



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO MARANHÃO
CAMPUS SÃO LUÍS - MONTE CASTELO
DEPARTAMENTO DE ELETROELETRÔNICA
CURSO DE BACHARELADO EM ENGENHARIA ELÉTRICA INDUSTRIAL

RÔMULO MENDES RODRIGUES

ALOCÇÃO DE ESTAÇÕES RÁDIO BASE DE TELEFONIA MÓVEL 5G
UTILIZANDO ALGORITMOS BIOINSPIRADOS: simulação computacional no município
de Paço do Lumiar - MA

São Luís - MA
2025

RÔMULO MENDES RODRIGUES

**ALOCÇÃO DE ESTAÇÕES RÁDIO BASE DE TELEFONIA MÓVEL 5G
UTILIZANDO ALGORITMOS BIOINSPIRADOS:** simulação computacional no município
de Paço do Lumiar - MA

Monografia apresentada ao Curso de Bacharelado em Engenharia Elétrica Industrial do Instituto Federal de Educação, Ciência e Tecnologia do Maranhão, Campus São Luís - Monte Castelo, como requisito para a obtenção do grau de Bacharel em Engenharia Elétrica Industrial.

Orientadora: Prof^ª. Dra. Danúbia Soares Pires

São Luís - MA

2025

R966a Rodrigues, Rômulo Mendes.

Alocação de estações rádio base de telefonia móvel 5G utilizando algoritmos bioinspirados: simulação computacional no município de Paço do Lumiar-MA/ Rômulo Mendes Rodrigues. – São Luís, 2025.
62p.: il.

Monografia (Graduação) – Curso de Bacharelado em Engenharia Elétrica Industrial, Instituto Federal de Educação, Ciência e Tecnologia do Maranhão, Campus São Luís – Monte de Castelo, 2025.

Orientadora: Profa. Dra. Danúbia Soares Pires.

1. Alocação de ERBs. 2. 5G. 3. Algoritmo Genético. 4. Otimização por enxame de partículas. I. Título.

CDU 621.396.97:004

RÔMULO MENDES RODRIGUES

**ALOCÇÃO DE ESTAÇÕES RÁDIO BASE DE TELEFONIA MÓVEL 5G
UTILIZANDO ALGORITMOS BIOINSPIRADOS: simulação computacional no município
de Paço do Lumiar - MA**

Monografia apresentada ao Curso de Bacharelado em Engenharia Elétrica Industrial do Instituto Federal de Educação, Ciência e Tecnologia do Maranhão, Campus São Luís - Monte Castelo, como requisito para a obtenção do grau de Bacharel em Engenharia Elétrica Industrial.

Aprovada em 20 de outubro de 2025.

BANCA EXAMINADORA

Prof^ª. Dra. Danúbia Soares Pires (Orientadora)
Instituto Federal de Educação, Ciência e Tecnologia do
Maranhão (IFMA)

Prof. Dr. Orlando Donato Rocha Filho (Examinador)
Instituto Federal de Educação, Ciência e Tecnologia do
Maranhão (IFMA)

Prof^ª. Dra. Priscila Lima Rocha (Examinador)
Instituto Federal de Educação, Ciência e Tecnologia do
Maranhão (IFMA)

Prof. Me. Diego Dutra Sampaio (Examinador)
Universidade Federal do Maranhão (UFMA)

Dedico este trabalho ao meu pai, Romaelson Ribeiro Rodrigues (*in memoriam*), à minha mãe e à minha irmã, com todo o meu amor e gratidão.

AGRADECIMENTOS

A Deus, pela dádiva da vida e por me conceder força e sabedoria para superar todos os desafios ao longo do curso.

Ao Sr. Paulo Roberto, servidor da ANATEL, cuja sugestão sobre a otimização da localização de ERBs foi o ponto de partida para a construção desta pesquisa.

À professora Danúbia Pires, pela orientação e pelo apoio durante o desenvolvimento deste trabalho.

Aos meus amigos de curso, pela convivência, companheirismo e apoio durante todos esses anos de caminhada acadêmica.

Aos professores, pelos ensinamentos e orientações que contribuíram significativamente para o meu desenvolvimento acadêmico e para a construção da minha formação profissional ao longo do curso.

RESUMO

Este trabalho teve como objetivo aplicar os algoritmos bioinspirados Algoritmo Genético (AG) e Otimização por Enxame de Partículas (PSO) para otimizar a alocação de Estações Rádio Base (ERBs) da rede móvel 5G na frequência de 3,5 GHz, maximizando a cobertura do sinal no município de Paço do Lumiar - MA, por meio de simulação computacional. A justificativa para a pesquisa decorreu da crescente demanda por conectividade de alta velocidade e baixa latência, bem como dos desafios técnicos e econômicos impostos pela implantação do 5G, que exige maior densidade de ERBs devido à menor propagação do sinal em altas frequências. Para enfrentar esse problema, classificado como difícil em tempo não polinomial (NP-difícil), optou-se pelo uso de técnicas meta-heurísticas capazes de explorar amplos espaços de busca e encontrar soluções satisfatórias em tempo viável. A metodologia consistiu na modelagem do problema de cobertura por meio da discretização do território em pontos de demanda, definição da função objetivo baseada no percentual de cobertura atendida e implementação computacional dos algoritmos em linguagem Python. Foram realizadas simulações em diferentes cenários, variando o número de ERBs por indivíduo/partícula (55, 50, 45 e 40), a fim de identificar o número mínimo necessário para garantir cobertura integral. Os resultados demonstraram que o AG atingiu até 99,46% de cobertura com 55 ERBs, enquanto o PSO alcançou até 94,02% nas mesmas condições, indicando melhor desempenho do AG quanto à cobertura. O AG também se mostrou mais eficiente em termos de tempo de processamento que o PSO. Concluiu-se que algoritmos analisados são eficientes para apoiar o planejamento de redes 5G, e que, no conjunto de cenários analisados, o AG se destacou como a abordagem mais vantajosa para conciliar qualidade de solução e eficiência computacional, contribuindo para reduzir custos de implantação e assegurar a cobertura necessária.

Palavras-chave: alocação de ERBs; 5G; Algoritmo Genético; Otimização por Enxame de Partículas.

ABSTRACT

This work aimed to apply the bioinspired algorithms Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) to optimize the allocation of Radio Base Stations (RBSs) in the 5G mobile network at 3.5 GHz, maximizing signal coverage in the municipality of Paço do Lumiar, Maranhão, through computational simulation. The rationale for the research stemmed from the growing demand for high-speed, low-latency connectivity, as well as the technical and economic challenges posed by 5G deployment, which requires a higher density of RBSs due to reduced signal propagation at high frequencies. To address this problem, classified as Non-deterministic Polynomial-time hard (NP-hard), we opted to use metaheuristic techniques capable of exploring broad search spaces and finding satisfactory solutions in a feasible time. The methodology consisted of modeling the coverage problem by discretizing the territory into demand points, defining the objective function based on the percentage of coverage achieved, and computational implementation of the algorithms in Python. Simulations were performed under different scenarios, varying the number of RBSs per individual/particle (55, 50, 45, and 40), to identify the minimum number required to ensure full coverage. The results demonstrated that the GA achieved up to 99.46% coverage with 55 RBSs, while the PSO achieved up to 94.02% under the same conditions, indicating better coverage performance for the GA. The GA also proved to be more efficient in terms of processing time than the PSO. It was concluded that the analyzed algorithms are effective in supporting 5G network planning, and that, in the set of scenarios analyzed, the GA stood out as the most advantageous approach for reconciling solution quality and computational efficiency, contributing to reducing deployment costs and ensuring the necessary coverage.

Keywords: RBS allocation; 5G; Genetic Algorithm; Particle Swarm Optimization.

LISTA DE FIGURAS

Figura 1 – Arquitetura simplificada de uma rede de telefonia móvel	17
Figura 2 – ERB do tipo <i>Greenfield</i>	18
Figura 3 – ERB do tipo <i>Rooftop</i>	18
Figura 4 – Extremos de uma função	20
Figura 5 – Fluxograma do AG	24
Figura 6 – Atualização da posição da partícula no espaço vetorial	28
Figura 7 – Fluxograma do PSO	28
Figura 8 – Polígono territorial de Paço do Lumiar	29
Figura 9 – Discretização do mapa	30
Figura 10 – Cobertura estimada de uma ERB 5G	32
Figura 11 – Vetor de coordenadas geográficas de um indivíduo	34
Figura 12 – Representação gráfica de um indivíduo	34
Figura 13 – Região de busca	35
Figura 14 – Exemplo de indivíduos selecionados como pais	35
Figura 15 – Exemplo de descendentes gerados	36
Figura 16 – Exemplo de indivíduo que sofreu mutação	36
Figura 17 – Exemplo de atualização da posição de uma partícula	38
Figura 18 – Evolução da cobertura em cada configuração de ERBs no AG	42
Figura 19 – Mapa com cobertura de 99,46% obtido com 55 ERBs pelo AG	42
Figura 20 – Evolução da cobertura em cada configuração de ERBs no PSO	44
Figura 21 – Mapa com cobertura de 94,02% obtido com 55 ERBs pelo PSO	45
Figura 22 – Média de cobertura dos algoritmos	46
Figura 23 – Tempo médio de processamento dos algoritmos	46

LISTA DE TABELAS

Tabela 1 – Simulações com 55 ERBs por indivíduo	40
Tabela 2 – Simulações com 50 ERBs por indivíduo	41
Tabela 3 – Simulações com 45 ERBs por indivíduo	41
Tabela 4 – Simulações com 40 ERBs por indivíduo	41
Tabela 5 – Simulações com 55 ERBs por partícula	43
Tabela 6 – Simulações com 50 ERBs por partícula	43
Tabela 7 – Simulações com 45 ERBs por partícula	43
Tabela 8 – Simulações com 40 ERBs por partícula	44

LISTA DE ABREVIATURAS E SIGLAS

5G	Quinta Geração
AG	Algoritmo Genético
ANATEL	Agência Nacional de Telecomunicações
BSC	<i>Base Station Controller</i> (Controlador de Estações Rádio Base)
CCC	Central de Comutação e Controle
ERB	Estação Rádio Base
IoT	Internet das Coisas
LOS	Linha de Visada
NLOS	Não Linha de Visada
NP-difícil	Difícil em tempo Não Polinomial
PSO	<i>Particle Swarm Optimization</i> (Otimização por Enxame de Partículas)
SMP	Serviço Móvel Pessoal
STFC	Serviço Telefônico Fixo Comutado

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Definição do problema	14
1.2	Objetivos	14
1.2.1	Objetivo geral	14
1.2.2	Objetivos específicos	14
1.3	Justificativa	15
1.4	Estrutura do trabalho	16
2	ESTAÇÕES RÁDIO BASE DE TELEFONIA MÓVEL	17
2.1	Tipos de ERBs	17
2.2	Licenciamento de ERBs	19
3	OTIMIZAÇÃO E COMPLEXIDADE COMPUTACIONAL	20
3.1	Otimização combinatória	20
3.2	Tempo de execução de algoritmos	21
3.3	Classificação de problemas computacionais	22
4	ALGORITMOS INSPIRADOS NA NATUREZA	23
4.1	Algoritmo Genético	23
4.1.1	Inspiração biológica	24
4.1.2	Estrutura de um AG	24
4.1.2.1	<i>Seleção</i>	25
4.1.2.2	<i>Cruzamento</i>	25
4.1.2.3	<i>Mutação</i>	26
4.2	Otimização por Enxame de Partículas	26
4.2.1	Inteligência de enxames	26
4.2.2	Estrutura de um PSO	27
5	METODOLOGIA	29
5.1	Cenário de estudo	29
5.2	Modelagem do problema de alocação de ERBs	30
5.2.1	Definição dos pontos de demanda	30
5.2.2	Formulação da função objetivo	31
5.3	Definição do raio de cobertura	32
5.4	Implementação do AG	32
5.4.1	Representação dos indivíduos e população inicial	33
5.4.2	Seleção dos pais e cruzamento	34
5.4.3	Mutação da população	35

5.5	Implementação do PSO	36
5.5.1	Atualização das posições das partículas	37
5.6	Ferramentas utilizadas	38
6	RESULTADOS	40
6.1	Simulações com o AG	40
6.2	Simulações com o PSO	43
6.3	Comparação dos resultados obtidos pelo AG e pelo PSO	45
7	CONCLUSÃO	48
	REFERÊNCIAS	49
	APÊNDICE A – CÓDIGOS DOS ALGORITMOS UTILIZADOS . . .	52
A.1	Código-fonte do AG implementado	52
A.2	Código-fonte do PSO implementado	56
A.3	Código-fonte da discretização de mapas	61

1 INTRODUÇÃO

O cenário atual das telecomunicações é caracterizado por uma demanda crescente por conectividade de alta velocidade, baixa latência e maior capacidade de rede, impulsionada pela proliferação de dispositivos inteligentes, serviços de *streaming*, Internet das Coisas (IoT) e aplicações de realidade virtual e aumentada (SHEN *et al.*, 2025). Nesse contexto, a quinta geração (5G) de redes móveis surge como uma tecnologia fundamental para atender a essas exigências, prometendo transformar a comunicação e impulsionar o desenvolvimento tecnológico em diversas esferas da sociedade (ANATEL, 2022). A tecnologia 5G representa a evolução dos sistemas de rádio, projetada para otimizar a comutação de pacotes e reduzir custos, com requisitos de alta eficiência espectral e altas taxas de pico de dados (ANATEL, 2022; COSTA, 2016).

No Brasil, a implementação do 5G tem sido um marco significativo, com a Agência Nacional de Telecomunicações (ANATEL), realizando leilões de radiofrequência para viabilizar essa tecnologia. O leilão do 5G, que ocorreu em novembro de 2021, foi a maior oferta simultânea de espectro na história da agência. Nesse evento, foram licitadas quatro faixas de radiofrequências: 700 MHz, 2,3 GHz, 3,5 GHz e 26 GHz, distribuídas em lotes e adquiridas por nove operadoras (ANATEL, 2025).

O planejamento e a otimização de redes celulares, especialmente em tecnologias de alta frequência como o 5G, permanecem como um processo de engenharia complexo (ISABONA *et al.*, 2023). A frequência de 3,5 GHz, embora ofereça maior largura de banda para as operadoras, impõe desafios na propagação do sinal, exigindo um número maior de Estações Rádio Base (ERBs) para cobrir uma determinada área quando comparada a frequências mais baixas, como eram utilizadas nas gerações anteriores (ESA; HIKMATUROKHMAN; DANISYA, 2020). Essa característica aumenta substancialmente os custos de implantação e manutenção da rede, tornando imperativa a busca por métodos de planejamento que maximizem a cobertura e a capacidade, ao mesmo tempo em que minimizam os gastos (COSTA, 2016).

Nesse cenário, abordagens tradicionais de planejamento, que dependem de modelos de propagação simplificados ou de processos manuais, tornam-se ineficientes e pouco confiáveis para capturar as complexas características do ambiente de propagação do sinal (YANGYANG *et al.*, 2004; COSTA, 2016). A necessidade de algoritmos de otimização eficientes para o posicionamento de ERBs e a configuração de seus elementos de antena é evidente (ISABONA *et al.*, 2023). Este trabalho se insere nessa lacuna, utilizando Algoritmo Genético (AG) e Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*) para otimizar a alocação de ERBs 5G, focando na maximização da cobertura, com um estudo prático de simulação no município de Paço do Lumiar - MA.

1.1 Definição do problema

O problema de alocação de ERBs em redes de telefonia móvel é inerentemente complexo e classificado como NP-difícil (YANGYANG *et al.*, 2004). Isso significa que não é possível encontrar uma solução ótima em tempo polinomial utilizando métodos determinísticos, especialmente em grandes redes ou ambientes heterogêneos (COSTA, 2016). A complexidade decorre da necessidade de considerar múltiplos fatores e objetivos simultaneamente, que frequentemente são conflitantes. Tais fatores incluem a densidade de tráfego, os cenários de interferência, o número de estações rádio base e outros parâmetros de planejamento de rede (PEREIRA; CAVALCANTI; MACIEL, 2014).

Em essência, o desafio consiste em determinar as posições ideais das ERBs para atender às demandas de cobertura. Para redes 5G, operando na faixa de 3,5 GHz, a atenuação do sinal é mais acentuada, o que implica raios de cobertura menores por ERB e, consequentemente, a necessidade de um número maior de estações para cobrir uma mesma área em comparação com tecnologias anteriores. Este aumento no número de ERBs impacta diretamente o custo total da rede, tornando a otimização da alocação um fator crítico para a viabilidade econômica do projeto (ESA; HIKMATUROKHMANN; DANISYA, 2020; COSTA, 2016).

