

ALGORITMO GENÉTICO APLICADO À LOCALIZAÇÃO DE INSTALAÇÕES CAPACITADAS COBERTAS POR DRONES DE ALCANCE LIMITADO

Tarcísio Barroso Marques

Departamento de Informática - Universidade Federal de Viçosa
Av. P H Rolfs, s/n - Campus Universitário, Viçosa - MG, 36570-900
tarcisio.marques@ufv.br

José Elias Claudio Arroyo

Departamento de Informática - Universidade Federal de Viçosa
Av. P H Rolfs, s/n - Campus Universitário, Viçosa - MG, 36570-900
jarroyo@ufv.br

RESUMO

Este artigo aborda o problema de Localização de Instalações Capacitadas de Máxima Cobertura com Drones de Alcance Limitado (LICMCD). Considerando n clientes com demandas, m locais candidatos para instalações e um conjunto de drones, o objetivo é determinar p locais para lançamento de drones e distribuição de recursos, maximizando a cobertura. Devido à alta complexidade computacional do LICMCD, propõe-se um Algoritmo Genético (AG) com heurística construtiva inspirada na metaheurística GRASP para a população inicial. O AG foi testado com dados da literatura e comparado com heurísticas e um modelo matemático de programação linear inteira, apresentando resultados muito satisfatórios. Diversos cenários foram analisados para avaliar a importância da quantidade de drones e do número de instalações capacitadas.

PALAVRAS CHAVE. Localização de facilidades. Máxima Cobertura. Algoritmo Genético.
Tópicos: Meta-heurísticas, Cidades Inteligentes

ABSTRACT

This paper addresses the problem of Location of Maximum Coverage Capable Facilities with Limited Range Drones (MCFLPD). Considering n customers with demands, m candidate locations for installations and a set of drones, the objective is to determine p locations for launching drones and distributing resources, maximizing coverage. Due to the high computational complexity of MCFLPD, a Genetic Algorithm (GA) with constructive heuristics inspired by the GRASP metaheuristic is proposed for the initial population. The GA was tested with literature data and compared with heuristics and an integer linear programming mathematical model, presenting very satisfactory results. Several scenarios were analyzed to assess the importance of the number of drones and the number of capable facilities.

KEYWORDS. Facility Location. Maximum Coverage. Genetic Algorithm.

Paper topics: Metaheuristics, Smart Cities

1. Introdução

O planejamento de distribuição é um problema crucial para empresas que lidam com a entrega de produtos ou serviços a seus clientes. Este problema envolve a otimização da cadeia de suprimentos, desde a produção realizada nas indústrias até o ponto de venda final, para garantir que os produtos cheguem aos clientes, no momento certo, e ao menor custo possível. Um dos desafios centrais do planejamento de distribuição é determinar a localização ideal dos centros de armazenamento e distribuição de mercadorias e nesse contexto, a Localização de Instalações Capacitadas de Máxima Cobertura (LICMC) se torna uma ferramenta poderosa para auxiliar na tomada de decisões estratégicas.

Os Veículos Aéreos Não Tripulados (VANTs), também conhecidos como drones, têm sido amplamente utilizados devido ao seu enorme potencial em logística, inspeção e monitoramento de terrenos [Martins, 2019]. O uso de drones teve um grande incremento com o lançamento do projeto “Prime Air” da Amazon com o objetivo de entregar mercadorias a clientes e também movimentá-las entre armazéns [Rose, 2013]. A exemplo da Amazon, muitas outras organizações também desenvolveram projetos-piloto [French, 2015]. Um projeto da DHL realiza a entrega urgente de bens e medicamentos em uma pequena ilha no norte da Alemanha [Hern, 2014]. O Project Wing da Google efetua a entrega de mercadorias através de um fio, sem a necessidade de pouso [Muoio, 2016]. Várias outras organizações estão aderindo ao uso de drones como UPS, Domino’s Pizza, FedEx [Sacramento, 2017] e Flytrex [Mack, 2018].

Por não dependerem de infraestrutura existente, os drones são ideais para entregas de última milha [LOGISTICS, 2014] e emergências [Karaca et al., 2018], como respostas emergenciais e fornecimento de suprimentos médicos pós-desastre [Thiels et al., 2015], [Scott e Scott, 2019]. Outras aplicações incluem irrigação de culturas e pulverização de pesticidas [Albornoz e Giraldo, 2017], [Berner e Chojnacki, 2017], [Burema e Filin, 2016], [Wang et al., 2016], [Giles et al., 2016], [Faïçal et al., 2014], [Costa et al., 2012].

A eficiência dos drones tem aumentado com o uso de materiais mais fortes e leves [Hasanalian e Abdelkefi, 2017], softwares de detecção de obstáculos [Yanmaz et al., 2017], [Yanmaz et al., 2018], e melhorias nas baterias [Li et al., 2017], [Fehrenbacher, 2018]. Com uma estrutura regulatória adequada, a adoção de UAVs, nos setores privados e públicos, deve acelerar em breve [Lei et al., 2022].

Mesmo com avanços, os UAVs enfrentam limitações de duração da bateria e capacidade de carga. Por isso, muitos estudos recentes sobre distribuição de encomendas exploram a cooperação entre caminhões e drones, combinando a maior capacidade dos caminhões com a agilidade dos drones [Ha et al., 2018]. Comparado a uma van a diesel RAM Pro, a capacidade de um drone é 378 vezes menor e seu alcance é quase 20 vezes menor [Figliozzi, 2017]. Drones são viáveis para entregas de última milha [Trop, 2016].

Diversos estudos exploram o uso de drones, como um problema de roteamento de veículos com drones. Por exemplo, Fu et al. [2018] propõem um algoritmo evolutivo para planejamento de rotas. Graf et al. [2017] apresentaram um algoritmo de otimização de colônia de formigas para planejamento de resgate com drones. Kyriakakis et al. [2023] propõem duas abordagens GRASP para o problema de roteamento de veículos elétricos com drones.

