos de execução médios significativamente inferiores. Observou-se também que, para cada instância, as execuções com menor número de clusters apresentaram, consistentemente, os custos totais mais baixos para ambos os algoritmos (GA e SA).

Adicionalmente, a Figura 5 ilustra a configuração da solução final obtida por meio do processo de otimização. A representação visual detalha a estrutura logística, que é composta por múltiplos *clusters* de entrega. Dentro de cada *cluster*, as rotas individuais

dos drones menores são demarcadas por cores distintas, facilitando a visualização e a distinção de cada trajetória. Todas as rotas convergem para pontos de coleta designados, indicados na figura por marcadores em forma de estrela. Ademais, a trajetória do drone-mãe é tracejada e conecta os diferentes *clusters* finalizando a operação logística.

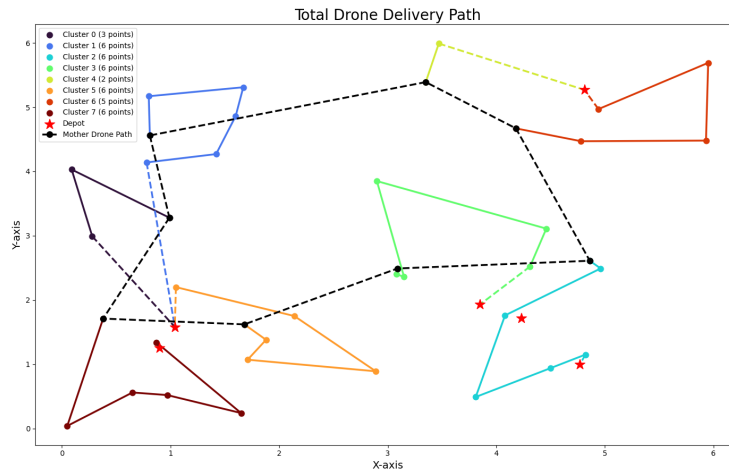


Figura 5. Rotas de todos os drones da instância 40_1.

A Figura 6 avalia a eficácia do algoritmo proposto por meio de uma análise comparativa, confrontando o desempenho do algoritmo quando o roteamento do drone-mãe é gerado a partir do SA e do GA. O gráfico expõe a média dos resultados obtidos a partir de cinco execuções para cada um dos quatro tipos de instância de problema testados. A métrica de avaliação utilizada para a comparação é a distância total percorrida, que corresponde à soma da distância percorrida pelo drone-mãe e pelas distâncias agregadas de todos os drones que operam dentro dos *clusters*.

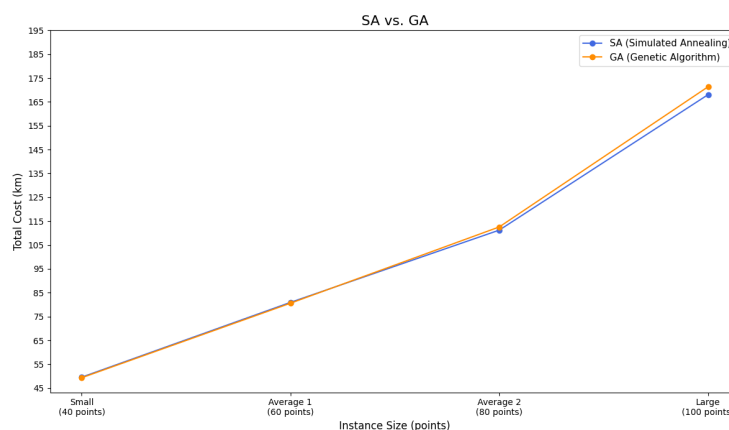


Figura 6. Comparação SA vs. GA.

5. Conclusão

O objetivo deste trabalho foi avaliar o desempenho de diferentes meta-heurísticas na resolução de um problema de otimização, comparando-as de forma direta para deter-

minar a abordagem mais eficaz. Para esta análise, foram selecionados os algoritmos GA e SA, cuja notória eficiência é corroborada por estudos anteriores, como o de [Toaza and Esztergár-Kiss 2023]. A experimentação em múltiplas instâncias revelou que a superioridade de cada algoritmo está diretamente relacionada à escala do problema. O GA demonstrou maior eficácia em cenários de menor porte, enquanto o SA obteve um desempenho consistentemente superior em instâncias de maior complexidade.

Para trabalhos futuros, propõe-se estender a abordagem atual em três áreas principais. Primeiramente, validar o método com dados mais complexos e reais. Em segundo lugar, refinar o modelo para incluir os custos monetários individualizados por drone. Por fim, explorar outras meta-heurísticas e implementar otimizações de desempenho, como a paralelização, a fim de diminuir o tempo de execução e elevar a eficiência do algoritmo.

Referências

- Andrei, L. and Pietro, S. O. (2019). Computational complexity analysis of genetic programming. Accessed on: 2025-11-13.
- Avrim, B., Chen, D., and Saeed, S. (2021). Learning complexity of simulated annealing. Accessed on: 2025-11-13.
- Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*. Accessed on: 2025-08-21.
- Kaufman, L. and Rousseeuw, P. J. (1987). Clustering by means of medoids. *Journal of Computational and Applied Mathematics*. Accessed on: 2025-08-20.
- ONU (2018). Transporte sustentável é destaque na conferência do clima da onu. Technical report. Accessed on: 2025-11-13.
- Paul, B., Niels, A., and Marie, S. (2018). Dynamic programming approaches for the traveling salesman problem with drone. Accessed on: 2025-11-13.
- Pratik, S. T., Rohit, K. V., and Rakesh, T. (2024). Analysis of time complexity of k-means and fuzzy c-means clustering algorithm. *Engineering Mathematics Letters*. Accessed on: 2025-11-13.
- Song, H., Jingi, A. M., and Yang, X. (2022). Drone-assisted last-mile delivery problem by covering salesman problem with nodes and segments. *SSRN Electronic Journal*. Accessed on: 2024-04-14.
- Toaza, B. and Esztergár-Kiss, D. (2023). A review of metaheuristic algorithms for solving tsp-based scheduling optimization problems. *Applied Soft Computing*. Accessed on: 2025-08-20.
- Wen, X. and Wu, G. (2022). Heterogenous multi-drone routing problem for parcel delivery. *Computers & Operations Research*. Accessed on: 2025-02-01.
- Yoo, W., Yu, E., and Jung, J. (2018). Drone delivery: Factors affecting the public's attitude and intention to adopt. *Telematics and Informatics*. Accessed on: 2024-04-14.