Diante desse cenário desafiador, este trabalho propõe investigar abordagens alternativas para a resolução do problema de alocação. Assim, a questão de pesquisa central que este trabalho busca responder é: quão eficientes são os algoritmos bioinspirados AG e PSO na alocação de ERBs 5G, na faixa de 3,5 GHz, visando à maximização da cobertura da rede?

1.2 Objetivos

Esta seção apresenta os objetivos deste trabalho monográfico, divididos em objetivo geral e objetivos específicos, que orientam a busca por respostas à questão principal apresentada na seção anterior.

1.2.1 Objetivo geral

Aplicar os algoritmos AG e PSO para otimizar a alocação de ERBs da rede móvel 5G, na faixa de frequência de 3,5 GHz, com o objetivo de maximizar a cobertura do sinal no município de Paço do Lumiar - MA, por meio de simulação computacional.

1.2.2 Objetivos específicos

1. Estimar o raio de alcance de uma ERB 5G com base em estudos anteriores que utilizaram modelos de propagação apropriados para a frequência de 3,5 GHz;

2. Implementar os algoritmos AG e PSO ao problema de alocação de ERBs utilizando a linguagem Python;
3. Simular a cobertura do sinal 5G no município de Paço do Lumiar – MA, com base nos resultados obtidos pelos algoritmos, visando à sua maximização;
4. Determinar o número mínimo de ERBs necessário para atender à demanda de cobertura no local de estudo, buscando a redução dos custos de implantação da rede;
5. Comparar os resultados e o desempenho dos algoritmos AG e PSO na resolução do problema de alocação de ERBs.

1.3 Justificativa

A comunicação é a base das relações humanas e impulsiona o desenvolvimento social e econômico (COSTA, 2016). No contexto atual, a tecnologia 5G se destaca como um pilar fundamental para o futuro das telecomunicações, proporcionando velocidades superiores, menor latência e uma utilização mais eficiente do espectro (ANATEL, 2022). Essa evolução é essencial para atender ao crescente número de usuários e à demanda por novas aplicações que dependem da robustez e da ampla cobertura das redes móveis (SHEN *et al.*, 2025).

Entretanto, a implantação de redes 5G representa um desafio significativo e um investimento considerável para as operadoras. Diferentemente das gerações anteriores, as frequências mais altas do 5G têm menor capacidade de penetração e alcance, exigindo, consequentemente, uma densidade muito maior de ERBs para garantir uma cobertura adequada (SAPKOTA *et al.*, 2024). Esse fator impacta diretamente os custos de capital e operacionais, tornando o planejamento de rede um processo que não pode mais ser baseado apenas em métodos tradicionais.

A otimização da localização das ERBs é, portanto, uma etapa crítica para o sucesso e a sustentabilidade das redes 5G. Um planejamento ineficiente pode acarretar em zonas de sombra (áreas sem cobertura), baixa qualidade de serviço, interferência excessiva entre estações, uso inadequado de recursos e, em última instância, perdas financeiras significativas para os investidores e insatisfação dos usuários (ISABONA *et al.*, 2023). A ANATEL (2021) estabelece, por meio dos compromissos de abrangência do leilão do 5G, requisitos mínimos de cobertura para municípios, sendo de pelo menos 95% da área urbana do distrito sede do município, o que reforça a necessidade de um planejamento preciso.

A utilização de algoritmos bioinspirados, como o AG e o PSO, é uma abordagem promissora para solucionar esse problema complexo e multiobjetivo. Esses algoritmos são capazes de explorar um vasto espaço de soluções e encontrar configurações que equilibram a maximização da cobertura com a minimização dos custos, superando as limitações dos métodos

heurísticos tradicionais. Estudos anteriores, como Costa (2016), Pei *et al.* (2022), Sapkota *et al.* (2024) e Gu, Liu e Li (2009), já demonstraram a eficácia dessas técnicas no planejamento de redes 3G e 4G, e sua adaptabilidade a cenários 5G é um campo de pesquisa relevante.

Este trabalho, ao realizar uma simulação computacional no município de Paço do Lumiar – MA, contribui de forma prática ao aplicar essas metodologias a um contexto geográfico real. Os resultados obtidos podem fornecer informações valiosas para operadoras de telefonia móvel, auxiliando na tomada de decisões estratégicas para a implantação e expansão eficiente de suas redes 5G. Além disso, a comparação entre os algoritmos AG e PSO proporcionará uma compreensão mais aprofundada sobre a performance de cada algoritmo diante dos desafios da alocação de ERBs, pavimentando o caminho para futuras pesquisas na área. Dessa forma, a pesquisa não apenas avança o conhecimento acadêmico, mas também oferece ferramentas e informações diretamente aplicáveis para otimizar o uso de recursos e melhorar a experiência dos usuários no ambiente 5G.

1.4 Estrutura do trabalho

Esta monografia está organizada em sete seções, nas quais:

Na seção 2, apresentam-se os conceitos fundamentais relacionados às ERBs, abordando seus tipos e o processo de licenciamento.

A seção 3 discute os princípios de otimização e complexidade computacional, destacando a relevância das técnicas heurísticas para a resolução de problemas de grande porte.

Na seção 4, são detalhados os algoritmos AG e PSO, com ênfase em seus fundamentos teóricos e estruturas de funcionamento.

A seção 5 descreve a metodologia aplicada, contemplando a definição do cenário de estudo, a modelagem do problema de alocação das ERBs, a formulação da função objetivo e a implementação computacional dos algoritmos.

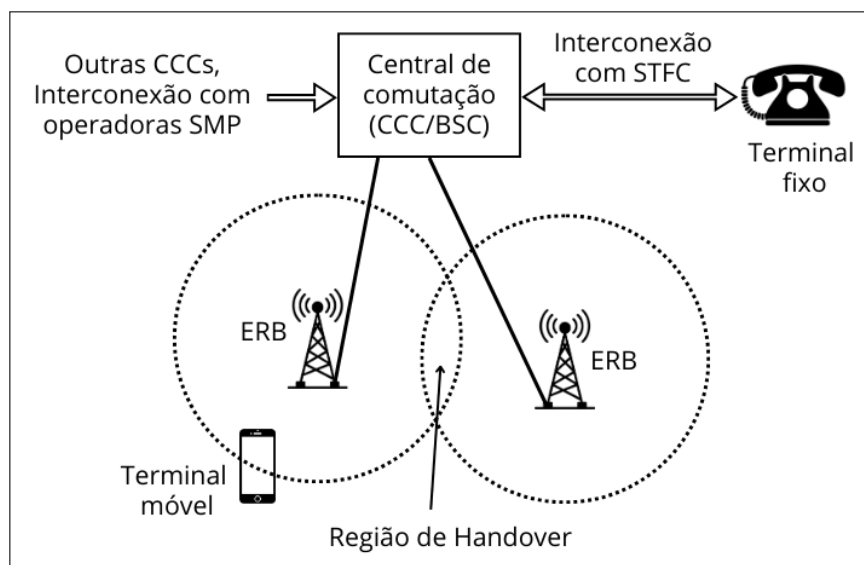
Na seção 6, apresentam-se os resultados obtidos nas simulações, acompanhados de análises comparativas entre o desempenho do AG e do PSO.

Por fim, a seção 7 reúne as conclusões do estudo, evidenciando as principais contribuições, limitações e possíveis direções para pesquisas futuras.

2 ESTAÇÕES RÁDIO BASE DE TELEFONIA MÓVEL

As Estações Rádio Base (ERBs) constituem elementos fundamentais na infraestrutura da telefonia móvel, pois viabilizam a comunicação entre os terminais móveis e a rede central. Em termos funcionais, uma ERB é conectada a uma Central de Comutação e Controle (CCC), responsável pela interconexão com o Serviço Telefônico Fixo Comutado (STFC) e com outras centrais, permitindo a realização de chamadas entre diferentes redes. Em determinadas arquiteturas de sistemas celulares, a ligação entre a ERB e a CCC pode ocorrer por meio de um *Base Station Controller* (BSC), que gerencia um conjunto de ERBs de forma centralizada (TELECO, 2009). Essa arquitetura de comunicação é ilustrada na Figura 1.

Figura 1 – Arquitetura simplificada de uma rede de telefonia móvel



Fonte: adaptado de Teleco (2009).

A função principal da ERB é prover cobertura em uma área delimitada, denominada célula, possibilitando a mobilidade do usuário. Durante a movimentação entre diferentes células, o serviço de comunicação é garantido pelo processo de *handoff* ou *handover*, no qual o terminal móvel transfere sua conexão de uma ERB para outra de forma contínua. A capacidade de cada célula é limitada pela quantidade de canais de rádio disponíveis, sendo necessário um planejamento criterioso para o reuso das frequências e a prevenção de interferências. As ERBs são compostas por transmissores, receptores e antenas, estas últimas instaladas em torres, mastros ou postes, de acordo com a área de cobertura desejada (MEDEIROS, 2016).

2.1 Tipos de ERBs

As ERBs podem ser classificadas em dois tipos principais: *Greenfield*, instaladas no solo (conforme a Figura 2), e *Rooftop*, localizadas em coberturas de edifícios (conforme a

Figura 3). Ambas podem operar com equipamentos *indoor*, que necessitam de infraestrutura de climatização, ou com equipamentos *outdoor*, projetados para resistir à exposição direta ao ambiente externo (TELECO, 2009).

Figura 2 – ERB do tipo *Greenfield*



Fonte: Teleco (2009).

Figura 3 – ERB do tipo *Rooftop*



Fonte: elaborada pelo autor (2025).

A cobertura fornecida por uma ERB não apresenta formato ou área uniformes, podendo variar significativamente conforme fatores geográficos e de densidade populacional. Em áreas urbanas e suburbanas, a cobertura demanda a instalação de múltiplas ERBs, enquanto em rodovias ou regiões rurais é necessário o planejamento de uma infraestrutura distribuída para assegurar conectividade em locais de menor densidade. A disposição e o tipo de ERB a serem implementados são, portanto, decisivos para a eficiência e qualidade da rede de telefonia móvel (MEDEIROS, 2016).

2.2 Licenciamento de ERBs

O licenciamento das ERBs representa uma etapa fundamental para sua regularização e operação (TELECO, 2009). De acordo com a ANATEL (2020), apenas entidades autorizadas, permissionárias ou concessionárias de serviços de telecomunicações podem obter o licenciamento de estações, incluindo aquelas destinadas ao Serviço Móvel Pessoal (SMP). Esse processo ocorre após a outorga do serviço e é indispensável para o funcionamento regular das estações no território nacional.

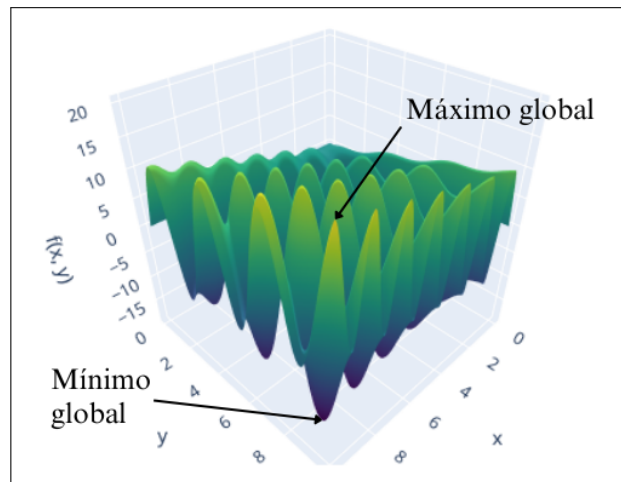
Adicionalmente, o licenciamento está condicionado ao atendimento de requisitos legais relacionados à exposição humana a campos eletromagnéticos. A Lei nº 11.934/2009 estabelece limites e procedimentos a serem observados no projeto, implantação, homologação e operação das ERBs, assegurando que os níveis de radiação emitidos permaneçam dentro dos padrões de segurança estabelecidos. Dessa forma, o processo de licenciamento não apenas legitima a operação das estações, mas também garante a proteção da saúde pública e a conformidade técnica das instalações (TELECO, 2009).

3 OTIMIZAÇÃO E COMPLEXIDADE COMPUTACIONAL

Segundo Haupt e Haupt (2004), o ato de projetar envolve escolhas que buscam reduzir custos ou melhorar determinadas características de um produto, como o peso ou a eficiência. Nesse contexto, a otimização pode ser entendida como o processo de ajustar parâmetros de um dispositivo, modelo matemático ou experimento de forma a alcançar o melhor desempenho possível. Esse processo baseia-se na manipulação de variáveis de entrada, avaliadas por meio de uma função que pode ser denominada **função de custo**, **função objetivo** ou **função de aptidão**. O resultado esperado é a obtenção de valores que representem a minimização ou maximização desejada, dependendo do problema em questão. Quando aplicada a experimentos, a otimização refere-se diretamente ao controle das variáveis físicas que compõem o sistema analisado.

De acordo com Gaspar-Cunha, Takahashi e Antunes (2012), a otimização constitui um campo de estudo voltado para a identificação dos extremos (máximos ou mínimos) de funções dentro de um domínio previamente definido (conforme a Figura 4). O valor ótimo encontrado representa a melhor forma de estruturar o sistema em análise, podendo significar, por exemplo, um projeto de menor custo, maior eficiência energética ou menor taxa de falhas. O espaço de busca, nesse contexto, é formado pelo conjunto de alternativas possíveis disponíveis para a construção ou modelagem do sistema, no qual se procura localizar a solução que atenda aos objetivos estabelecidos de forma mais eficaz.

Figura 4 – Extremos de uma função



Fonte: adaptado de Haupt e Haupt (2004).

3.1 Otimização combinatória

Quando todas as variáveis de um problema assumem valores discretos, tem-se a chamada **otimização combinatória**, uma subárea da otimização. Nesse contexto, diferentes abordagens

são utilizadas para a busca de soluções. Um **algoritmo** é entendido como um procedimento capaz de garantir a obtenção do ótimo exato da função objetivo em um número finito de etapas. Já um **método de aproximação** assegura que, após um número finito de passos, será encontrada uma solução próxima do ótimo exato, dentro de um limite de erro previamente estabelecido. Por outro lado, as **heurísticas** diferenciam-se por não oferecerem garantias formais quanto à proximidade em relação ao ótimo, mas buscam soluções satisfatórias em tempo viável, sendo amplamente empregadas em problemas de grande porte (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012).

O principal desafio da otimização combinatória está no crescimento exponencial do número de alternativas à medida que aumenta a quantidade de variáveis. Embora o exame completo de todas as combinações possíveis sempre conduza ao ótimo exato, esse tipo de abordagem se torna inviável para problemas de maior dimensão, pois o crescimento exponencial do espaço de busca rapidamente ultrapassa as capacidades computacionais disponíveis. Assim, a questão central deixa de ser a existência de um algoritmo que resolva o problema e passa a ser a viabilidade de encontrar soluções em um tempo computacional aceitável. Nesse cenário, heurísticas e métodos aproximados desempenham um papel primordial, equilibrando qualidade da solução e eficiência no processamento (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012).

3.2 Tempo de execução de algoritmos

A análise do tempo de execução de algoritmos constitui um aspecto substancial na avaliação de desempenho computacional. Mais do que obter a medida exata do tempo gasto em um ambiente específico, busca-se compreender como o tempo de processamento cresce à medida que aumenta o volume de dados de entrada. Essa abordagem possibilita a definição de métricas que permanecem válidas independentemente do *hardware*, do sistema operacional ou do compilador utilizado, permitindo uma avaliação mais geral e comparativa entre diferentes algoritmos. Dessa forma, é possível identificar qual solução apresenta maior eficiência em termos de tempo de execução e, em alguns casos, também em relação ao consumo de recursos computacionais (LINDEN, 2008).

Entre as formas de mensuração do desempenho, destacam-se o tempo de execução no **pior caso**, no **caso médio** e no **melhor caso**. Dentre essas métricas, a análise do pior caso é a mais relevante, pois fornece um limite superior para o tempo necessário à execução do algoritmo em qualquer situação. Tal medida garante maior previsibilidade, uma vez que, se o desempenho for considerado aceitável no pior cenário, ele será satisfatório em todas as demais condições possíveis. Essa perspectiva torna-se essencial em problemas complexos, nos quais a eficiência e a escalabilidade dos algoritmos desempenham papel determinante na viabilidade prática de sua aplicação (LINDEN, 2008).