Poucos estudos foram conduzidos sobre o Problema de Localização de Facilidades com uso de Drones (PLFD) [Ilkhanizadeh et al., 2020]. Alguns trabalhos que abordam o PLFD consideram diversas restrições como, a vida útil da bateria, o alcance de voo e outras restrições [Udroiu et al., 2021], [Deaconu et al., 2021].

A primeira abordagem do problema de Localização de Instalações Capacitadas de Máxima Cobertura com Drones de Alcance Limitado (LICMCD) foi feita por Chauhan et al. [2019].

Estes autores apresentaram uma heurística de três fases (3SH) para decidir onde instalar depósitos, servido estes, como base para lançamento de drones. Al-Rabiaah et al. [2022b] também abordaram o problema LICMCD e apresentaram uma heurística denominada MCGRH, visando a maximização da cobertura de serviços a pacientes. Recentemente, um estudo apresentado por Dukkanci et al. [2023] analisa pesquisas sobre como encontrar os melhores locais para instalações usadas em sistemas de entrega por drones.

Neste trabalho propõe-se um Algoritmo Genético simples e eficiente para resolver o problema LICMCD. A solução do problema consiste das seguintes fases: (i) Seleção de um número pré-definido de instalações a serem abertas; (ii) Atribuição de clientes para instalações abertas; (iii) Alocação de uma quantidade suficiente de drones para as instalações abertas.

Para avaliar a qualidade das soluções obtidas pelo AG, utilizou-se a base de dados disponibilizada por Chauhan et al. [2019], onde foram disponibilizados os resultados da heurística 3SH. O AG também foi comparado com a heurística MCGRH de Al-Rabiaah et al. [2022a] utilizando a mesma base de dados.

Este trabalho está estruturado da seguinte forma: A Seção 2 apresenta a formulação do problema. A Seção 3 apresenta uma heurística construtiva gulosa-aleatória usada para gerar as soluções da população inicial. A seção 4 apresenta a implementação do Algoritmo Genético. A calibração dos parâmetros do AG e os experimentos computacionais são exibidos na Seção 5, seguidas das conclusões na Seção 6.

2. Formulação do problema

O problema MCFLPD, em estudo, pode ser formalmente definido da seguinte forma. Há um conjunto I de n clientes (cada cliente i com uma demanda w_i), um conjunto J de m locais potenciais (candidatos a centros de distribuição de recursos) e um conjunto D de nd drones com baterias totalmente carregadas. A duração máxima da bateria de um drone é B joule.

O objetivo do problema é selecionar/localizar p instalações ($p < m$) para serem os centros de distribuição (locais de lançamento dos drones) e atribuir drones a cada instalação selecionada de tal forma que a demanda total dos clientes servidos seja maximizada.

Como apresentado por Pirkul e Schilling [1989] e Chauhan et al. [2019], os locais candidatos à instalação de facilidades (centro de distribuição) têm uma capacidade U que representa a quantidade máxima de demanda que pode ser atendida por essa instalação, sendo definida como:

$$U = \frac{\sum_{i=1}^n w_i}{0,8 \times p} \quad (1)$$

Depois que um drone é atribuído a uma instalação, ele só pode fazer entregas nesse local. Os drones fazem apenas uma entrega por vez. Ao final de cada entrega, ele deverá retornar à instalação, e caso tenha capacidade residual de bateria, outras entregas poderão ser feitas. Não são consideradas recargas de bateria do drone. Supõe-se que o peso da demanda de cada cliente seja menor que a capacidade de carga de cada drone. Os custos de transporte das encomendas e dos drones, dos armazéns para os centros de distribuição, não são considerados.

Assim como em Figliozzi [2017] e Chauhan et al. [2019], o consumo da bateria b_{ij} para fazer a entrega de uma demanda w_i da instalação $j \in J$ para o cliente $i \in I$ (viagem de ida e volta do drone), é determinado pela Equação (2), onde d_{ij} é a distância euclidiana entre o cliente i e a instalação j , m_a é o peso do drone mais o peso da bateria, G representa o valor da gravidade, n_e é a eficiência de transferência de energia, r é a razão do arrasto e F_{sb} é o fator de segurança da bateria.

$$b_{ij} = \frac{d_{ij} \times (2 \times m_a + w_i) \times F_{sb} \times G}{n_e \times r} \quad (2)$$

A matriz de consumos b_{ij} tem por objetivo, quantificar o consumo de bateria do drone, desconsiderando-se obstáculos no trajeto, impacto dos ciclos de carregamento na bateria e impacto do clima (vento e temperatura);

Como em Chauhan et al. [2019], neste trabalho são usados $r = 3,5$; $ne = 0,66$; $F_{sb} = 1,25$; $B = 2797200 \text{ kg m}^2/\text{s}^2$ (joule); $m_a = 10,1 \text{ Kg}$; $G = 9,81 \text{ m/s}^2$;

3. Heurística construtiva

Para determinar uma solução inicial do problema LICMCD, propõe-se uma heurística construtiva gulosa com escolhas aleatórias, sendo composta de quatro fases distintas: (1) Abertura de p facilidades; (2) Alocação dos clientes para as facilidades abertas; (3) Alocação dos drones nas facilidades abertas; (4) Atribuição de drones para os clientes de cada facilidade aberta. A seguir são descritas cada uma destas fases.

3.1. Abertura das facilidades

A heurística construtiva inicia construindo um conjunto J_{open} com as p facilidades que possuam as maiores coberturas. Para isso são realizados os seguintes passos. (i) Calcule os consumos de bateria b_{ij} de todo cliente i para toda facilidade j (Equação (2)). (ii) Calcule a matriz de pesos wt_{ij} definida pela Equação (3) [Chauhan et al., 2019]. A matriz wt_{ij} atribui um peso para que um cliente i seja atendido pela instalação j . Note que, quanto maior seja a demanda do cliente i e menor seja o consumo de bateria b_{ij} , maior será o peso wt_{ij} . O peso será zero, se o trajeto consumir mais energia do que a duração máxima da bateria. (iii) Ordene os elementos da matriz wt_{ij} de forma decrescente obtendo uma lista ordenada L , e faça $J_{open} = \emptyset$. Faça $L' = L$ (L' será utilizada na seção 3.2). (iv) Enquanto $|J_{open}| < p$ e $L \neq \emptyset$ faça: a) Escolha aleatoriamente um elemento $wt_{i'j'}$ dentre os t primeiros elementos da lista L (t é denominado o tamanho da lista restrita de candidatos). b) Se $j' \notin J_{open}$, então faça $J_{open} = J_{open} \cup \{j'\}$. c) Faça $L = L - \{wt_{i'j'}\}$.