3.3 Classificação de problemas computacionais

A classificação de problemas computacionais está diretamente relacionada à sua complexidade e ao tempo requerido para encontrar soluções viáveis. Segundo Linden (2008), um problema é considerado tratável quando pode ser resolvido em tempo polinomial, ou seja, quando o número de operações necessárias para a resolução cresce de forma controlada em função do tamanho da entrada. Em contrapartida, problemas cuja resolução demanda tempo exponencial ou fatorial são classificados como intratáveis, pois exigem um esforço computacional impraticável diante das necessidades reais de usuários e aplicações.

Um exemplo clássico desse tipo de problema é o Problema do Caixeiro Viajante, no qual se busca determinar a rota de menor custo para visitar todas as cidades de um conjunto. Embora, em teoria, seja possível obter a solução ótima por meio da enumeração exaustiva de todas as alternativas, tal procedimento torna-se inviável na prática devido ao crescimento exponencial do espaço de busca. Esse problema não se restringe ao exemplo do problema do caixeiro, mas encontra aplicações diretas em logística, transporte e redes de comunicação, configurando-se como um desafio intratável com forte impacto prático (LINDEN, 2008).

Nesse contexto, a literatura em complexidade computacional formaliza diferentes classes de problemas. De acordo com Gaspar-Cunha, Takahashi e Antunes (2012), a **classe P** reúne os problemas de decisão que podem ser resolvidos em tempo polinomial por uma máquina de Turing determinística. Já a **classe NP** (não deterministicamente polinomial) compreende problemas de decisão para os quais, embora não exista garantia da obtenção de uma solução exata em tempo polinomial, qualquer solução candidata pode ser verificada de forma eficiente nesse limite. Contudo, a verificação exaustiva de todas as possibilidades de solução, em geral, demandaria tempo exponencial. Entre os problemas de maior relevância teórica e prática estão os **NP-completos**, para os quais não existem algoritmos conhecidos de tempo polinomial e acredita-se que tais algoritmos talvez sequer possam existir.

Há ainda os chamados problemas **NP-difíceis**, definidos como pelo menos tão complexos quanto os NP-completos a eles relacionados. Muitos problemas de otimização pertencem a essa categoria, já que encontrar a solução ótima pode exigir um número inatingível de operações. Diante dessa realidade, heurísticas e métodos aproximados se tornam alternativas indispensáveis, uma vez que, em contraste com os algoritmos exatos de complexidade exponencial, permitem obter soluções satisfatórias em prazos compatíveis com as demandas práticas (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012).

4 ALGORITMOS INSPIRADOS NA NATUREZA

Os algoritmos inspirados na natureza surgem como ferramentas poderosas para resolver problemas complexos de otimização em espaços de busca amplos e, muitas vezes, não lineares. Diferentemente de métodos tradicionais baseados em gradiente, essas técnicas não exigem a derivação da função objetivo, o que as torna adequadas para lidar com variáveis discretas, funções não contínuas e problemas de difícil modelagem analítica (HAUPT; HAUPT, 2004).

Sua lógica de funcionamento consiste em explorar o espaço de soluções por meio de operadores que geram novas possibilidades a partir das soluções atuais, conduzindo o processo de busca de maneira probabilística em direção a regiões mais promissoras. Assim, conseguem combinar estratégias de busca local (exploração mais detalhada de uma região) com busca global (investigação de diferentes áreas do espaço de soluções), aumentando a probabilidade de encontrar soluções próximas ao ótimo global (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012).

Esses métodos são classificados como **meta-heurísticas**, ou seja, abordagens estocásticas e de propósito geral, que diferem das heurísticas clássicas por não se limitarem a problemas específicos. Fundamentam-se em conceitos genéricos, como vizinhança de pontos, trajetórias de busca e organização das informações adquiridas ao longo das iterações, o que lhes garante aplicabilidade em diferentes contextos (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012).

Entre os algoritmos meta-heurísticos mais consolidados, destacam-se o AG e o PSO, ambos inspirados em fenômenos naturais de grande eficiência: a evolução biológica e o comportamento coletivo de enxames (TAVARES; NEDJAH; MOURELLE, 2015).

4.1 Algoritmo Genético

O Algoritmo Genético (AG) é uma técnica de otimização e busca inspirada nos princípios da evolução biológica, em especial nos mecanismos de genética e seleção natural. Por meio dele, uma população formada por diversos indivíduos é conduzida, ao longo de sucessivas gerações, por regras de seleção que favorecem as soluções mais aptas, ou seja, aquelas que minimizam ou maximizam a função objetivo. Esse método foi desenvolvido por John Holland nas décadas de 1960 e 1970, e posteriormente ganhou notoriedade com seu aluno David Goldberg, que o aplicou com sucesso na resolução de um problema complexo relacionado ao controle de transmissão em gasodutos em sua dissertação (HAUPT; HAUPT, 2004).

4.1.1 Inspiração biológica

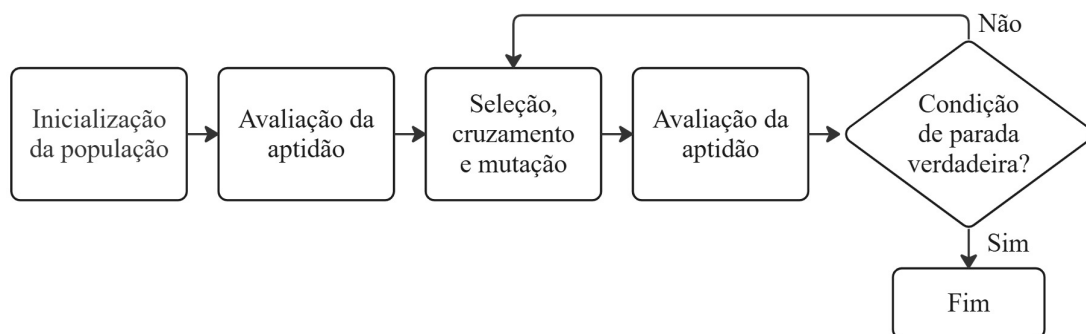
A evolução biológica pode ser entendida como um processo de busca em um vasto e complexo espaço de combinações genéticas, no qual apenas algumas resultam em organismos viáveis. Esse mecanismo natural favorece, ao longo do tempo, a sobrevivência e reprodução dos indivíduos mais adaptados ao ambiente, permitindo que suas características se propaguem por gerações. Esse ciclo contínuo de seleção e variação genética possibilita a adaptação gradual das populações ao meio em que vivem (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012).

Com o avanço da genética no início do século XX, tornou-se possível compreender de forma mais clara como essas variações ocorrem. A incorporação de conceitos como recombinação e mutação genética resultou na chamada teoria Neodarwinista, que combina os princípios de Darwin com a genética Mendeliana. Essa visão moderna explica como a diversidade genética e os operadores de variação são essenciais para o processo evolutivo, servindo de inspiração para o funcionamento dos AGs (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012).

4.1.2 Estrutura de um AG

O AG opera sobre um conjunto de soluções candidatas, usualmente designado por **população**, que é iterativamente transformado ao longo de gerações. Parte-se de várias soluções iniciais escolhidas aleatoriamente e, em cada ciclo, os indivíduos mais aptos são selecionados (tipicamente por um critério probabilístico) para atuar como progenitores. Em seguida, aplicam-se operadores genéticos para produzir novos candidatos, buscando aproximar a população de uma solução de alta qualidade para o problema (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012; TAVARES; NEDJAH; MOURELLE, 2015). O fluxograma da Figura 5 fornece uma visão geral do AG.

Figura 5 – Fluxograma do AG



Fonte: adaptado de Tavares, Nedjah e Mourelle (2015).

Para aplicar um AG com eficácia, duas decisões são fundamentais: a representação das soluções no espaço de busca e a definição da função de aptidão, que mede a qualidade de cada

candidato. Na terminologia dos AGs, cada iteração corresponde a uma **geração**; os elementos da população são chamados **indivíduos** ou **cromossomos**; e seus componentes básicos, **os genes**, representam características elementares da solução, ocupam posições específicas ou **locus** e podem assumir valores em um conjunto de possibilidades, denominadas **alelos** (LINDEN, 2008; GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012).

Ao longo das gerações, a seleção tende a elevar a qualidade média da população, enquanto os operadores genéticos (**cruzamento** e **mutação**) geram novas soluções e preservam diversidade suficiente para evitar estagnação. Os detalhes de implementação desses operadores dependem diretamente da representação escolhida, por isso, a combinação entre representação, aptidão e operadores deve ser cuidadosamente projetada para maximizar a eficácia do processo de otimização (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012). As seções seguintes apresentam a descrição do processo de seleção e dos operadores genéticos.

4.1.2.1 Seleção

O processo de seleção é responsável por escolher os indivíduos que irão gerar os descendentes da próxima geração. Indivíduos com maior aptidão têm mais chances de serem selecionados, o que garante que suas características sejam transmitidas, aumentando gradualmente a qualidade média da população ao longo das gerações (TAVARES; NEDJAH; MOURELLE, 2015).

Entre os principais métodos de seleção, destaca-se a **Roleta Viciada**, em que cada indivíduo ocupa uma fatia proporcional à sua aptidão em uma espécie de “roleta”. A seleção é realizada repetidamente, com reposição, até compor a nova população. Já o **Elitismo** assegura que o melhor indivíduo de cada geração seja automaticamente mantido na próxima, evitando a perda de soluções de alta qualidade durante o processo evolutivo (LINDEN, 2008).

Outro método bastante utilizado é a **Seleção por Torneio**, que compara um subconjunto aleatório de indivíduos e escolhe o melhor para reprodução. Esse método não depende diretamente dos valores absolutos da aptidão, apenas de uma hierarquização relativa. Além disso, o tamanho do torneio é um parâmetro que permite controlar a pressão seletiva: torneios maiores aumentam a probabilidade de seleção dos melhores indivíduos, enquanto torneios menores promovem maior diversidade (GASPAR-CUNHA; TAKAHASHI; ANTUNES, 2012).

4.1.2.2 Cruzamento

O cruzamento, também chamado de **recombinação**, é um operador estocástico responsável por promover a troca de informações genéticas entre dois indivíduos selecionados da população. A partir desse processo, são gerados novos descendentes que herdam partes das

sequências genéticas de cada progenitor, combinando características distintas em uma nova solução (LINDEN, 2008; GASPARG-CUNHA; TAKAHASHI; ANTUNES, 2012).

Esse operador é considerado um dos mecanismos centrais de exploração do espaço de busca nos AGs, pois possibilita que soluções promissoras da população atual sejam recombinadas, potencializando a descoberta de indivíduos com maior aptidão. O princípio fundamental é simples: ao reunir informações complementares de diferentes indivíduos, o cruzamento aumenta a chance de originar soluções mais adequadas ao problema em análise (GASPARG-CUNHA; TAKAHASHI; ANTUNES, 2012).

4.1.2.3 Mutação

A mutação é um operador aplicado sobre os indivíduos resultantes do processo de cruzamento, introduzindo pequenas alterações aleatórias em suas características. Esse mecanismo desempenha um papel fundamental na manutenção da diversidade genética da população, evitando que o algoritmo fique restrito a um conjunto limitado de soluções. No AG, a mutação é essencial para permitir o reaparecimento de características que poderiam ser perdidas ao longo das gerações, ampliando a capacidade de exploração do espaço de busca. Embora seja um processo aleatório, sua função estratégica está em garantir que a evolução da população não convirja prematuramente para soluções locais, preservando a variabilidade necessária para encontrar soluções de maior qualidade (GASPARG-CUNHA; TAKAHASHI; ANTUNES, 2012).

4.2 Otimização por Enxame de Partículas

O algoritmo *Particle Swarm Optimization* (PSO) foi proposto por Kennedy e Eberhart (1995), tendo como inspiração o comportamento coletivo observado em sistemas naturais, como o movimento coordenado de bandos de pássaros ou cardumes de peixes. Assim como o AG, o PSO inicia sua busca a partir de uma população gerada aleatoriamente. Entretanto, a principal diferença em relação ao AG está no fato de que o PSO não utiliza operadores evolutivos, como cruzamento e mutação. Nesse método, cada indivíduo da população é denominado **partícula**, e sua dinâmica é guiada pelo compartilhamento de informações entre os membros do enxame, permitindo que ajustem suas trajetórias em direção a regiões mais promissoras do espaço de busca (HAUPT; HAUPT, 2004; KENNEDY; EBERHART, 1995).

4.2.1 Inteligência de enxames

A inteligência de enxames descreve o modo como grupos de organismos simples conseguem, a partir de interações locais e descentralizadas, realizar comportamentos coletivos eficientes. Em diversas situações na natureza, a movimentação em grupo garante vantagens

evolutivas, como maior proteção contra predadores e maior eficiência na busca por recursos. Embora bandos de presas possam ser mais visíveis do que indivíduos isolados, o deslocamento coletivo reduz o risco de ataque a cada membro e aumenta a probabilidade de sobrevivência (KENNEDY; EBERHART; SHI, 2001).

Além disso, a vida em grupo favorece a troca de informações sobre o ambiente. Animais sociais, como aves, conseguem identificar fontes de alimento observando o comportamento de seus companheiros, por exemplo, ao seguir a direção de onde um pássaro retorna alimentado ou ao notar vários indivíduos convergindo para o mesmo ponto. Em insetos sociais, como formigas e abelhas, esse princípio se manifesta por meio da divisão de trabalho, em que diferentes membros do enxame desempenham papéis complementares. Dessa forma, o comportamento coletivo se mostra mais vantajoso do que a ação individual, servindo de inspiração para algoritmos como o PSO (KENNEDY; EBERHART; SHI, 2001).

4.2.2 Estrutura de um PSO

No PSO, cada partícula representa uma solução potencial para o problema em análise. O comportamento dessas partículas é inspirado na dinâmica de bandos de pássaros em busca de alimento, combinando dois tipos de aprendizado: o **cognitivo**, referente às experiências individuais, e o **social**, derivado da interação com o grupo. O termo partícula é utilizado em analogia à física, por possuir uma posição e uma velocidade bem definidas, ainda que não tenha massa nem volume. Já o **enxame** corresponde ao conjunto completo de partículas, ou seja, ao espaço de soluções explorado coletivamente (TAVARES; NEDJAH; MOURELLE, 2015).

A evolução do PSO ocorre de forma iterativa, em que a posição de cada partícula é atualizada a partir de seu **vetor velocidade**, conforme a Figura 6. Esse vetor resulta da soma de três componentes: inércia, que mantém a partícula seguindo a mesma trajetória anterior; memória, que a direciona para a melhor posição já encontrada por ela mesma; e cooperação, que a atrai para a melhor posição identificada pelo enxame como um todo. Assim, o equilíbrio entre exploração individual e social permite ao PSO buscar soluções de forma eficiente (TAVARES; NEDJAH; MOURELLE, 2015; HAUPT; HAUPT, 2004). O cálculo da velocidade é realizado pela Equação 1, enquanto a atualização da posição de cada partícula é dada pela Equação 2:

$$v_i^{(t+1)} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (p_i^{local\ best} - p_i^t) + c_2 \cdot r_2 \cdot (p^{global\ best} - p_i^t) \quad (1)$$

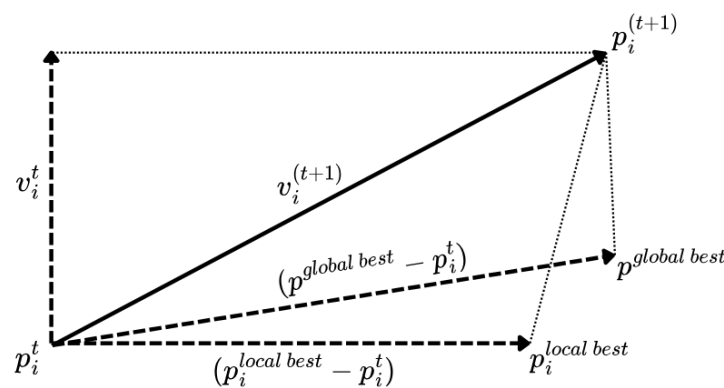
$$p_i^{(t+1)} = p_i^t + v_i^{(t+1)} \quad (2)$$

onde:

- $v_i^{(t+1)}$ é a velocidade da partícula i no tempo $t + 1$;
- $p_i^{(t+1)}$ é a posição da partícula i no tempo $t + 1$;

- w é a inércia da partícula;
- r_1 e r_2 são números aleatórios entre 0 e 1;
- c_1 e c_2 são constantes que representam os pesos atribuídos às componentes cognitiva e social, respectivamente (usualmente $c_1 = c_2 = 2$);
- $p_i^{local\ best}$ é a melhor posição encontrada pela partícula i ;
- $p^{global\ best}$ é a melhor posição encontrada pelo enxame.

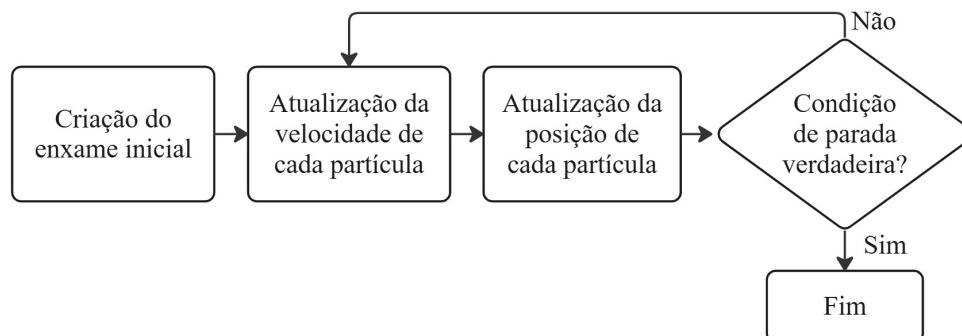
Figura 6 – Atualização da posição da partícula no espaço vetorial



Fonte: adaptado de Alam (2016).

As posições das partículas são restritas ao espaço de busca definido para o problema. O processo iterativo do PSO é executado repetidamente até que um critério de parada seja alcançado, o qual pode estar associado ao número máximo de iterações, à obtenção de um valor satisfatório da função objetivo ou a outra condição previamente estabelecida (TAVARES; NEDJAH; MOURELLE, 2015). A dinâmica desse processo é representada no fluxograma da Figura 7.

Figura 7 – Fluxograma do PSO



Fonte: adaptado de Tavares, Nedjah e Mourelle (2015).

5 METODOLOGIA

Esta seção apresenta a metodologia adotada para a resolução do problema de alocação das ERBs de tecnologia 5G. O objetivo é identificar as localizações mais adequadas para a instalação das ERBs, de modo a maximizar a cobertura de sinal na área de interesse. Para isso, foram implementados dois algoritmos de otimização inspirados em processos naturais: o AG e o PSO, ambos adaptados para considerar as particularidades do problema abordado. Além disso, esta seção descreve as etapas de definição do cenário de estudo, a estimativa do raio de alcance das ERBs 5G, a modelagem do problema e detalha a implementação dos algoritmos.

5.1 Cenário de estudo

De acordo com o IBGE (2025), o município de Paço do Lumiar, localizado no estado do Maranhão, possui uma área territorial de 127,193 km² (dados de 2024), sendo aproximadamente 30% dessa área urbanizada (dados de 2019). Trata-se de um território consideravelmente menor quando comparado à área de 583,063 km² do município vizinho, São Luís, que inicialmente havia sido considerado como região de estudo. No entanto, conforme discutido na seção 3, quanto menor o espaço de busca, menor tende a ser o tempo de processamento do algoritmo. Por esse motivo, optou-se pela escolha de Paço do Lumiar como área de análise.

Figura 8 – Polígono territorial de Paço do Lumiar



Fonte: elaborada pelo autor (2025).

Na Figura 8, observa-se o polígono territorial do município utilizado nas simulações, com área territorial calculada de aproximadamente 132 km².

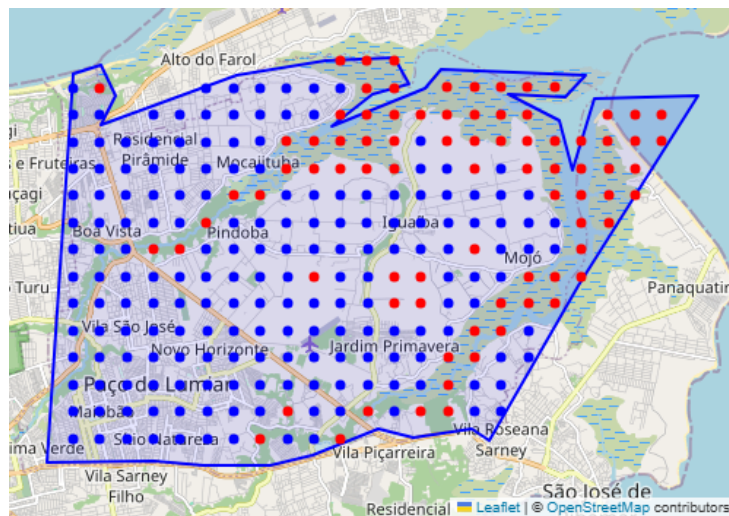
5.2 Modelagem do problema de alocação de ERBs

O problema de alocação de ERBs envolve a definição estratégica de suas localizações dentro de uma área geográfica específica, de modo a garantir a cobertura eficiente da rede (COSTA, 2016). O objetivo é assegurar que todos os pontos da região com demanda por serviço móvel estejam devidamente atendidos, considerando que usuários podem estar distribuídos em qualquer local dessa área.

5.2.1 Definição dos pontos de demanda

Para viabilizar a modelagem computacional do problema, a área de interesse é discretizada por meio da inserção de pontos de demanda. Esses pontos representam possíveis localizações onde usuários podem estar presentes e, portanto, exigem cobertura de sinal. A distância entre os pontos, denotada por d_{pontos} , determina a resolução da discretização. Valores menores de d_{pontos} resultam em uma representação mais precisa da região, mas também elevam significativamente o custo computacional da simulação (COSTA, 2016).

Figura 9 – Discretização do mapa



Fonte: elaborada pelo autor (2025).

Na Figura 9, ilustra-se a discretização do mapa de Paço do Lumiar, utilizando um espaçamento entre os pontos igual a 700 metros ($d_{\text{pontos}} = 700 \text{ m}$). Os pontos em azul (184 pontos) representam as localizações onde há demanda por serviço móvel, enquanto os pontos em vermelho (82 pontos) indicam áreas que foram descartadas por estarem fora da região de interesse.

5.2.2 Formulação da função objetivo

O objetivo do problema é posicionar um conjunto N de ERBs de modo que todo o conjunto M de pontos de demanda seja adequadamente atendido (COSTA, 2016). Trata-se, portanto, de um problema de natureza combinatória, dada a grande quantidade de possíveis combinações de posicionamento (YANGYANG *et al.*, 2004).

Dessa forma, Costa (2016) propôs a seguinte formulação para a função objetivo:

$$\text{maximizar } f = \sum_{i=1}^m (M_i) \cdot \frac{100}{m} \quad (3)$$

sujeito a:

$$m > n \quad (4)$$

$$M_i \in \{0, 1\}, i = 1, \dots, m \quad (5)$$

$$d_{ij} = \text{dist}(M_i, N_j), i = 1, \dots, m; j = 1, \dots, n \quad (6)$$

$$M_i = \{1 \Leftrightarrow d_{ij} < R\} \vee \{0 \Leftrightarrow d_{ij} \geq R\} \quad (7)$$

onde:

- M é o conjunto dos pontos de demanda, onde cada ponto i assume valor 1 se for atendido por ao menos uma ERB, ou 0 caso contrário;
- m é a quantidade de pontos de demanda;
- N é o conjunto de ERBs a serem instaladas;
- n é a quantidade de ERBs a serem instaladas;
- d_{ij} é a distância entre o ponto de demanda M_i e a ERB N_j ;
- R é o raio de cobertura da ERB.

A função objetivo f é definida como o percentual de pontos de demanda cobertos por, no mínimo, uma ERB. Assim, quanto maior a quantidade de pontos atendidos, melhor será considerada a qualidade da solução. A solução ótima corresponde ao cenário em que 100% dos pontos de demanda são contemplados pela cobertura da rede (COSTA, 2016).

As equações 4, 5, 6 e 7 são restrições da função objetivo, estabelecida a partir da Equação 3. A restrição 4 define que a quantidade de pontos de demanda deve ser maior que a quantidade de ERBs a serem instaladas. A restrição 5 define que os valores aceitos para M_i são 0 ou 1. As restrições 6 e 7 indicam que um ponto de demanda é considerado atendido quando a distância entre ele e uma ERB é menor que o raio de alcance da ERB. Assim, se o ponto de demanda M_i for atendido por pelo menos uma ERB, seu valor será 1; caso não seja atendido por nenhuma ERB, seu valor será 0 (COSTA, 2016).

5.3 Definição do raio de cobertura

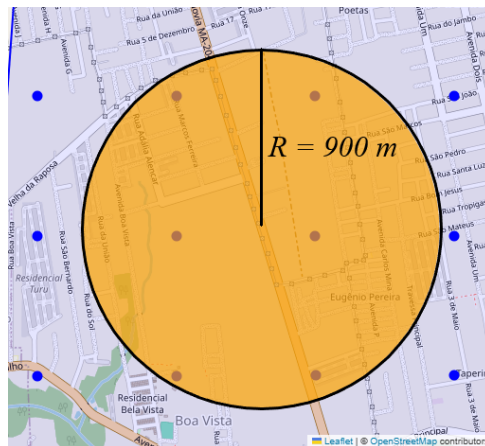
Neste trabalho, a estimativa do raio de cobertura da ERB 5G é fundamental para a simulação de otimização da rede. Considera-se a operação na faixa de 3,5 GHz e avalia-se o alcance com base em estudos prévios que adotam modelos de propagação apropriados.

Segundo Biswas e Lempiainen (2021), simulações baseadas em *ray tracing* indicam que, para largura de banda de 20 MHz e em linha de visada (LOS), uma ERB 5G pode atingir até 1,2 km em ambiente macrocelular. Com menor largura de banda (1 MHz), esse alcance pode ultrapassar 5 km, porém, larguras mais altas resultam em menor alcance devido ao aumento do limiar de recepção.

Já Halvarsson *et al.* (2018), em testes reais com largura de banda de 80 MHz e tecnologia de *beamforming*, observaram *throughput* elevado (acima de 700 Mbps) até 700 metros em linha de visada e cobertura funcional (até 200 Mbps) em distâncias de até 380 metros em não linha de visada (NLOS), em ambientes urbanos densos.

Dessa forma, considerando as condições típicas de operação com 20 a 80 MHz de largura de banda e em linha de visada, estima-se que o raio de cobertura de uma ERB 5G na frequência de 3,5 GHz esteja compreendido entre 700 metros e 1,2 km. Com base nesse intervalo, adotou-se, para fins de simulação, um valor intermediário de 900 metros como raio fixo de alcance do sinal, cobrindo uma área de aproximadamente 2,54 km², conforme ilustrado na Figura 10.

Figura 10 – Cobertura estimada de uma ERB 5G



Fonte: elaborada pelo autor (2025).

5.4 Implementação do AG

No Algoritmo 1, apresenta-se o pseudocódigo do AG implementado para resolver o problema de alocação de ERBs. As linhas 1 e 4, responsáveis pela definição dos pontos de

demanda e da função objetivo, foram discutidas na seção 5.2. A linha 2, que estabelece o raio de cobertura das ERBs, foi abordada na seção 5.3. A quantidade de ERBs por indivíduo, especificada na linha 3, foi variada nas simulações, que serão apresentadas na seção 6, com o objetivo de determinar o número mínimo necessário para alcançar a cobertura máxima. Os parâmetros definidos na linha 5 são: tamanho da população igual a 16 indivíduos, taxa de mutação de 20%, taxa de seleção de 50% e número máximo de gerações igual a 300. Por fim, os procedimentos a partir da linha 6 são detalhados nas seções subsequentes.

Algoritmo 1 – Pseudocódigo do AG para alocação de ERBs 5G

```

1: Carregar os pontos de demanda ( $M$ )
2: Definir o raio de cobertura ( $R$ )
3: Definir a quantidade de ERBs por indivíduo ( $N$ )
4: Definir a função objetivo  $f$  (Equação 3)
5: Definir parâmetros (tamanho da população, taxa de mutação, taxa de seleção e gerações ( $G_{max}$ ))
6: Gerar a população inicial
7: Avaliar a cobertura da população inicial por meio da função  $f$ 
8: Classificar a população em ordem decrescente de acordo com a cobertura
9:  $G = 0$ 
10: enquanto  $G < G_{max}$  faça
11:    $G = G + 1$ 
12:   Selecionar os pais para cruzamento
13:   Gerar descendentes
14:   Aplicar mutações
15:   Avaliar a cobertura da população por meio da função  $f$ 
16:   Classificar a nova população em ordem decrescente de acordo com a cobertura
17:   se gerar um indivíduo com cobertura de 100% então
18:     escapar do laço
19:   fim do se
20: fim do enquanto
21: Retornar o melhor indivíduo gerado

```

5.4.1 Representação dos indivíduos e população inicial

Cada indivíduo da população representa uma possível solução para o problema, ou seja, uma configuração específica das posições das ERBs dentro da área de interesse. A codificação dos indivíduos foi realizada por meio de um vetor de coordenadas geográficas, em que cada gene corresponde à localização de uma ERB, conforme apresentado na Figura 11. A representação gráfica do mesmo indivíduo é ilustrada na Figura 12, em que os círculos laranja indicam tanto as posições das ERBs (os genes do indivíduo) quanto suas respectivas áreas de cobertura.

A população inicial é gerada aleatoriamente dentro da área delimitada pelos pontos de demanda extremos. Essa região, ilustrada na Figura 13 por um retângulo vermelho, também

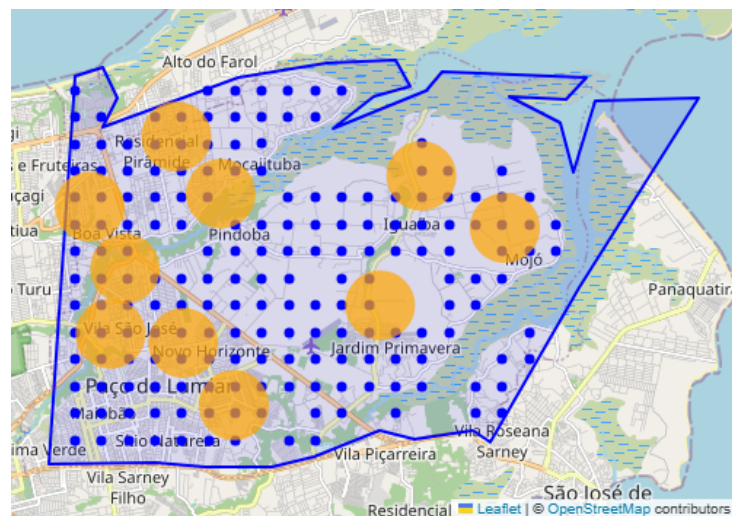
define os limites do espaço de busca considerado pelo algoritmo, abrangendo uma área aproximada de 114,66 km².

Figura 11 – Vetor de coordenadas geográficas de um indivíduo

Gene (ERB)	Latitude	Longitude
0	[[-2,53758773	-44,14168715],
1	[-2,49091947	-44,17566718],
2	[-2,52076119	-44,17076554],
3	[-2,47439127	-44,15525073],
4	[-2,50545967	-44,16734441],
5	[-2,49588469	-44,07783850],
6	[-2,52294658	-44,15363736],
7	[-2,48783505	-44,14477762],
8	[-2,51407599	-44,10724306],
9	[-2,48364413	-44,09751802]]

Fonte: elaborada pelo autor (2025).

Figura 12 – Representação gráfica de um indivíduo



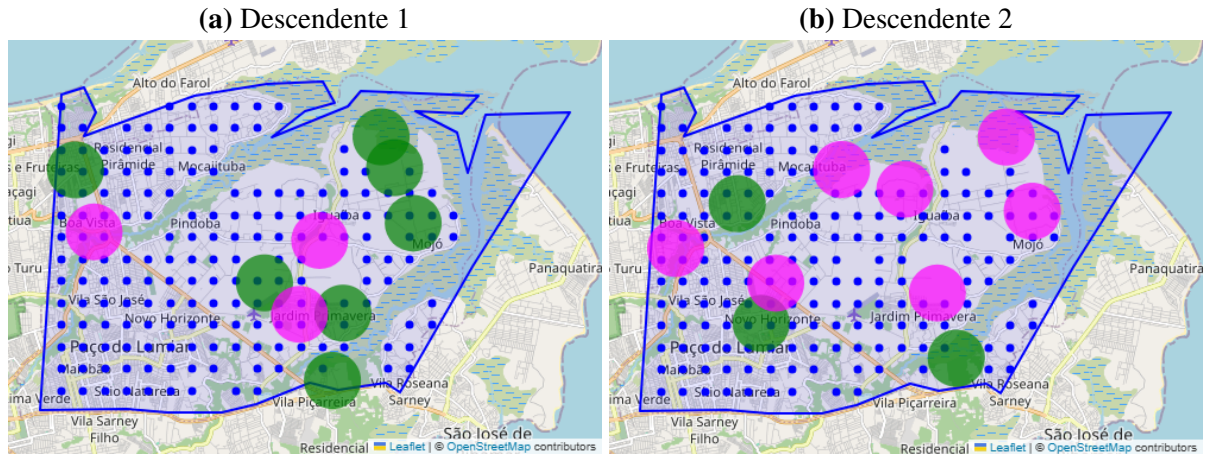
Fonte: elaborada pelo autor (2025).