$$wt_{ij} = \begin{cases} \frac{w_i}{b_{ij}}, & \text{se } b_{ij} \leq B \\ 0, & \text{caso contrário} \end{cases} \quad (3)$$

3.2. Alocação de clientes nas facilidades abertas.

Nesta fase é construído um conjunto C contendo pares (i, j) . Cada par informa que um cliente i é atendido pela facilidade $j \in J_{open}$. Iniciando com $C = \emptyset$, a construção é feita nos seguintes passos. (i) Para cada elemento wt_{ab} da lista ordenada L' (criada na seção 3.1), se o cliente a já for atendido por uma facilidade, vá para o próximo elemento de L' ; caso contrário, identifique uma instalação $x \in J_{open}$ com capacidade residual disponível que possua o maior peso wt_{ax} . (ii) Adicione o par (a, x) ao conjunto C e diminua a capacidade residual da facilidade x (a capacidade de x será diminuída em w_a). Repita os passos (i) e (ii) até que toda lista L' seja percorrida ou até que não existam facilidades em J_{open} com capacidades residuais disponíveis.

3.3. Alocação dos drones nas facilidades abertas.

A alocação de drones para cada facilidade aberta segue estes passos: (i) Para cada facilidade $j \in J_{open}$, determine o número de drones necessários para atender as demandas dos clientes atribuídos a j : $S_j = \left\lceil \sum_{(i,j) \in C} \frac{b_{ij}}{B} \right\rceil$. (ii) Ordene as facilidades $j \in J_{open}$ de forma decrescente pelo valor S_j , obtendo a lista J'_{open} . (iii) Atribua drones às facilidades de J'_{open} sequencialmente. Na primeira rodada, atribua p drones. Depois de atribuir um drone a cada facilidade aberta, retorne à primeira e continue atribuindo um drone a cada facilidade sequencialmente. Se o número de drones atribuídos a uma facilidade atingir S_j , remova-a de J'_{open} . Pare o processo quando todos os drones forem atribuídos ou quando todas as facilidades tiverem recebido os drones necessários.

3.4. Atendimento dos drones aos clientes.

Na fase final da heurística, os drones entregam as demandas aos clientes. Para atribuir os drones, siga estes passos: (i) Para cada facilidade $j \in C$, ordene os clientes atendidos em ordem decrescente de wt_{ij} , criando a lista C' . (ii) Atribua sequencialmente os clientes de C' ao primeiro drone, respeitando as restrições de bateria. Quando uma restrição for violada, passe o cliente ao próximo drone disponível na facilidade. (iii) Repita o processo até que todos os clientes dessa facilidade estejam atribuídos ou até que nenhum cliente possa ser atribuído ao último drone disponível, sem violar a restrição. (iv) Repita para todas as facilidades.

3.5. Cálculo da função objetivo.

Considere a variável $x_{ijk} = 1$ se o cliente i é atendido pela facilidade aberta j usando o drone k , $x_{ijk} = 0$ caso contrário. A função objetivo é definida como o percentual de cobertura alcançada com todos os atendimentos realizados, ou seja, o percentual de demandas atendidas. Assim, a função objetivo a ser maximizada é definida como:

$$f = \frac{\sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^{nd} w_i \times x_{ijk}}{\sum_{i=1}^n w_i} \times 100\%.$$

4. Metaheurística: Algoritmo Genético.

Algoritmos genéticos (AGs) são algoritmos de busca probabilística baseados na teoria da evolução de Charles Darwin, que sugere que as espécies evoluem ao longo do tempo através da seleção natural [Holland, 1992]. As etapas básicas do AG proposto são. (i) Inicialização da População gerando um conjunto de soluções. (ii) Avaliação da população, onde são calculados a função de aptidão (*fitness*) de cada solução. (iii) Seleção, que consiste na escolha de soluções para reprodução. (iv) Cruzamento (Crossover), onde as soluções selecionadas geram novas soluções. (v) Mutação, que aplica pequenas perturbações nas novas soluções geradas. (vi) Atualização da População, que substitui parte da população antiga pelas novas soluções. As etapas (ii)-(vii) são repetidas até atingir o critério de parada, retornando a melhor solução encontrada. Todos estes passos podem ser visualizados na Figura (1a).

4.1. Representação de uma solução e construção da população inicial

Um cromossomo ou indivíduo no AG é representado por um conjunto de p facilidades abertas (p genes). O valor da função objetivo (% de cobertura) de um indivíduo é seu *fitness*. A população inicial (*Pop*) do AG é gerada da seguinte maneira. Metade de *Pop* é criada de forma aleatória, ou seja, as p facilidades são escolhidas aleatoriamente entre os m locais candidatos, sendo que, uma solução não pode conter facilidades repetidas. A outra metade de *Pop* é criada usando a heurística construtiva descrita na seção 3.1. Assim, a população contém tanto soluções totalmente aleatórias quanto soluções com um certo nível de gulosidade, aumentando significativamente o espaço de busca. A Figura (1b) exibe uma população de soluções (cromossomos), onde são mostrados os *fitness* de cada solução.

4.2. Seleção, crossover e mutação

O processo de seleção no AG é uma etapa fundamental que determina quais indivíduos de uma população serão escolhidos, com base no *fitness*, para reprodução. Neste trabalho foi implementada a seleção por torneio, onde dois indivíduos são escolhidos aleatoriamente de *Pop* e o indivíduo com o maior *fitness* é inserido em *Pop'* (população de pais). Este processo é repetido enquanto $|Pop'| < |Pop|$. Um exemplo da seleção por torneio pode ser visualizado na Figura (2a). O crossover cria novos indivíduos (filhos) combinando a informação genética dos pais. Neste trabalho, é usado o método de cruzamento proposto por Jánošíková et al. [2017], que preserva os genes comuns aos pais, pois esses genes representam facilidades que tendem a ter uma boa cobertura. A

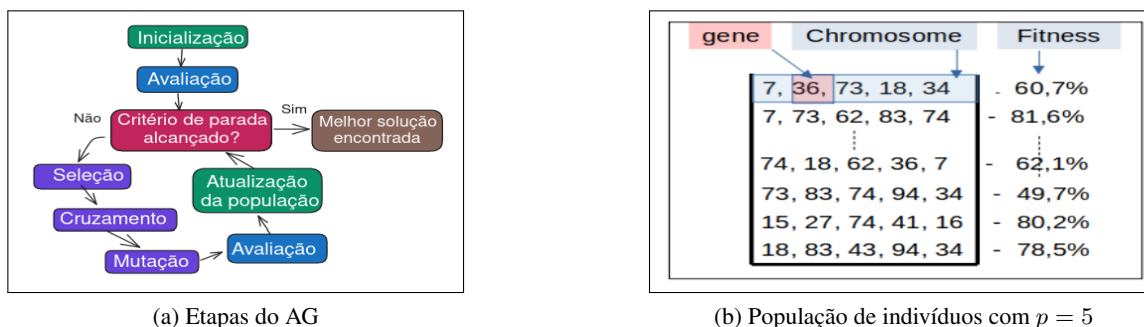


Figura 1: Estrutura do AG e Exemplo de uma população de cromossomos.

implementação do crossover segue os seguintes passos. (i) Crie uma nova população vazia Pop'' . (ii) Enquanto $|Pop''| < |Pop'|$: a) Selecione aleatoriamente dois indivíduos de Pop' (pais), p_1 e p_2 ; b) Crie dois novos filhos vazios (sem facilidades abertas), f_1 e f_2 ; c) Insira em f_1 e f_2 os genes comuns de p_1 e p_2 ; d) Complete o restante dos genes dos filhos até atingir p , escolhendo aleatoriamente genes dos pais p_1 e p_2 ;

Os filhos f_1 e f_2 gerados pelo crossover podem sofrer mutação, de acordo com uma probabilidade (taxa de mutação). A mutação é uma operação que introduz variações nos filhos, ajudando a manter a diversidade genética e evitando a convergência prematura da população. Para o problema em estudo, a mutação consiste em trocar uma facilidade aberta (gene do cromossomo) por uma facilidade fechada.

Na Figura (2b), observa-se um exemplo de crossover onde os pais selecionados são: $p_1 = \{7, 36, 73, 18, 34\}$ e $p_2 = \{73, 83, 74, 94, 34\}$. Os filhos f_1 e f_2 herdam os genes comuns (34, 73) dos pais, e os três genes restantes são escolhidos aleatoriamente de p_1 e p_2 . Se não houver genes comuns, todos os genes são escolhidos aleatoriamente de ambos os pais. Nesta Figura também mostra-se um exemplo de mutação, onde um gene (destacado em vermelho) de um dos filhos gerado é alterado. O pseudocódigo do AG é mostrado no Algoritmo 1.

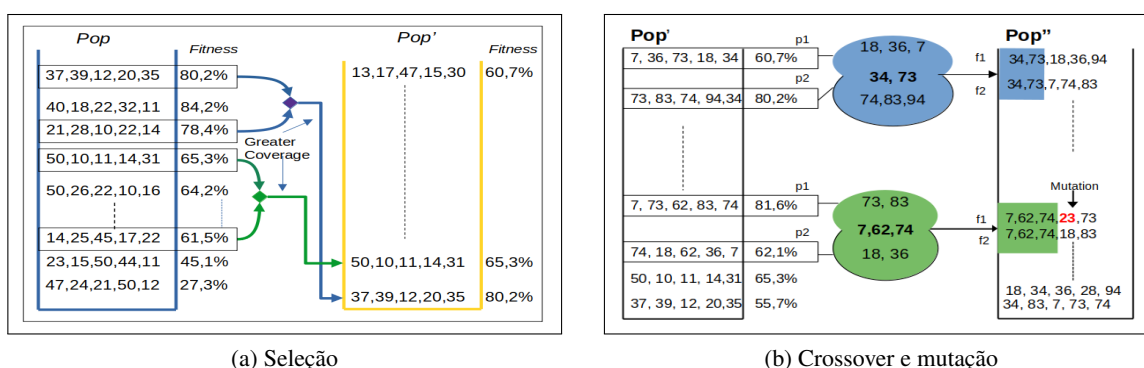


Figura 2: Seleção, Crossover e Mutação

5. Experimentos Computacionais

Os experimentos foram realizados em um computador com processador Intel Core i7-8700 CPU 3.19 GHz, 4 núcleos físicos, 8 lógicos, e 8 GB de RAM, rodando Ubuntu 22.04 64-bit.

Foram usadas as instâncias geradas por Chauhan et al. [2019], que possuem 104 instâncias comunitários (conjunto J), com espaço para lançamento de drones e armazenamento de

Algorithm 1 AG (*PopSize*, *Niter*, *taxaM*)

```

1: Pop ← Crie uma lista inicial com PopSize indivíduos
2: for iter ← 1 to Niter do
3:   Pop' ← ∅ ▷ Inicie Pop' como uma lista vazia
4:   while  $|Pop'| < |Pop|$  do
5:     I1 ← Escolha aleatoriamente um indivíduo de Pop
6:     I2 ← Escolha aleatoriamente um indivíduo de Pop
7:     I ← Indivíduo com melhor fitness entre I1 e I2
8:     Adicione em Pop' o indivíduo I
9:   end while
10:  Pop'' ← ∅ ▷ Inicie Pop'' como uma lista vazia
11:  while  $|Pop''| < |Pop|$  do
12:    p1, p2 ← Escolha aleatoriamente dois indivíduos de Pop'
13:    f1, f2 ← Crossover(p1, p2)
14:    f1 ← Mutação(f1, taxaM)
15:    f2 ← Mutação(f2, taxaM)
16:    Adicione em Pop'' os indivíduos f1 e f2
17:  end while
18:  Pop ← Pop''
19: end for
20: return O melhor indivíduo encontrado
  
```

recursos; e 122 clientes (conjunto *I*). Para cada instalação e cliente são disponibilizadas as coordenadas de latitude e longitude. As demandas w_i de cada cliente *i* variam de 1 kg a 5 kg em intervalos de 0,25 kg, totalizando 366,5 kg. Cada instalação $j \in J$ possui a mesma capacidade *U* que é determinada conforme a Equação (1). O número de facilidades *p* a serem abertas varia de 5 a 30, em intervalos de 5. O número de drones *nd* varia de 20 a 100. Combinando *p* e *nd*, Chauhan et al. [2019] geraram 22 instâncias.