5.4.2 Seleção dos pais e cruzamento

Após a geração da população inicial, os indivíduos são avaliados e classificados de acordo com o percentual de cobertura que proporcionam. Em seguida, aplica-se um processo análogo à seleção natural: metade dos indivíduos com pior desempenho é eliminada, enquanto a outra metade, composta pelos mais aptos, avança para a próxima geração (Elitismo). Esses sobreviventes tornam-se responsáveis por produzir os descendentes que irão recompor o tamanho original da população. A escolha dos pais para a reprodução é feita aleatoriamente entre os indivíduos sobreviventes, garantindo diversidade genética nas próximas gerações. Ilustra-se na Figura 14 um exemplo de indivíduos selecionados como pais.

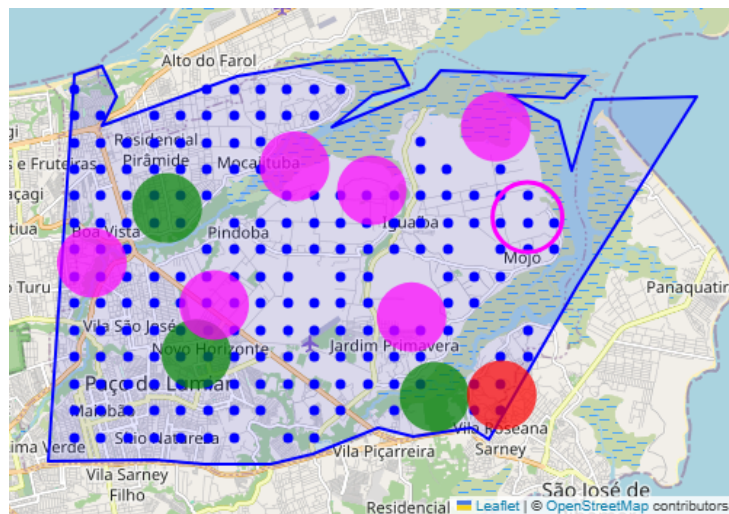
da população, preservando-se o melhor indivíduo da geração anterior a fim de manter sua configuração ótima. Na Figura 16, ilustra-se esse processo: o descendente 2, ilustrado na Figura 15(b), sofreu mutação, na qual uma nova ERB (círculo vermelho) foi inserida no lugar do gene original (circunferência rosa) herdado da mãe.

Figura 15 – Exemplo de descendentes gerados



Fonte: elaborada pelo autor (2025).

Figura 16 – Exemplo de indivíduo que sofreu mutação



Fonte: elaborada pelo autor (2025).

5.5 Implementação do PSO

No Algoritmo 2, apresenta-se o pseudocódigo do PSO implementado para resolver o problema de alocação de ERBs 5G, descrevendo o fluxo do código desenvolvido. As linhas 1 a 4 definem os mesmos dados utilizados no AG apresentado no Algoritmo 1. Em seguida, na linha 5, são estabelecidos os parâmetros: tamanho do enxame igual a 16 partículas, fatores cognitivo e social iguais a 2 e número máximo de iterações fixado em 300. As principais operações a partir da linha 6 serão detalhadas ao longo desta seção.

Algoritmo 2 – Pseudocódigo do PSO para alocação de ERBs 5G

```

1: Carregar os pontos de demanda ( $M$ )
2: Definir o raio de cobertura ( $R$ )
3: Definir a quantidade de ERBs por partícula ( $N$ )
4: Definir a função objetivo  $f$  (Equação 3)
5: Definir parâmetros (tamanho do enxame, fator cognitivo, fator social e iterações ( $I_{max}$ ))
6: Gerar o enxame
7: Gerar as velocidades das partículas
8: Avaliar a cobertura do enxame por meio da função  $f$ 
9: Salvar a melhor posição local e global
10:  $I = 0$ 
11: enquanto  $I < I_{max}$  faça
12:    $I = I + 1$ 
13:   Calcular a inércia
14:   Gerar os números aleatórios  $r_1$  e  $r_2$  dentro do intervalo  $[0, 1]$ 
15:   Calcular a velocidade das partículas por meio da Equação 1
16:   Atualizar a posição das partículas por meio da Equação 2
17:   Avaliar a cobertura das partículas por meio da função  $f$ 
18:   Atualizar a melhor posição local e global
19:   se uma partícula encontrar cobertura de 100% então
20:     escapar do laço
21:   fim do se
22: fim do enquanto
23: Retornar a melhor posição global
  
```

De forma análoga aos indivíduos do AG, no PSO cada partícula representa uma possível solução para o problema. A codificação adotada é um vetor de coordenadas geográficas, em que cada posição do vetor corresponde à localização de uma ERB (conforme a Figura 11). O enxame inicial é gerado aleatoriamente dentro da área delimitada pelo retângulo vermelho ilustrado na Figura 13. As velocidades iniciais das partículas são atribuídas aleatoriamente, variando entre $-0,005$ e $0,005$. Após essa etapa, procede-se à avaliação da cobertura obtida pelo enxame gerado.

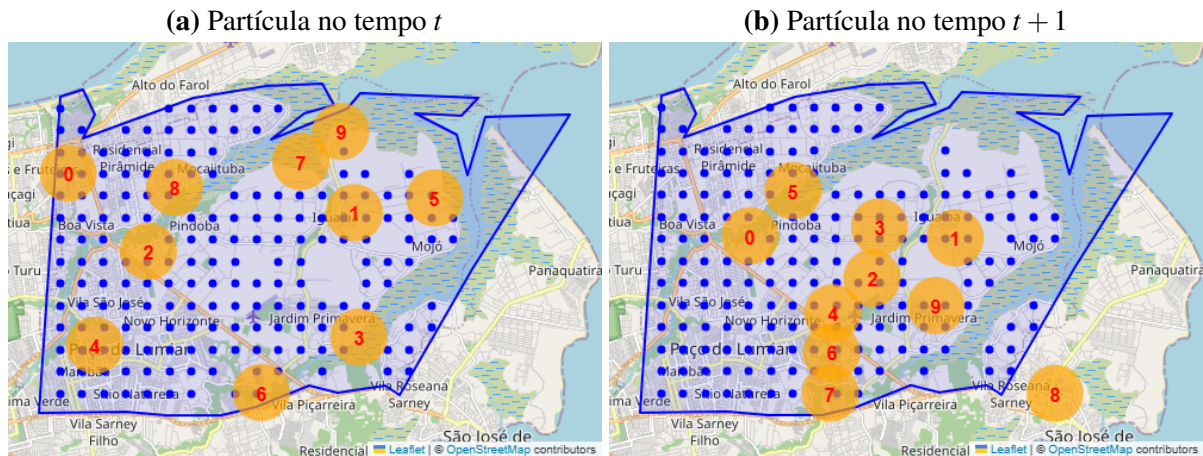
5.5.1 Atualização das posições das partículas

Antes da atualização das posições, é necessário calcular a inércia e gerar os números aleatórios que serão utilizados no cálculo das velocidades das partículas. A inércia atua como um peso que equilibra o efeito de exploração e convergência do algoritmo, reduzindo gradativamente o deslocamento das partículas ao longo das iterações. Os números aleatórios r_1 e r_2 são gerados no intervalo $[0, 1]$. De posse dessas constantes, as velocidades são calculadas conforme a Equação 1, sendo posteriormente utilizadas para atualizar as posições das partículas.

A atualização das posições, definida pela Equação 2, consiste no acréscimo das velocidades às coordenadas das ERBs que compõem cada partícula. Com isso, as ERBs se deslocam em diferentes direções e magnitudes de forma estocástica. No entanto, esse

movimento é restringido à região de busca previamente apresentada na Figura 13. Esse processo é ilustrado na Figura 17, em que os números em vermelho identificam as ERBs que compõem a partícula.

Figura 17 – Exemplo de atualização da posição de uma partícula



Fonte: elaborada pelo autor (2025).

5.6 Ferramentas utilizadas

Para a realização das simulações, foi utilizado um Notebook Dell Inspiron 15-3567, com as seguintes especificações de *hardware* e *software*:

- Processador: Intel Core i3-6006U de 2,00 GHz;
- Memória RAM: 8 GB DDR4;
- Armazenamento: SSD de 240 GB;
- Sistema Operacional: Windows 10 Home.

Os códigos foram escritos em Python utilizando as bibliotecas:

- Geopandas: utilizada para a manipulação e análise de informações geoespaciais;
- Shapely: possibilitou a realização de operações geométricas envolvendo objetos espaciais;
- Geopy: utilizada para cálculo de distâncias geodésicas entre coordenadas geográficas;
- Pandas: empregada no tratamento e análise de dados tabulares;
- NumPy: ofereceu suporte às operações matemáticas e manipulação de *arrays*;
- Matplotlib: permitiu a criação de gráficos;

- Folium: responsável pela construção de mapas interativos com sobreposição de informações geográficas.

Durante o desenvolvimento dos algoritmos, foram utilizadas duas plataformas: Visual Studio Code, com a extensão Jupyter, e o Google Colab. O Visual Studio Code foi empregado na execução local dos experimentos. Já o Google Colab foi utilizado nas fases iniciais de desenvolvimento dos algoritmos, por disponibilizar um ambiente em nuvem, facilitando os testes e o compartilhamento do código.

6 RESULTADOS

De forma simplificada, ao dividir a área da região de busca (conforme a Figura 13), que corresponde a $114,66 \text{ km}^2$, pela área de cobertura de uma única ERB (conforme a Figura 10), igual a $2,54 \text{ km}^2$, obtém-se uma estimativa mínima de aproximadamente 46 ERBs necessárias para cobrir toda a região. Contudo, como os pontos de demanda não estão distribuídos de forma homogênea no espaço considerado e, na prática, o posicionamento das ERBs também não segue uma disposição uniforme, a quantidade real de estações necessárias para garantir a cobertura pode divergir da estimativa inicial, sendo maior ou menor conforme a distribuição da demanda. Considerando esse aspecto, nas simulações foram adotados cenários com 55, 50, 45 e 40 ERBs por indivíduo/partícula, de modo a determinar o número mínimo capaz de garantir a cobertura integral dos pontos de demanda da área analisada. Além disso, para cada configuração de ERBs foram executadas cinco simulações, uma vez que ambos os algoritmos possuem natureza estocástica, podendo gerar soluções distintas a cada execução.

6.1 Simulações com o AG

As Tabelas 1, 2, 3 e 4 apresentam os resultados de percentual de cobertura e o tempo de processamento, em minutos, obtidos nas cinco simulações realizadas com o AG para cada configuração de ERBs. Observa-se que, em uma mesma configuração, os valores de cobertura tendem a ser bastante próximos, chegando, em alguns casos, a coincidir. Já os tempos de processamento, embora também apresentem proximidade, apresentam variações em todas as simulações.

Tabela 1 – Simulações com 55 ERBs por indivíduo

Simulação	Cobertura máxima (%)	Tempo (min)
1	98,37	145,32
2	98,37	143,18
3	97,28	136,55
4	99,46	138,55
5	97,83	142,80

Fonte: elaborada pelo autor (2025).

A evolução da cobertura ao longo das gerações é apresentada nos gráficos da Figura 18. Observa-se que as curvas de cobertura apresentam um crescimento acentuado nas primeiras 100 gerações, seguido por uma desaceleração. Esse comportamento está associado à maior complexidade do ajuste fino na alocação das ERBs após os ganhos iniciais de cobertura. Apesar dessa desaceleração, a curva de melhor cobertura não se encontra estagnada ao final das 300 gerações, o que indica que, com um número maior de gerações, o algoritmo possivelmente

alcançaria percentuais de cobertura ainda mais elevados. Entretanto, tal incremento implicaria também em maior tempo de processamento. Nota-se, ainda, que as curvas de melhor cobertura (linha azul) e de cobertura média (linha laranja) iniciam-se distantes, mas rapidamente a média se aproxima da melhor cobertura. Esse comportamento ocorre porque os descendentes são gerados a partir da recombinação dos indivíduos da geração anterior, reduzindo a variabilidade da população e, conseqüentemente, elevando a média de cobertura.

Tabela 2 – Simulações com 50 ERBs por indivíduo

Simulação	Cobertura máxima (%)	Tempo (min)
1	96,74	135,28
2	94,02	143,50
3	95,65	147,48
4	95,65	147,20
5	94,57	136,95

Fonte: elaborada pelo autor (2025).

Tabela 3 – Simulações com 45 ERBs por indivíduo

Simulação	Cobertura máxima (%)	Tempo (min)
1	91,30	128,68
2	94,57	128,53
3	91,85	133,45
4	93,48	129,13
5	92,39	130,72

Fonte: elaborada pelo autor (2025).

Tabela 4 – Simulações com 40 ERBs por indivíduo

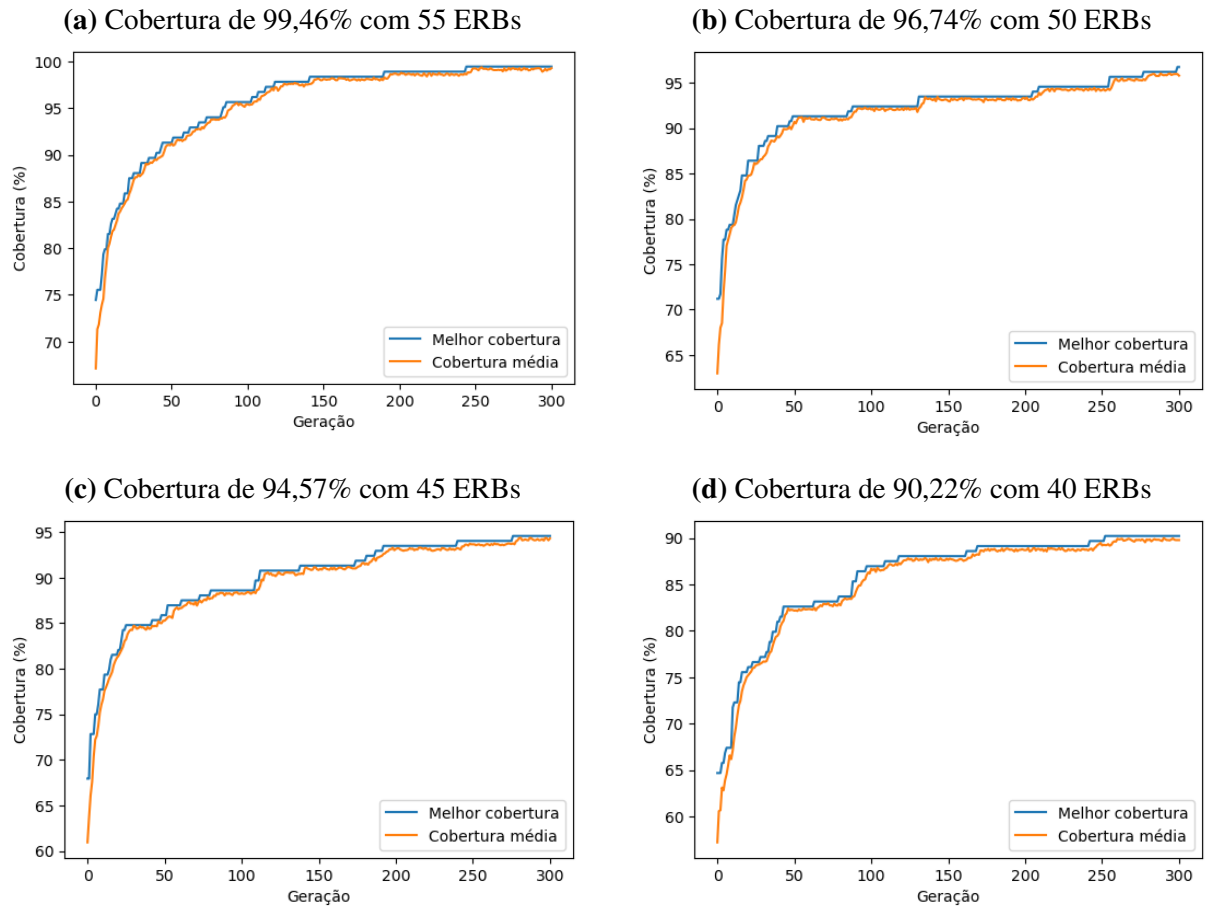
Simulação	Cobertura máxima (%)	Tempo (min)
1	90,22	120,62
2	86,96	123,50
3	89,67	122,92
4	86,41	125,23
5	88,59	123,27

Fonte: elaborada pelo autor (2025).