O tamanho *t* da lista restrita de candidatos, na heurística construtiva, foi determinado de forma empírica. Bons resultados foram obtidos com *t* = 10. A taxa de crossover é sempre 100%, ou seja, dois pais sempre serão cruzados gerando dois filhos. A taxa de mutação (*taxaM*), o tamanho da população (*PopSize*) e o critério de parada (*Niter*) foram calibrados experimentalmente. Para diferentes valores dos parâmetros, o AG foi rodado trinta vezes. Os resultados médios (valores médios do percentual de coberturas) obtidos na calibração de *taxaM*, *PopSize* e *Niter* são mostrados nas Tabelas 1, 2 e 3, respectivamente. Valores em negrito representam os melhores resultados. Para visualizar melhor esses resultados, foram realizados gráficos de boxplot com o objetivo tornar mais nítida a visualização das estatísticas para fazer análises em relação a dispersão dos resultados, média, outliers (mínimos e máximos) e mediana. A qualidade da solução obtida pelo AG é avaliado considerando o Desvio Percentual Relativo (*RPD*) com relação à melhor solução conhecida f_{best} (obtida em todas as rodadas do AG). O *RPD* é calculada como: $RPD = \frac{f_{best} - f}{f_{best}}$. Utilizando o *RPD* como variável de resposta, nas Figuras (3a), (3b), (3c) são exibidos os boxplots obtidos na calibração da taxa de mutação, tamanho da população e critério de parada, respectivamente. Nos boxplots das Figuras (3a), (3b) e (3c), observam-se que para *taxaM* = 27%, *PopSize* = 200 e *Niter* = 200 (critério de parada) o AG apresentou os melhores resultados. Estes parâmetros são utilizados nos próximos experimentos. Para tempos de resposta mais rápidos, uma população com 100 indivíduos poderá ser utilizada, apresentando bons resultados (Tabela 2).

5.1. Comparação dos Resultados com a Literatura

O desempenho do AG é comparado com as heurísticas 3SH e MCGRH, propostas, respectivamente, por Chauhan et al. [2019] e Al-Rabiaah et al. [2022a], para o mesmo problema abordado neste trabalho. Os resultados do AG também são comparados com os resultados do modelo de Programação Linear Inteira (PLI) de Chauhan et al. [2019], que é resolvido pelo solver Gurobi. Este solver é executado com um tempo limite de CPU de 1800 segundos para cada instância.

A Tabela 4 mostra os resultados obtidos pelo Gurobi, pelas heurísticas 3SH e MCGRH, e pelo AG proposto. As heurísticas 3SH, MCGRH e AG foram executados 30 vezes para cada

Tabela 1: Valores médios de cobertura com diferentes taxas de mutação.

		Taxa de Mutação (%)									
<i>p</i>	<i>nd</i>	3	6	9	12	15	18	21	24	27	30
5	20	53,9	53,9	54,0	54,1	54,0	53,9	54,1	53,9	54,1	54,1
5	25	59,4	59,5	59,5	59,3	59,5	59,6	59,5	59,7	59,5	59,6
5	30	63,2	63,2	63,7	63,6	63,9	63,9	63,7	63,8	63,8	63,7
5	35	66,6	66,6	66,3	66,5	66,5	66,5	66,5	67,1	67,2	67,2
5	40	68,7	69,0	69,1	68,8	68,9	69,0	69,2	68,9	69,5	69,1
10	20	62,2	62,0	62,0	62,2	62,4	62,3	62,6	62,1	62,3	62,0
10	30	71,5	71,3	71,5	71,8	71,7	71,6	71,7	71,8	71,4	72,0
10	40	78,4	78,9	79,0	79,2	79,5	79,5	79,2	79,1	79,3	79,1
15	30	76,7	77,3	77,4	77,4	77,6	77,8	77,6	77,7	77,4	77,3
15	45	86,6	87,1	87,4	87,2	87,2	87,2	87,4	87,2	87,5	87,2
15	60	89,7	89,9	89,7	90,0	89,6	89,8	89,9	90,1	89,8	90,2
20	20	69,9	70,1	70,4	70,3	70,2	70,2	70,3	70,4	70,4	70,3
20	40	86,8	87,5	87,2	87,7	87,4	87,3	87,5	87,5	87,7	87,5
20	60	92,1	92,0	91,8	92,1	91,9	92,1	92,2	91,9	92,1	92,4
20	80	91,6	92,0	92,4	92,0	91,9	92,0	92,0	92,0	92,4	92,0
25	25	77,6	78,2	78,6	78,6	78,5	78,4	78,3	78,7	78,6	78,7
25	50	92,3	92,5	92,8	92,8	92,8	93,0	92,8	92,5	92,8	92,8
25	75	93,4	93,2	93,3	93,5	93,5	93,7	93,4	93,7	93,7	93,8
25	100	93,4	93,5	93,3	93,5	93,7	93,6	93,5	93,5	93,5	93,6

Tabela 2: Valores médios da cobertura e tempo em diferentes tamanhos de população.