O mapa com o melhor resultado de cobertura obtido nas simulações com o algoritmo genético é apresentado na Figura 19. Nessa simulação, o AG alcançou 99,46% de cobertura com 55 ERBs, atendendo 183 dos 184 pontos de demanda. Observa-se que poucas estações foram alocadas em regiões sem pontos de demanda, o que demonstra a boa capacidade do algoritmo em direcionar os recursos para áreas relevantes. Contudo, a presença de sobreposição entre algumas ERBs evidencia redundâncias na alocação, indicando que 55 estações constituem um número

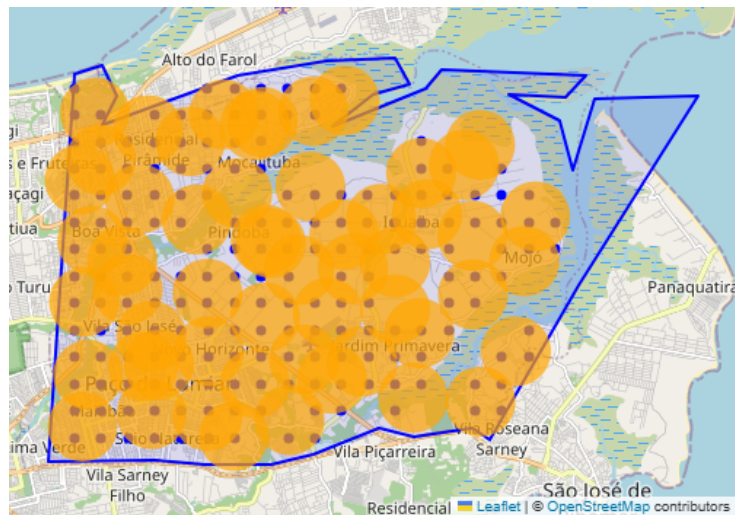
elevado para a região em estudo. Esse resultado sugere que, embora o algoritmo consiga atingir quase a totalidade da cobertura, há margem para otimização quanto ao número de ERBs, de modo a equilibrar eficiência de cobertura e custo de implantação.

Figura 18 – Evolução da cobertura em cada configuração de ERBs no AG



Fonte: elaborada pelo autor (2025).

Figura 19 – Mapa com cobertura de 99,46% obtido com 55 ERBs pelo AG



Fonte: elaborada pelo autor (2025).

6.2 Simulações com o PSO

Os resultados de percentual de cobertura e tempo de processamento obtidos nas cinco simulações realizadas com o PSO para cada configuração de ERBs são apresentados nas Tabelas 5, 6, 7 e 8. Assim como observado no AG, verifica-se que, em uma mesma configuração, os valores de cobertura tendem a ser bastante próximos, chegando, em alguns casos, a coincidir. Os tempos de processamento, por sua vez, embora próximos, apresentam variações em todas as simulações. Ademais, nota-se que, em comparação com o AG, o PSO resultou em menores percentuais de cobertura e maiores tempos de processamento.

Tabela 5 – Simulações com 55 ERBs por partícula

Simulação	Cobertura máxima (%)	Tempo (min)
1	84,78	185,36
2	90,22	186,27
3	88,04	187,85
4	84,78	204,08
5	94,02	181,90

Fonte: elaborada pelo autor (2025).

Tabela 6 – Simulações com 50 ERBs por partícula

Simulação	Cobertura máxima (%)	Tempo (min)
1	86,96	160,02
2	89,13	174,32
3	88,04	180,92
4	91,85	177,01
5	84,78	168,67

Fonte: elaborada pelo autor (2025).

Tabela 7 – Simulações com 45 ERBs por partícula

Simulação	Cobertura máxima (%)	Tempo (min)
1	81,52	169,35
2	82,61	166,17
3	86,96	172,95
4	77,32	169,37
5	83,70	165,43

Fonte: elaborada pelo autor (2025).

Na Figura 20 são apresentados os gráficos de evolução da cobertura referentes às simulações com os melhores resultados obtidos pelo PSO. Observa-se que a curva de melhor cobertura (linha azul) apresenta certa instabilidade, decorrente do fato de a partícula com

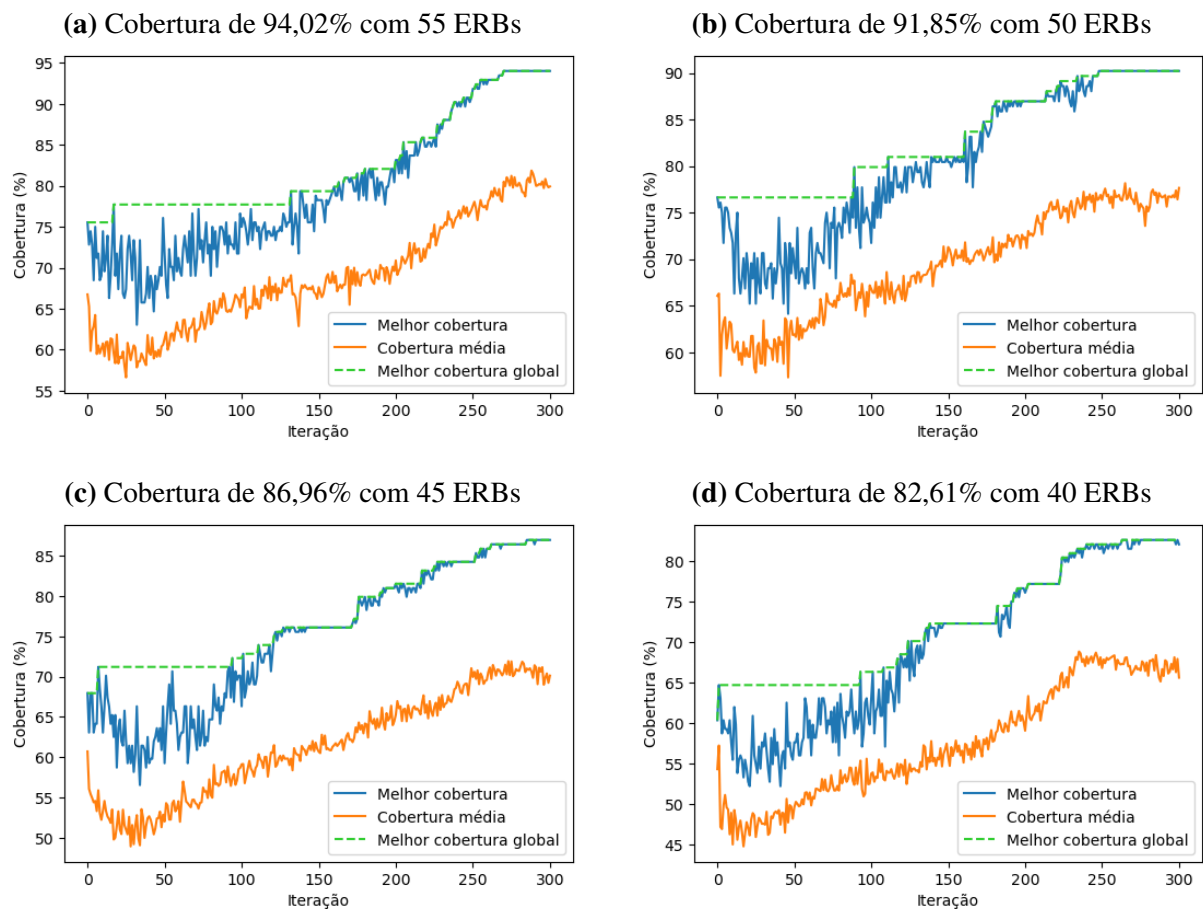
melhor posição se mover a cada iteração do algoritmo. Essa instabilidade, entretanto, tende a ser gradualmente reduzida ao longo das gerações em função do efeito da inércia, que limita progressivamente o deslocamento das partículas. Nota-se ainda que as curvas de melhor cobertura e de cobertura média (linha laranja) mantêm uma distância relativamente constante durante as iterações, o que reflete a variabilidade existente entre as posições das partículas do enxame.

Tabela 8 – Simulações com 40 ERBs por partícula

Simulação	Cobertura máxima (%)	Tempo (min)
1	75,54	160,72
2	79,35	148,42
3	82,61	159,07
4	80,98	152,98
5	79,89	152,88

Fonte: elaborada pelo autor (2025).

Figura 20 – Evolução da cobertura em cada configuração de ERBs no PSO



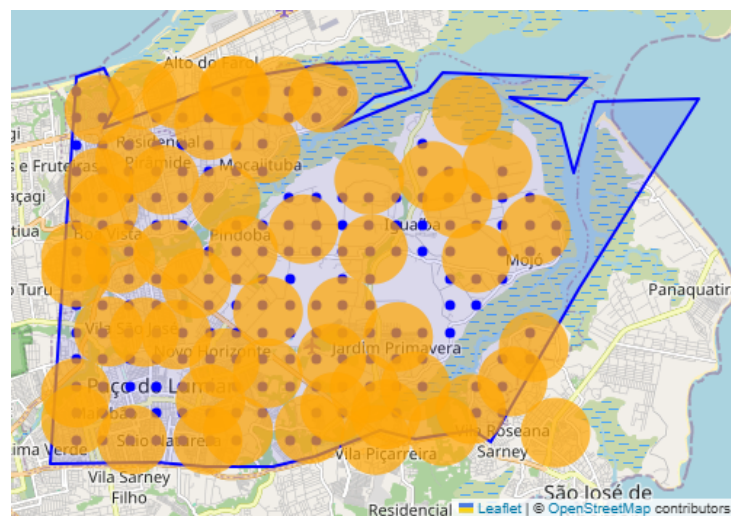
Fonte: elaborada pelo autor (2025).

A curva de melhor cobertura global (linha verde tracejada) representa a cobertura da melhor posição já encontrada por qualquer partícula, sendo análoga à curva de melhor cobertura

do AG, que corresponde ao melhor indivíduo gerado. Inicialmente, essa curva cresce lentamente durante aproximadamente as 100 primeiras iterações, mas a partir desse ponto passa a evoluir de forma mais acelerada, até desacelerar novamente por volta da iteração 280. Esse comportamento inicial mais lento decorre da ausência de valores de referência consistentes nas componentes social e cognitiva das partículas, ou seja, da carência de boas soluções locais e globais que orientem a atualização de suas posições. Já a desaceleração final, assim como observado no AG, está associada à maior complexidade do ajuste fino na alocação das ERBs.

O mapa de cobertura obtido pelo algoritmo PSO está apresentado na Figura 21. Na simulação com 55 ERBs, o PSO atingiu uma cobertura de 94,02% dos pontos de demanda, valor cerca de 5% inferior ao obtido pelo AG. Além da menor cobertura, é possível observar no mapa a presença de ERBs posicionadas fora da região de concentração dos pontos de demanda e um maior número de sobreposições entre ERBs em comparação ao resultado gerado pelo AG. Esses aspectos indicam que, embora o PSO tenha alcançado uma solução próxima do ótimo, seu processo de otimização apresentou menor eficiência na alocação espacial das ERBs.

Figura 21 – Mapa com cobertura de 94,02% obtido com 55 ERBs pelo PSO



Fonte: elaborada pelo autor (2025).

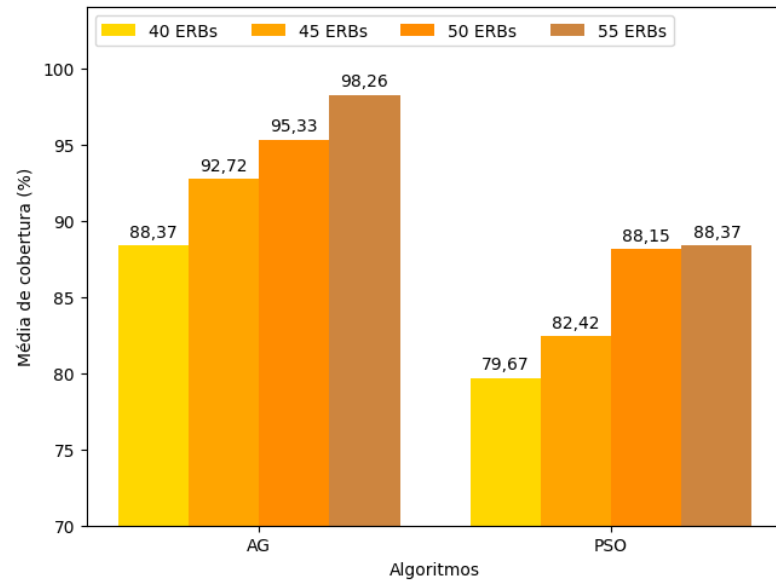
6.3 Comparação dos resultados obtidos pelo AG e pelo PSO

Ao comparar os percentuais médios de cobertura e os tempos médios de processamento das simulações, é possível avaliar de forma mais precisa qual algoritmo apresentou melhor desempenho no problema de alocação das ERBs.

Na Figura 22, apresenta-se um gráfico de barras com os percentuais médios de cobertura para as quatro configurações de ERBs, permitindo a comparação direta entre os algoritmos. Observa-se que o AG superou consistentemente o PSO em todos os cenários analisados. No caso do AG, a redução de 5 ERBs resultou em uma queda de aproximadamente

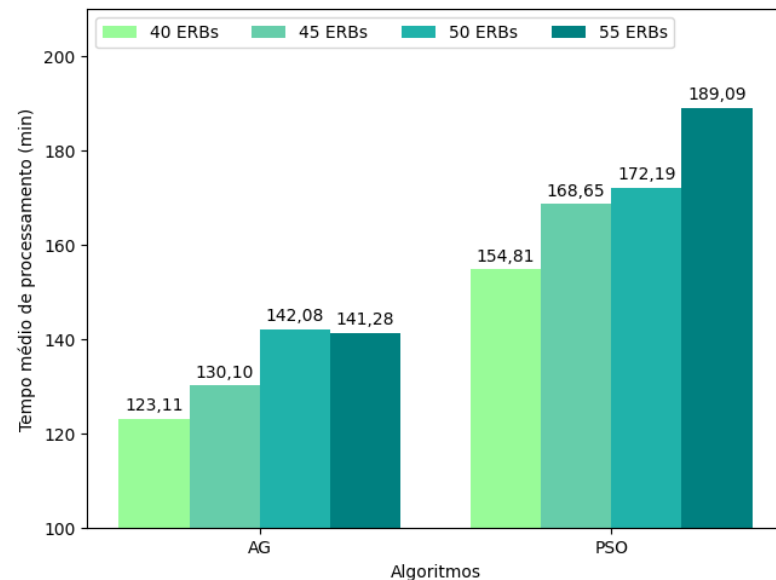
3% na cobertura, indicando um impacto relativamente controlado na qualidade da solução. Já no PSO, os percentuais de cobertura mantiveram-se próximos entre os cenários com 55 e 50 ERBs, sugerindo uma menor sensibilidade à variação no número de estações.

Figura 22 – Média de cobertura dos algoritmos



Fonte: elaborada pelo autor (2025).

Figura 23 – Tempo médio de processamento dos algoritmos



Fonte: elaborada pelo autor (2025).

O tempo médio de processamento dos algoritmos é comparado no gráfico da Figura 23. Verifica-se que o PSO apresentou, em todos os cenários analisados, tempos médios superiores aos do AG. Observa-se ainda que, com a redução de 5 ERBs, o tempo de processamento diminuiu em ambos os algoritmos; entretanto, no AG, as configurações com 55 e 50 ERBs resultaram em

tempos muito próximos, sugerindo que a redução do número de estações nem sempre implica em ganhos expressivos de eficiência computacional.

O desempenho inferior do PSO em termos de tempo de execução pode ser explicado pelas equações de velocidade (Equação 1) e de atualização da posição das partículas (Equação 2). Esses cálculos impõem um custo computacional maior quando comparados aos operadores do AG, que, apesar de envolverem mais etapas (seleção, cruzamento e mutação), apresentam operações mais simples.