		Tempo (S)			Cv (%)		
		<i>PopSize</i>			<i>PopSize</i>		
<i>p</i>	<i>nd</i>	100	150	200	100	150	200
5	20	6,3	9,5	12,7	53,8	53,8	53,9
5	25	6,8	10,2	13,6	59,2	59,0	59,0
5	30	7,2	10,9	14,5	63,3	63,2	63,2
5	35	7,6	11,5	15,4	66,7	66,7	66,6
5	40	8,0	12,1	16,1	68,6	68,6	68,8
10	20	9,1	13,8	18,4	61,5	61,6	62,1
10	30	10,0	15,0	20,1	71,9	71,8	71,7
10	40	10,9	16,5	22,1	78,5	78,9	79,2
15	30	12,4	18,7	24,8	76,6	77,1	77,1
15	45	13,9	20,9	27,8	86,9	87,2	87,2
15	60	14,6	24,3	29,1	89,6	89,6	89,8
20	20	14,1	24,4	28,1	70,1	70,0	70,2
20	40	15,8	27,1	31,4	87,5	87,1	87,6
20	60	17,0	29,1	33,8	92,3	91,8	92,1
20	80	17,0	25,8	34,1	91,8	92,1	92,0
25	25	17,8	26,5	35,3	78,0	78,1	78,6
25	50	19,7	29,0	38,8	92,5	92,7	93,0
25	75	19,6	29,4	39,2	93,3	93,7	93,6
25	100	19,7	29,3	39,2	93,5	93,6	93,6
30	30	20,6	30,7	40,8	83,0	83,3	83,4
30	60	22,9	34,2	45,7	93,8	93,7	93,8
30	90	22,9	34,2	45,7	93,7	93,8	93,8

Tabela 3: Critérios de parada.

		Tempo (sec)				Cv (%)			
		<i>Niter</i>				<i>Niter</i>			
<i>p</i>	<i>nd</i>	50	100	150	200	50	100	150	200
5	20	3,7	6,6	9,7	12,8	53,8	54,0	54,0	53,9
5	25	3,9	7,2	10,3	13,6	59,0	58,9	59,1	59,0
5	30	4,2	7,6	11,0	14,5	63,2	63,4	63,2	63,3
5	35	4,4	8,0	11,7	15,3	66,7	66,5	66,8	66,8
5	40	4,6	8,3	12,1	16,0	68,7	68,8	68,7	68,6
10	20	5,2	9,5	13,9	18,4	61,3	62,0	62,2	62,1
10	30	5,7	10,4	15,2	20,0	71,0	71,4	71,4	71,8
10	40	6,2	11,4	16,5	21,9	78,6	78,7	78,5	79,0
15	30	7,0	12,9	19,0	25,0	76,8	77,0	77,4	77,5
15	45	7,6	14,3	21,1	27,8	86,5	87,2	87,2	86,9
15	60	7,8	14,8	21,9	29,3	89,5	89,8	89,6	89,6
20	20	8,0	14,7	21,5	28,4	69,7	69,7	69,6	69,9
20	40	8,7	16,3	24,1	31,7	87,0	87,7	87,6	87,6
20	60	9,2	17,4	25,9	34,2	92,0	92,3	92,1	92,1
20	80	9,2	17,6	25,8	34,2	92,0	92,0	92,0	92,0
25	25	10,2	18,0	26,8	35,4	77,5	78,3	78,6	78,5
25	50	11,3	20,0	29,6	39,0	92,0	92,6	92,7	92,7
25	75	10,6	20,1	29,8	39,2	93,2	93,5	93,6	93,7
25	100	10,6	20,1	29,9	39,2	93,3	93,4	93,4	93,6
30	30	11,1	21,1	31,2	41,2	82,2	83,4	83,5	83,3
30	60	12,0	23,3	34,6	45,9	93,8	93,8	93,8	93,8
30	90	12,0	23,3	34,6	45,7	93,8	93,8	93,7	93,8

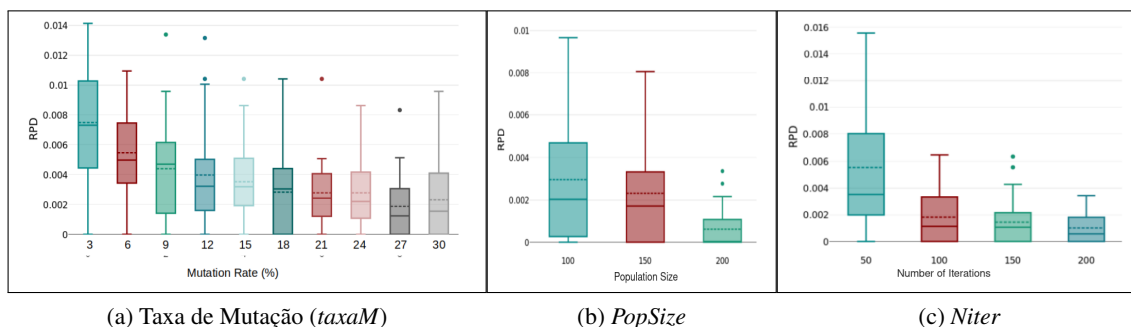


Figura 3: Gráfico de boxplots obtidos na calibração dos parâmetros do AG.

instância. A primeira coluna da tabela indica os tamanhos das instâncias (número de facilidades a serem abertas e número de drones). As colunas subsequentes apresentam os valores médios das coberturas máximas (Cv%) e os tempos médios de CPU (em segundos). Valores em **negrito** representam os melhores resultados. Observa-se que, o AG foi superior às heurísticas 3SH e MCGRH em 90,9% e 72,7% dos casos. As coberturas encontradas pelo Gurobi sempre foram melhores, no entanto o tempo computacional gasto foram bem maiores. Os resultados mostram que o AG teve resultados muito superiores, especialmente nas instâncias com maiores números de facilidades a serem abertas.

A qualidade das soluções encontradas pelas heurísticas também é avaliada através do Desvio Percentual Relativo (RPD) com relação à melhor solução conhecida (encontrada pelas 3 heurísticas). Na Figura 4 mostra-se o gráfico de boxplots que usa o RPD como variável de resposta. Observa-se claramente que, o desempenho do AG é significativamente superior que o desempenho das heurísticas 3SH e MCGRH (a dispersão dos resultados, valores mínimos, máximos e mediana são bem menores).

Tabela 4: Comparação entre Gurobi, 3SH, MCGRH e AG.