De modo geral, nota-se que os tempos de processamento de ambos os algoritmos são elevados. Esse comportamento está diretamente relacionado à modelagem do problema de alocação de ERBs, cuja complexidade cresce com o aumento do número de pontos de demanda, da quantidade de ERBs consideradas e do número de iterações executadas. Isso ocorre porque, a cada avaliação da função objetivo, é necessário verificar a cobertura de cada ponto de demanda em relação a todas as ERBs, elevando significativamente o custo computacional.

7 CONCLUSÃO

Este trabalho teve como propósito aplicar os algoritmos bioinspirados AG e PSO na resolução do problema de alocação de ERBs da rede móvel 5G, na frequência de 3,5 GHz, tendo como área de estudo o município de Paço do Lumiar – MA.

Os resultados demonstraram que o AG apresentou desempenho superior em relação ao PSO, alcançando cobertura de até 99,46% com 55 ERBs, contra 94,02% obtidos pelo PSO na mesma configuração. Além disso, o AG mostrou-se mais eficiente em termos de tempo de processamento, ainda que ambos os algoritmos tenham demandado elevado esforço computacional devido à complexidade do problema. Também foi possível verificar que a redução do número de ERBs impacta diretamente a cobertura, embora de forma mais acentuada no AG do que no PSO. Esses resultados confirmam a viabilidade do uso de algoritmos bioinspirados no planejamento de redes 5G, reforçando a importância de técnicas de otimização avançadas para apoiar decisões estratégicas no setor de telecomunicações.

Apesar dos resultados obtidos, algumas limitações foram identificadas. O modelo adotou simplificações, como a utilização de um raio de cobertura fixo para as ERBs e a discretização do território considerando pontos de demanda com distância elevada entre si. Essas escolhas, necessárias para viabilizar as simulações, podem não refletir com precisão as condições reais de propagação do sinal em ambientes urbanos complexos. Além disso, a elevada carga computacional dos algoritmos restringiu o número de simulações e de gerações/iterações, o que pode ter limitado a obtenção de soluções ainda mais próximas do ótimo global.

Como trabalhos futuros, recomenda-se o aprimoramento do modelo adotado, incorporando fatores como relevo, densidade de edificações e variabilidade do ambiente urbano. Outra oportunidade consiste em aplicar os algoritmos a áreas geográficas maiores e mais heterogêneas, bem como explorar outras meta-heurísticas, ampliando a comparação de desempenho. Além disso, a integração de objetivos múltiplos, como a minimização de custos e a redução de interferências, pode enriquecer as análises, tornando-as ainda mais aplicáveis a cenários reais de planejamento de redes.

Em síntese, o trabalho cumpriu os objetivos inicialmente propostos, oferecendo contribuições relevantes tanto do ponto de vista acadêmico quanto prático. A análise comparativa entre AG e PSO aplicada ao contexto de Paço do Lumiar – MA demonstrou o potencial dessas técnicas para apoiar o processo de implantação do 5G no Brasil, indicando caminhos promissores para futuras pesquisas e para o aprimoramento do planejamento de redes móveis de nova geração.

REFERÊNCIAS

- ALAM, M. N. **Particle Swarm Optimization: Algorithm and its Codes in MATLAB**. ResearchGate, 2016. Disponível em: <http://dx.doi.org/10.13140/RG.2.1.4985.3206>. Acesso em: 2 set. 2025. Citado na página 28.
- ANATEL. **Estações Rádio Base**. 2020. Disponível em: <https://www.gov.br/anatel/pt-br/regulado/outorga/telefonica-movel/estacoes-radio-base>. Acesso em: 5 ago. 2025. Citado na página 19.
- ANATEL. **Editais da Licitação Nº 1/2021-SOR/SPR/CD-ANATEL**. 2021. Disponível em: https://sei.anatel.gov.br/sei/modulos/pesquisa/md_pesq_documento_consulta_externa.php?eEP-wqk1skrd8hSlk5Z3rN4EVg9uLJqrLYJw_9INcO6OyRD8iqL9S9Xn8pP7_7nZL7eBIQWeOiWAZmPzrP88GsKp4NawWkp9n0q_6bMePRxnAsc5z5FRrlZQdkKAChw. Acesso em: 17 jul. 2025. Citado na página 15.
- ANATEL. **Tecnologia 5G**. 2022. Disponível em: <https://www.gov.br/anatel/pt-br/assuntos/5G/tecnologia-5g>. Acesso em: 14 jul. 2025. Citado 2 vezes nas páginas 13 e 15.
- ANATEL. **Painel de acompanhamento e controle do 5G**. 2025. Disponível em: <https://informacoes.anatel.gov.br/paineis/acompanhamento-e-controle/5g>. Acesso em: 18 jul. 2025. Citado na página 13.
- BISWAS, R.; LEMPIAINEN, J. Assessment of 5G as an ambient signal for outdoor backscattering communications. **Wireless Networks**, Springer, v. 27, p. 4083–4094, 2021. Disponível em: <https://doi.org/10.1007/s11276-021-02731-x>. Acesso em: 26 jul. 2025. Citado na página 32.
- COSTA, V. O. **Alocação de antenas para rede celular de 4G utilizando algoritmos meméticos**. 82 p. Dissertação (Mestrado em Modelagem Computacional e Sistemas) — Universidade Federal do Tocantins, Palmas, 2016. Programa de Pós-graduação em Modelagem Computacional e Sistemas. Disponível em: <https://repositorio.uft.edu.br/handle/11612/973?mode=full>. Acesso em: 8 mai. 2025. Citado 6 vezes nas páginas 13, 14, 15, 16, 30 e 31.
- ESA, R. N.; HIKMATUROKHMAN, A.; DANISYA, A. R. 5G NR planning at frequency 3.5 GHz: study case in Indonesia industrial area. **2nd International Conference on Industrial Electrical and Electronics (ICIEE)**, p. 187–193, 2020. Disponível em: <https://www.researchgate.net/publication/348293841>. Acesso em: 15 jul. 2025. Citado 2 vezes nas páginas 13 e 14.
- GASPAR-CUNHA, A.; TAKAHASHI, R.; ANTUNES, C. H. **Manual de computação evolutiva e metaheurística**. [S.l.]: Imprensa da Universidade de Coimbra e Editora UFMG, 2012. Citado 7 vezes nas páginas 20, 21, 22, 23, 24, 25 e 26.
- GU, F.; LIU, H.; LI, M. Evolutionary Algorithm for the Radio Planning and Coverage Optimization of 3G Cellular Networks. **2009 International Conference on Computational Intelligence and Security**, IEEE, v. 2, p. 109–113, 2009. Disponível em: <https://doi.org/10.1109/CIS.2009.240>. Acesso em: 17 jul. 2025. Citado na página 16.
- HALVARSSON, B. *et al.* 5G NR Testbed 3.5 GHz Coverage Results. **2018 IEEE 87th Vehicular Technology Conference (VTC Spring)**, IEEE, 2018. Disponível em:

<https://doi.org/10.1109/VTCSpring.2018.8417704>. Acesso em: 26 jul. 2025. Citado na página 32.

HAUPT, R. L.; HAUPT, S. E. **Practical genetic algorithms**. 2. ed. [S.l.]: John Wiley & Sons, 2004. Citado 5 vezes nas páginas 20, 23, 26, 27 e 52.

IBGE. **Cidades e Estados do Brasil**. 2025. Disponível em: <https://cidades.ibge.gov.br/>. Acesso em: 25 jul. 2025. Citado na página 29.

ISABONA, J. *et al.* Accurate Base Station Placement in 4G LTE Networks Using Multiobjective Genetic Algorithm Optimization. **Wireless Communications and Mobile Computing**, v. 2023, p. 1–9, 2023. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2023/7476736>. Acesso em: 15 jul. 2025. Citado 2 vezes nas páginas 13 e 15.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. **Proceedings of ICNN'95 - International Conference on Neural Networks**, IEEE, Perth, WA, Australia, v. 4, p. 1942–1948, 1995. Disponível em: <https://doi.org/10.1109/ICNN.1995.488968>. Acesso em: 31 ago. 2025. Citado na página 26.

KENNEDY, J.; EBERHART, R. C.; SHI, Y. **Swarm Intelligence**. San Francisco: Morgan Kaufmann, 2001. Citado na página 27.

LINDEN, R. **Algoritmos Genéticos: Uma Importante Ferramenta da Inteligência Computacional**. 2. ed. Rio de Janeiro: Brasport, 2008. Citado 4 vezes nas páginas 21, 22, 25 e 26.

MEDEIROS, J. C. de O. **Princípios de telecomunicações: teoria e prática**. 5. ed. São Paulo: Érica, 2016. Citado 2 vezes nas páginas 17 e 18.

PEI, A. *et al.* Research on location selection model of base station based on improved genetic algorithm. **Highlights in Science, Engineering and Technology**, v. 24, p. 162–170, 2022. Disponível em: <https://doi.org/10.54097/hset.v24i.3908>. Acesso em: 16 jul. 2025. Citado na página 16.

PEREIRA, M. B.; CAVALCANTI, F. R. P.; MACIEL, T. F. Particle swarm optimization for base station placement. **2014 International Telecommunications Symposium (ITS)**, p. 1–5, 2014. Disponível em: <https://doi.org/10.1109/ITS.2014.6948033>. Acesso em: 15 jul. 2025. Citado na página 14.

SAPKOTA, B. *et al.* 5G network deployment planning using metaheuristic approaches. **Telecom**, v. 5, n. 3, p. 588–608, 2024. Disponível em: <https://doi.org/10.3390/telecom5030030>. Acesso em: 16 jul. 2025. Citado 2 vezes nas páginas 15 e 16.

SHEN, J. *et al.* Evolutionary particle swarm optimization algorithm based on collective prediction for deployment of base stations. **Computers, Materials & Continua**, v. 82, n. 1, p. 345–369, 2025. Disponível em: <https://doi.org/10.32604/cmc.2024.060335>. Acesso em: 14 jul. 2025. Citado 2 vezes nas páginas 13 e 15.

TAVARES, Y. M.; NEDJAH, N.; MOURELLE, L. de M. Utilização de otimização por enxame de partículas e algoritmos genéticos em rastreamento de padrões. **Anais do 12 Congresso Brasileiro de Inteligência Computacional**, ABRICOM, Curitiba, PR, p. 1–6, 2015. Disponível em: https://sbic.org.br/eventos/cbic_2015/cbic2015_submission_49/. Acesso em: 28 ago. 2025. Citado 5 vezes nas páginas 23, 24, 25, 27 e 28.

TELECO. **Implantação de Estações Rádio Base (ERBs)**. 2009. Disponível em: <https://www.teleco.com.br/tutoriais/tutorialerb/default.asp>. Acesso em: 5 ago. 2025. Citado 3 vezes nas páginas 17, 18 e 19.

YANGYANG, Z. *et al.* Particle swarm optimization for base station placement in mobile communication. **IEEE International Conference on Networking, Sensing and Control, 2004**, IEEE, v. 1, p. 428–432, 2004. Disponível em: <https://doi.org/10.1109/ICNSC.2004.1297476>. Acesso em: 15 jul. 2025. Citado 3 vezes nas páginas 13, 14 e 31.

APÊNDICE A – CÓDIGOS DOS ALGORITMOS UTILIZADOS

Os algoritmos AG e PSO empregados neste trabalho foram adaptados de Haupt e Haupt (2004), com modificações para atender aos objetivos da pesquisa. Além disso, os códigos, desenvolvidos em Python, também estão disponíveis em repositório público no GitHub (<https://github.com/olumor10/tcc>), juntamente com os demais arquivos complementares.

A.1 Código-fonte do AG implementado

```

1  ### AG aplicado ao problema de localização de ERBs 5G ###
2  # Bibliotecas
3  import matplotlib.pyplot as plt
4  import folium
5  import numpy as np
6  from geopy.distance import geodesic
7  import geopandas as gpd
8  from shapely.geometry import Point
9  import pandas as pd
10
11 # 1 DADOS DO PROBLEMA DE ERBs
12 # 1.1 Carrega a geometria do município de Paço do Lumiar - MA
13 geojson_url = "https://raw.githubusercontent.com/tbrugz/geodata-br/
    master/geojson/geojs-21-mun.json"
14 gdf = gpd.read_file(geojson_url)
15 gdf_municipio = gdf[gdf["name"] == "Paço do Lumiar"].reset_index(drop=
    True)
16 geometria_municipio = gdf_municipio.loc[0, "geometry"]
17
18 # 1.2 Carrega os Pontos de demanda
19 pontos_demanda = pd.read_csv(
20     "C:/Users/Romul/Documents/Visual_Studio/TCC/pontos_demanda/
    pontos_demanda.csv")
21 latmin_pd = pontos_demanda["latitude"].min()
22 latmax_pd = pontos_demanda["latitude"].max()
23 lonmin_pd = pontos_demanda["longitude"].min()
24 lonmax_pd = pontos_demanda["longitude"].max()
25 pontos_demanda = pontos_demanda.values.tolist()
26
27 # 1.3 Gera as ERBs aleatoriamente dentro da geometria do município
28 def gerar_ERB(n_pontos):
29     pontos = []
30     while len(pontos) < n_pontos:
31         y = np.random.uniform(latmin_pd, latmax_pd)
32         x = np.random.uniform(lonmin_pd, lonmax_pd)

```

```

33     ponto = Point(x, y)
34     if geometria_municipio.contains(ponto):
35         pontos.append((y, x))
36     return pontos
37
38 # 1.4 Raio estimado de cobertura de uma ERB 5G
39 raio = 900 # metros
40
41 # 1.5 Qtd de ERBs por indivíduo
42 erbs_por_individuo = 40
43
44 # 1.6 Função objetivo
45 def funcao_objetivo(populacao, pontos_demanda, raio_metros):
46     resultados = []
47     m = len(pontos_demanda)
48
49     for individuo in populacao:
50         Mi = np.zeros(m)
51
52         for i, ponto in enumerate(pontos_demanda):
53             for erb in individuo:
54                 distancia = geodesic(ponto, erb).meters
55                 if distancia < raio_metros:
56                     Mi[i] = 1
57                     break
58
59         f = (np.sum(Mi) * 100) / m
60         resultados.append(f)
61
62     return np.array(resultados)
63
64 # 2 INÍCIO DO ALGORITMO GENÉTICO
65 # 2.1 Critérios de parada do loop
66 max_iteracoes = 300 # Número máximo de gerações
67 cobertura_max = 100 # Cobertura máxima (100%)
68
69 # 2.2 Parâmetros do AG
70 tam_pop = 16 # Tamanho da população
71 tx_mut = 0.2 # Taxa de mutação
72 tx_selec = 0.5 # Taxa de seleção
73 sobreviventes = int(np.floor(tx_selec * tam_pop))
74 num_mut = int(np.ceil(tam_pop * tx_mut)) # Número de mutações
75 # Número de cruzamentos
76 num_cruzamentos = int(np.ceil((tam_pop - sobreviventes) / 2))
77
78 # 2.3 População inicial
79 geracao = 0

```

```

80 populacao = np.array([gerar_ERB(erbs_por_individuo) for _ in range(
    tam_pop)])
81 cobertura = funcao_objetivo(populacao, pontos_demanda, raio)
82 indices_ordenados_decre = np.argsort(-cobertura)
83 cobertura = cobertura[indices_ordenados_decre]
84 populacao = populacao[indices_ordenados_decre]
85 melhor_cobertura = [np.max(cobertura)]
86 cobertura_media = [np.mean(cobertura)]
87
88 # 2.4 Iterar por gerações
89 while geracao < max_iteracoes:
90     geracao += 1
91
92     # 2.4.1 Seleção dos pais
93     pesos = np.flip(np.arange(1, sobreviventes + 1)) / \
94         np.sum(np.arange(1, sobreviventes + 1))
95     distribuicao = np.hstack(([0], np.cumsum(pesos)))
96     selecao_1 = np.random.rand(num_cruzamentos)
97     selecao_2 = np.random.rand(num_cruzamentos)
98     mae = []
99     pai = []
100     for s1, s2 in zip(selecao_1, selecao_2):
101         mae.append(np.searchsorted(distribuicao, s1) - 1)
102         pai.append(np.searchsorted(distribuicao, s2) - 1)
103
104     # 2.4.2 Cruzamento
105     fator_mistura = np.random.rand(erbs_por_individuo)
106
107     for i in range(num_cruzamentos):
108         m = mae[i]
109         p = pai[i]
110
111         descendente1 = np.copy(populacao[m])
112         mae_ = np.copy(populacao[m])
113         descendente2 = np.copy(populacao[p])
114         pai_ = np.copy(populacao[p])
115
116         for j in range(erbs_por_individuo):
117             if fator_mistura[j] < 0.5:
118                 descendente1[j] = pai_[j]
119                 descendente2[j] = mae_[j]
120
121         # Adiciona os descendentes à população
122         populacao[sobreviventes + 2 * i] = descendente1
123         populacao[sobreviventes + 2 * i + 1] = descendente2
124
125     # 2.4.3 Mutar a população

```

```

126     mut_linhas = np.random.randint(1, tam_pop, size=num_mut)
127     mut_colunas = np.random.randint(0, erbs_por_individuo, size=num_mut)
128
129     for linha, coluna in zip(mut_linhas, mut_colunas):
130         populacao[linha, coluna] = gerar_ERB(1)[0]
131
132     # 2.4.3 Checar a convergência
133     cobertura = funcao_objetivo(populacao, pontos_demanda, raio)
134
135     # 2.4.4 Ordena a cobertura e os individuos associados
136     indices_ordenados_decre = np.argsort(-cobertura)
137     cobertura = cobertura[indices_ordenados_decre]
138     populacao = populacao[indices_ordenados_decre]
139
140     # 2.4.5 Estatísticas para uma única geração
141     melhor_cobertura.append(np.max(cobertura))
142     cobertura_media.append(np.mean(cobertura))
143
144     # 2.4.6 Parada por cobertura máxima
145     if cobertura.max() >= cobertura_max:
146         break
147
148 # 2.5 Resultados
149 # 2.5.1 Valor numérico
150 print(f"Gerações: {geracao}")
151 print(f"Melhor cobertura: {cobertura[0]:.2f} %")
152
153 # 2.5.2 Visualização da evolução da cobertura
154 plt.figure(figsize=(6, 4))
155 plt.plot(range(max_iteracoes + 1), melhor_cobertura, label="Melhor
156         cobertura")
157 plt.plot(range(max_iteracoes + 1), cobertura_media, label="Cobertura mé
158         dia")
159 plt.title("Evolução da cobertura")
160 plt.xlabel("Geração")
161 plt.ylabel("Cobertura (%)")
162 plt.legend(loc="lower right")
163 plt.show()
164
165 # 2.5.3 Mapa da melhor cobertura gerada
166 lon_min, lat_min, lon_max, lat_max = geometria_municipio.bounds
167 center_lat = (lat_min + lat_max) / 2
168 center_lon = (lon_min + lon_max) / 2
169 mapa = folium.Map(location=[center_lat, center_lon],
170                   zoom_start=12, width=500, height=350, zoomControl=
171                   False)

```



```

170 folium.GeoJson(
171     geometria_municipio,
172     style_function=lambda x: {
173         "fillColor": "blue",
174         "color": "blue",
175         "weight": 2,
176         "fillOpacity": 0.1,
177     }
178 ).add_to(mapa)
179
180 for ponto in pontos_demanda:
181     folium.CircleMarker(
182         location=[ponto[0], ponto[1]],
183         radius=2,
184         color="blue",
185         fill=True,
186         fill_color="blue",
187         fill_opacity=1
188     ).add_to(mapa)
189
190 for ponto in populacao[0]:
191     folium.Circle(
192         location=[ponto[0], ponto[1]],
193         radius=raio,
194         color=None,
195         fill=True,
196         fill_color="orange",
197         fill_opacity=0.7,
198         popup=f"Coordenada: [{ponto[0]}, {ponto[1]}]"
199     ).add_to(mapa)
200
201 # Salva o mapa em HTML
202 mapa.save("C:/Users/Romul/Documents/Visual_Studio/TCC/AG/mapa_cobertura.
    html")

```

A.2 Código-fonte do PSO implementado

```

1 ### PSO aplicado ao problema de localização de ERBs 5G ###
2 # Bibliotecas
3 import matplotlib.pyplot as plt
4 import folium
5 import numpy as np
6 from geopy.distance import geodesic
7 import geopandas as gpd
8 from shapely.geometry import Point

```

```

9 import pandas as pd
10
11 # 1 DADOS DO PROBLEMA DE ERBs
12 # 1.1 Carrega a geometria do município de Paço do Lumiar - MA
13 geojson_url = "https://raw.githubusercontent.com/tbrugz/geodata-br/
    master/geojson/geojs-21-mun.json"
14 gdf = gpd.read_file(geojson_url)
15 gdf_municipio = gdf[gdf["name"] == "Paço do Lumiar"].reset_index(drop=
    True)
16 geometria_municipio = gdf_municipio.loc[0, "geometry"]
17
18 # 1.2 Carrega os Pontos de demanda
19 pontos_demanda = pd.read_csv("C:/Users/Romul/Documents/Visual_Studio/TCC
    /pontos_demanda/pontos_demanda.csv")
20 latmin_pd = pontos_demanda["latitude"].min()
21 latmax_pd = pontos_demanda["latitude"].max()
22 lonmin_pd = pontos_demanda["longitude"].min()
23 lonmax_pd = pontos_demanda["longitude"].max()
24 pontos_demanda = pontos_demanda.values.tolist()
25
26 # 1.3 Gera as ERBs aleatoriamente dentro da geometria do município
27 def gerar_ERB(n_pontos):
28     pontos = []
29     while len(pontos) < n_pontos:
30         y = np.random.uniform(latmin_pd, latmax_pd)
31         x = np.random.uniform(lonmin_pd, lonmax_pd)
32         ponto = Point(x, y)
33         if geometria_municipio.contains(ponto):
34             pontos.append((y, x))
35     return pontos
36
37 # 1.4 Raio estimado de cobertura de uma ERB 5G
38 raio = 900 # metros
39
40 # 1.5 Qtd de ERBs por indivíduo
41 erbs_por_individuo = 40
42
43 # 1.6 Limita o espaço de busca
44 def limitar_busca(array):
45     latitudes = array[..., 0]
46     longitudes = array[..., 1]
47     lat_limitada = np.clip(latitudes, latmin_pd, latmax_pd)
48     lon_limitada = np.clip(longitudes, lonmin_pd, lonmax_pd)
49     array_limitado = np.stack((lat_limitada, lon_limitada), axis=-1)
50     return array_limitado
51
52 # 1.7 Função objetivo

```

```

53 def funcao_objetivo(populacao, pontos_demanda, raio_metros):
54     resultados = []
55     m = len(pontos_demanda)
56
57     for individuo in populacao:
58         Mi = np.zeros(m)
59
60         for i, ponto in enumerate(pontos_demanda):
61             for erb in individuo:
62                 distancia = geodesic(ponto, erb).meters
63                 if distancia < raio_metros:
64                     Mi[i] = 1
65                     break
66
67         f = (np.sum(Mi) * 100) / m
68         resultados.append(f)
69
70     return np.array(resultados)
71
72 # 2 INÍCIO DO ALGORITMO PSO
73 # 2.1 Critérios de parada do loop
74 max_iteracoes = 300 # Número máximo de iterações
75 cobertura_max = 100 # Cobertura máxima (100%)
76
77 # 2.1 Parâmetros do PSO
78 tam_pop = 16 # Tamanho da população
79 cognitivo = 2 # Fator cognitivo
80 social = 4 - cognitivo # Fator social
81
82 # 2.3 População (Enxame)
83 populacao = np.array([gerar_ERB(erbs_por_individuo) for _ in range(
84     tam_pop)])
85 velocidades = np.random.uniform(low=-0.005, high=0.005, size=populacao.
86     shape)
87 cobertura = np.array(funcao_objetivo(populacao, pontos_demanda, raio))
88 melhor_cobertura_iteracao = [cobertura.max()] # Melhor cobertura na
89     iteração atual
90 cobertura_media_iteracao = [cobertura.mean()] # Cobertura média do
91     enxame em cada iteração
92 melhor_cobertura_global = [melhor_cobertura_iteracao[0]] # Melhor
93     cobertura global até agora
94 melhor_posicao_global = populacao[cobertura.argmax()] # Melhor posição
95     global até agora
96 melhores_posicoes_locais = populacao.copy() # Melhor posição local de
97     cada partícula
98 melhores_coberturas_locais = cobertura.copy() # Melhor cobertura local
99     de cada partícula

```

```

92 posicoes = populacao.copy()
93
94 # 2.4 Início das iterações
95 iteracao = 0
96 while iteracao < max_iteracoes:
97     iteracao += 1
98
99     # 2.4.1 Inércia das partículas
100     inercia = (max_iteracoes - iteracao) / max_iteracoes
101
102     # 2.4.2 Números aleatórios para atualização das velocidades
103     r1 = np.random.uniform(0, 1, size=populacao.shape)
104     r2 = np.random.uniform(0, 1, size=populacao.shape)
105
106     # 2.4.3 Atualiza as velocidades
107     velocidades = (
108         inercia * velocidades
109         + cognitivo * r1 * (melhores_posicoes_locais - posicoes)
110         + social * r2 * (melhor_posicao_global[np.newaxis, :, :] -
111             posicoes)
112     )
113
114     # 2.4.4 Atualiza as posições das partículas
115     posicoes = posicoes + velocidades
116
117     # 2.4.5 Limitar as ERBs à geometria do município
118     posicoes = limitar_busca(posicoes)
119
120     # 2.4.6 Avalia a cobertura das novas posições
121     cobertura = funcao_objetivo(posicoes, pontos_demanda, raio)
122
123     # 2.4.7 Atualiza a melhor cobertura e posição local
124     melhorou = cobertura > melhores_coberturas_locais
125     melhores_coberturas_locais[melhorou] = cobertura[melhorou]
126     melhores_posicoes_locais[melhorou] = posicoes[melhorou]
127
128     # 2.4.8 Atualiza a melhor cobertura e posição global
129     if cobertura.max() > melhor_cobertura_global[-1]:
130         melhor_cobertura_global.append(cobertura.max())
131         melhor_posicao_global = posicoes[cobertura.argmax()]
132     else:
133         melhor_cobertura_global.append(melhor_cobertura_global[-1])
134
135     # 2.4.9 Salva os dados para plotagem
136     melhor_cobertura_iteracao.append(cobertura.max())
137     cobertura_media_iteracao.append(cobertura.mean())

```

```

138 # 2.4.10 Parada por cobertura máxima
139 if cobertura.max() >= cobertura_max:
140     break
141
142 # 2.5 Resultados
143 # 2.5.1 Valor numérico
144 print(f"Iterações: {iteracao}")
145 print(f"Melhor cobertura global: {max(melhor_cobertura_global):.2f} %")
146
147 # 2.5.2 Visualização da evolução da cobertura
148 plt.figure(figsize=(6, 4))
149 plt.plot(range(max_iteracoes + 1), melhor_cobertura_iteracao, label="
    Melhor cobertura")
150 plt.plot(range(max_iteracoes + 1), cobertura_media_iteracao, label="
    Cobertura média")
151 plt.plot(range(max_iteracoes + 1), melhor_cobertura_global, label="
    Melhor cobertura global", linestyle="--", color="LimeGreen")
152 plt.title("Evolução da cobertura")
153 plt.xlabel("Iteração")
154 plt.ylabel("Cobertura (%)")
155 plt.legend(loc="lower right")
156 plt.show()
157
158 # 2.5.3 Mapa da melhor cobertura gerada
159 lon_min, lat_min, lon_max, lat_max = geometria_municipio.bounds
160 center_lat = (lat_min + lat_max) / 2
161 center_lon = (lon_min + lon_max) / 2
162 mapa = folium.Map(location=[center_lat, center_lon], zoom_start=12,
    width=500, height=350, zoomControl=False)
163
164 folium.GeoJson(
165     geometria_municipio,
166     style_function=lambda x:{
167         "fillColor": "blue",
168         "color": "blue",
169         "weight": 2,
170         "fillOpacity": 0.1,
171     }
172 ).add_to(mapa)
173
174 for ponto in pontos_demanda:
175     folium.CircleMarker(
176         location=[ponto[0], ponto[1]],
177         radius=2,
178         color="blue",
179         fill=True,
180         fill_color="blue",

```

```

181         fill_opacity=1,
182         popup=f"Coordenadas: {ponto[0]}, {ponto[1]}"
183     ).add_to(mapa)
184
185 for ponto in melhor_posicao_global:
186     folium.Circle(
187         location=[ponto[0], ponto[1]],
188         radius=raio,
189         color=None,
190         fill=True,
191         fill_color="orange",
192         fill_opacity=0.7
193     ).add_to(mapa)
194
195 # Salva o mapa em HTML
196 mapa.save("C:/Users/Romul/Documents/Visual_Studio/TCC/PS0/mapa_cobertura
    .html")

```

A.3 Código-fonte da discretização de mapas

Este código gera uma malha de pontos sobre toda a área territorial do local selecionado, conforme ilustrado na Figura 9, sendo necessário remover manualmente os pontos localizados em regiões não desejadas.

```

1  ### MALHA DE PONTOS ###
2  # Bibliotecas
3  import geopandas as gpd
4  import numpy as np
5  from shapely.geometry import Point
6  import pandas as pd
7  import folium
8
9  # Carrega o GeoJSON com os municípios do Maranhão
10 geojson_url = "https://raw.githubusercontent.com/tbrugz/geodata-br/
    master/geojson/geojs-21-mun.json"
11 gdf = gpd.read_file(geojson_url)
12
13 # Filtra para o município de Paço do Lumiar
14 gdf_paco_lumiar = gdf[gdf["name"] == "Paço do Lumiar"].reset_index(drop=
    True)
15 geometria_municipio = gdf_paco_lumiar.loc[0, "geometry"]
16 lon_min, lat_min, lon_max, lat_max = geometria_municipio.bounds
17
18 # Projeta para um sistema métrico (UTM zona 23S)
19 gdf_paco_lumiar = gdf_paco_lumiar.to_crs(epsg=31983)

```

```

20 geometria_proj = gdf_paco_lumiar.loc[0, "geometry"]
21 minx, miny, maxx, maxy = geometria_proj.bounds
22
23 # Espaçamento dos pontos da malha
24 espacamento = 700 # (d_pontos) metros
25 x_coords = np.arange(minx, maxx, espacamento)
26 y_coords = np.arange(miny, maxy, espacamento)
27
28 # Cria os pontos da malha
29 pontos = []
30 for x in x_coords:
31     for y in y_coords:
32         ponto = Point(x, y)
33         if geometria_proj.contains(ponto):
34             pontos.append(ponto)
35
36 # Converte os pontos em GeoDataFrame e reprojeta de volta para WGS84
37 gdf_pontos = gpd.GeoDataFrame(geometry=pontos, crs="EPSG:31983")
38 gdf_pontos_wgs84 = gdf_pontos.to_crs(epsg=4326)
39
40 # Extrai latitude e longitude
41 gdf_pontos_wgs84["latitude"] = gdf_pontos_wgs84.geometry.y
42 gdf_pontos_wgs84["longitude"] = gdf_pontos_wgs84.geometry.x
43 malha_pontos = gdf_pontos_wgs84[["latitude", "longitude"]].values.tolist
44     ()
45
46 # Gera o mapa com a malha de pontos
47 center_lat = (lat_min + lat_max) / 2
48 center_lon = (lon_min + lon_max) / 2
49 m = folium.Map(location=[center_lat, center_lon],
50                 zoom_start=13, control_scale=True)
51 folium.GeoJson(
52     geometria_municipio,
53     style_function=lambda x: {
54         "fillColor": "blue",
55         "color": "blue",
56         "weight": 2,
57         "fillOpacity": 0.1,
58     }
59 ).add_to(m)
60 for ponto in malha_pontos:
61     folium.CircleMarker(
62         location=[ponto[0], ponto[1]],
63         radius=2,
64         color="blue",
65         fill=True,
66         fill_color="blue",

```

```
66         fill_opacity=1,
67         popup=f"Coordenadas: {ponto[0]}, {ponto[1]}"
68     ).add_to(m)
69
70 # Salva o mapa em HTML
71 m.save("C:/Users/Romul/Documents/Visual_Studio/TCC/pontos_demanda/
72     mapa_pontos.html")
73
74 # Salva os pontos em arquivo CSV
75 df_malha_pontos = pd.DataFrame(malha_pontos, columns=["latitude", "
76     longitude"])
77 df_malha_pontos.to_csv(
78     "C:/Users/Romul/Documents/Visual_Studio/TCC/pontos_demanda/
79     malha_pontos.csv", index=False)
```