<i>p</i>	<i>nd</i>	Gurobi			3SH		MCGRH		AG	
		tempo (s)	Cv%	Gap	tempo (s)	Cv%	tempo (s)	Cv%	tempo (s)	Cv%
5	20	1800,4	56,5	2,1	14,7	54,5	0,7	51,5	14,0	54,0
5	25	1800,3	61,9	2,0	15,9	59,5	0,6	57,9	15,3	59,6
5	30	1800,3	66,0	2,9	16,7	63,7	0,7	63,4	16,2	63,7
5	35	1800,3	69,4	2,8	18,3	67,0	0,7	68,1	16,9	67,0
5	40	1800,3	70,8	6,2	18,8	69,9	0,7	71,0	17,8	69,0
10	20	1800,3	64,3	2,8	16,6	61,4	0,7	57,4	19,9	62,2
10	30	1800,4	74,8	3,2	18,3	71,5	0,8	70,0	21,6	71,7
10	40	1800,4	83,8	1,9	20,1	78,4	0,9	79,2	23,5	79,1
15	30	1800,5	79,9	3,2	21,9	75,2	1,1	72,4	26,8	77,5
15	45	1800,5	89,0	3,1	24,3	83,9	0,9	86,4	30,0	87,3
15	60	1800,6	92,4	1,5	24,8	85,0	1,1	90,4	31,4	89,9
20	20	1800,4	71,5	3,5	25,1	65,8	1,0	59,3	30,7	70,3
20	40	1800,4	90,5	2,5	28,7	84,2	1,0	83,8	34,2	87,6
20	60	72,2	93,8	0,0	30,4	87,2	1,1	92,3	36,9	92,1
20	80	72,2	93,8	0,0	30,9	87,5	1,0	91,4	36,8	92,1
25	25	1800,3	79,9	3,5	33,0	71,5	1,1	67,7	39,0	78,5
25	50	1574,2	93,8	0,0	36,6	88,9	1,2	90,9	41,5	92,9
25	75	44,5	93,8	0,0	37,4	88,2	1,2	92,2	41,6	93,6
25	100	46,5	93,8	0,0	38,1	89,5	1,2	92,2	41,7	93,6
30	30	1800,4	85,9	4,0	40,3	76,8	1,3	72,2	43,9	83,3
30	60	78,7	93,8	0,0	44,3	90,9	1,5	93,2	48,9	93,8
30	90	30,7	93,8	0,0	45,1	90,7	1,2	92,9	48,8	93,8

6. Conclusões.

Neste trabalho foi abordado o problema de Localização de Facilidades Capacitadas de Máxima Cobertura com Drones. O objetivo do problema é selecionar *p* facilidades para o lançamento de drones, visando maximizar a cobertura das entregas aos clientes. Foi proposto um Algoritmo Genético (AG) simples e eficiente que utiliza uma heurística construtiva gulosa para avaliação das

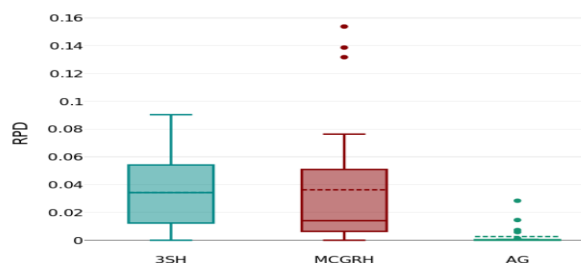


Figura 4: Gráfico de boxplots obtidos na comparação das heurísticas 3SH, MCGRH e AG.

soluções. O AG foi testado em diversos cenários variando-se a taxa de mutação, tamanho da população e número de iterações. Os experimentos computacionais mostraram que o AG é altamente adaptável a diferentes instâncias, bastando-se ajustar os seus parâmetros, para equilibrar a qualidade da solução e o tempo computacional. Por exemplo, uma população de 100 indivíduos oferece um tempo de resposta baixo, ideal para situações que exigem rapidez.

O AG apresentou desempenhos próximos ao modelo matemático resolvido pelo Gurobi, mas com um tempo computacional muito menor. Comparado às heurísticas 3SH e MCGRH da literatura, o AG superou a 3SH em 90,9% dos casos e a MCGRH em 72,7%.

Apesar dos resultados promissores, o desempenho do AG pode ser comprometido em instâncias com grande número de clientes e facilidades, especialmente quando a calibração dos parâmetros não é adequada. Além disso, cenários com alta variabilidade nas demandas podem dificultar o equilíbrio no atendimento aos clientes. Investigações futuras podem focar em análises mais aprofundadas desses aspectos, bem como na implementação de uma busca local (BL) para refinar a melhor solução gerada pelo AG. Outra direção promissora inclui o uso do IRace para facilitar a calibração dos parâmetros, ajustando-os de maneira mais eficiente, e a realização de testes estatísticos para validar os resultados obtidos.

7. Agradecimentos

Os autores agradecem o apoio financeiro da CAPES.

Referências

- Al-Rabiaah, S., Hosny, M., e AlMuhaideb, S. (2022a). An efficient greedy randomized heuristic for the maximum coverage facility location problem with drones in healthcare. *Applied Sciences*, 12 (3):1403.
- Al-Rabiaah, S., Hosny, M., e AlMuhaideb, S. (2022b). A greedy heuristic based on optimizing battery consumption and routing distance for transporting blood using unmanned aerial vehicles. *Electronics*, 11(20):3399.
- Albornoz, C. e Giraldo, L. F. (2017). Trajectory design for efficient crop irrigation with a uav. In *2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)*, p. 1–6. IEEE.
- Berner, B. e Chojnacki, J. (2017). Use of drones in crop protection.
- Burema, H. e Filin, A. (2016). Aerial farm robot system for crop dusting, planting, fertilizing and other field jobs. US Patent 9,382,003.
- Chauhan, D., Unnikrishnan, A., e Figliozzi, M. (2019). Maximum coverage capacitated facility location problem with range constrained drones. *Transportation Research Part C: Emerging Technologies*, 99:1–18.

- Costa, F. G., Ueyama, J., Braun, T., Pessin, G., Osório, F. S., e Vargas, P. A. (2012). The use of unmanned aerial vehicles and wireless sensor network in agricultural applications. In *2012 IEEE International Geoscience and Remote Sensing Symposium*, p. 5045–5048. IEEE.
- Deaconu, A. M., Udriou, R., e Nanau, C.-Ş. (2021). Algorithms for delivery of data by drones in an isolated area divided into squares. *Sensors*, 21(16):5472.
- Dukkanci, O., Campbell, J. F., e Kara, B. Y. (2023). Facility location decisions for drone delivery: A literature review. *European Journal of Operational Research*.
- Façal, B. S., Costa, F. G., Pessin, G., Ueyama, J., Freitas, H., Colombo, A., Fini, P. H., Villas, L., Osório, F. S., Vargas, P. A., et al. (2014). The use of unmanned aerial vehicles and wireless sensor networks for spraying pesticides. *Journal of Systems Architecture*, 60(4):393–404.
- Fehrenbacher, K. (2018). A new lithium-metal battery takes flight in drones. *Greentech Media*.
- Figliozzi, M. A. (2017). Lifecycle modeling and assessment of unmanned aerial vehicles (drones) co2e emissions. *Transportation Research Part D: Transport and Environment*, 57:251–261.
- French, S. (2015). Drone delivery is already here—and it works. *Market Watch*, December, 15.
- Fu, Z., Yu, J., Xie, G., Chen, Y., Mao, Y., et al. (2018). A heuristic evolutionary algorithm of uav path planning. *Wireless Communications and Mobile Computing*, 2018.
- Giles, D., Billing, R., e Singh, W. (2016). Performance results, economic viability and outlook for remotely piloted aircraft for agricultural spraying.
- Graf, M., Poy, M., Bischof, S., Dornberger, R., e Hanne, T. (2017). Rescue path optimization using ant colony systems. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, p. 1–7. IEEE.
- Ha, Q. M., Deville, Y., Pham, Q. D., e Hà, M. H. (2018). On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, 86:597–621.
- Hassanalian, M. e Abdelkefi, A. (2017). Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace sciences*, 91:99–131.
- Hern, A. (2014). Dhl launches first commercial drone ‘parcelcopter’ delivery service. *The Guardian*, 25:2014.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- Ilkhanizadeh, S., Golabi, M., Hesami, S., e Rjoub, H. (2020). The potential use of drones for tourism in crises: A facility location analysis perspective. *Journal of Risk and Financial Management*, 13(10):246.
- Jánošíková, L., Herda, M., e Haviar, M. (2017). Hybrid genetic algorithms with selective crossover for the capacitated p-median problem. *Central European Journal of Operations Research*, 25: 651–664.
- Karaca, Y., Cicek, M., Tatli, O., Sahin, A., Pasli, S., Beser, M. F., e Turedi, S. (2018). The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations. *The American journal of emergency medicine*, 36(4):583–588.

- Kyriakakis, N. A., Stamadianos, T., Marinaki, M., e Marinakis, Y. (2023). A grasp approach for the energy-minimizing electric vehicle routing problem with drones. *World Electric Vehicle Journal*, 14(12):354.
- Lei, T., Wang, Y., Jin, X., Min, Z., Zhang, X., e Zhang, X. (2022). An optimal fuzzy logic-based energy management strategy for a fuel cell/battery hybrid power unmanned aerial vehicle. *Aerospace*, 9(2):115.
- Li, K.-R., See, K.-Y., Koh, W.-J., e Zhang, J.-W. (2017). Design of 2.45 ghz microwave wireless power transfer system for battery charging applications. In *2017 Progress in Electromagnetics Research Symposium-Fall (PIERS-FALL)*, p. 2417–2423. IEEE.
- LOGISTICS, O.-C. (2014). A dhl perspective on implications and use cases for the logistics industry.
- Mack, E. (2018). How delivery drones can help save the world. *Forbes, Forbes Magazine*.
- Martins, J. P. R. (2019). Utilização de veículos aéreos não tripulados no registo e investigação de acidentes de viação: viabilidade e implicações.
- Muoio, D. (2016). Google's secretive drone delivery project just got cleared for testing—here's everything we know about the program. URL <http://www.techinsider.io/google-projectwing-drone-service-2016-8>.
- Pirkul, H. e Schilling, D. (1989). The capacitated maximal covering location problem with backup service. *Annals of Operations Research*, 18(1):141–154.
- Rose, C. (2013). Amazon's jeff bezos looks to the future. *CBS News*, 1.
- Sacramento, D. (2017). Heuristics for solving the drone-vehicle routing problem (masters thesis). Technical report, Technical report, Technical University of Denmark.
- Scott, J. E. e Scott, C. H. (2019). Models for drone delivery of medications and other healthcare items. *Unmanned aerial vehicles: Breakthroughs in research and practice*, p. 376–392.
- Thiels, C. A., Aho, J. M., Zietlow, S. P., e Jenkins, D. H. (2015). Use of unmanned aerial vehicles for medical product transport. *Air medical journal*, 34(2):104–108.
- Trop, J. (2016). Drone delivery is about to disrupt the trucking industry.
- Udroiu, R., Deaconu, A. M., e Nanau, C.-Ş. (2021). Data delivery in a disaster or quarantined area divided into triangles using dtn-based algorithms for unmanned aerial vehicles. *Sensors*, 21(11): 3572.
- Wang, C., He, X., Liu, Y., Song, J., Zeng, A., et al. (2016). The small single-and multi-rotor unmanned aircraft vehicles chemical application techniques and control for rice fields in china. *Aspects of Applied Biology*, 132:73–81.
- Yanmaz, E., Quaritsch, M., Yahyanejad, S., Rinner, B., Hellwagner, H., e Bettstetter, C. (2017). Communication and coordination for drone networks. In *Ad Hoc Networks: 8th International Conference, ADHOCNETS 2016, Ottawa, Canada, September 26-27, 2016, Revised Selected Papers*, p. 79–91. Springer.
- Yanmaz, E., Yahyanejad, S., Rinner, B., Hellwagner, H., e Bettstetter, C. (2018). Drone networks: Communications, coordination, and sensing. *Ad Hoc Networks*, 68:1–15.