UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GABRIEL ALEXANDRE DE SOUZA BRAGA

MÉTODOS META-HEURÍSTICOS DE LOCALIZAÇÃO PELA POTÊNCIA DO SINAL ÓPTICO RECEBIDO

TOLEDO

2023

GABRIEL ALEXANDRE DE SOUZA BRAGA

MÉTODOS META-HEURÍSTICOS DE LOCALIZAÇÃO PELA POTÊNCIA DO SINAL ÓPTICO RECEBIDO

Metaheuristics methods of locating by the received optical signal strength

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Luis Carlos Mathias

TOLEDO

2023

GABRIEL ALEXANDRE DE SOUZA BRAGA

MÉTODOS META-HEURÍSTICOS DE LOCALIZAÇÃO PELA POTÊNCIA DO SINAL ÓPTICO RECEBIDO

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 21/junho/2023

Prof. Álvaro Ricieri Castro e Souza
Doutorado
Universidade Tecnológica Federal do Paraná

Prof. Daniel Cavalcanti Jeronymo
Doutorado
Universidade Tecnológica Federal do Paraná

Prof. Luis Carlos Mathias
Doutorado
Universidade Tecnológica Federal do Paraná

Prof. Ricardo Tavares de Oliveira
Doutorado
Universidade Tecnológica Federal do Paraná

TOLEDO

2023

Dedico este trabalho a minha família e aos
meus amigos, pelos momentos de ausência.

AGRADECIMENTOS

O presente trabalho não poderia ser finalizado sem a ajuda de diversas pessoas e/ou instituições às quais presto meus agradecimentos. Certamente, esses parágrafos não abrangem todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre estas palavras, mas elas podem estar certas que fazem parte do meu pensamento e tem minha gratidão.

A minha família, pelo carinho, incentivo e total apoio em todos os momentos da minha vida.

Ao meu orientador, que me mostrou os caminhos a serem seguidos e pela confiança depositada.

A todos os professores e colegas do curso, que ajudaram de forma direta e indireta na realização e/ou conclusão deste trabalho.

A todos os demais que de alguma forma contribuíram para meu crescimento pessoal e profissional.

"Primeira Lei: um robô não pode ferir um ser humano ou, por omissão, permitir que um ser humano sofra algum mal.
Segunda Lei: um robô deve obedecer às ordens que lhe sejam dadas por seres humanos, exceto nos casos em que tais ordens contrariem a Primeira Lei.
Terceira Lei: um robô deve proteger sua própria existência desde que tal proteção não entre em conflito com a Primeira ou Segunda Leis."(ASIMOV, 1950, p. 37, tradução)

RESUMO

Este trabalho acadêmico apresenta estruturas de sistemas de posicionamento por luz visível (VLP), que utilizam tanto o algoritmo clássico de Newton-Raphson quanto algoritmos meta-heurísticos, como otimização por enxame de partículas, otimização por colônia de formigas e k-vizinhos mais próximos. O objetivo é comparar os impactos dos algoritmos meta-heurísticos em relação ao método clássico amplamente estabelecido pela comunidade científica. Todos os métodos de posicionamento avaliados aproveitam a informação da potência do sinal recebido (RSS) para evitar a necessidade de sincronização entre os relógios do receptor e dos transmissores. Isso possibilita o desenvolvimento de um sistema de VLP baseado em comunicação por luz visível (VLC), utilizando modulação baseada em multiplexação por divisão de frequências ortogonais (OFDM). Essa abordagem oferece a vantagem de discriminar as potências captadas pelo receptor, fornecendo informações valiosas sobre o RSS. Além disso, neste trabalho foram implementados os três algoritmos meta-heurísticos propostos, com o objetivo de otimizar o sistema de VLP. Para obter as melhores estimativas possíveis, os parâmetros desses algoritmos também foram ajustados. Os principais resultados obtidos foram analisados em termos de precisão, acurácia, taxa de convergência e tempo de execução, comparando os algoritmos meta-heurísticos com o clássico. Esses resultados validam as propostas apresentadas nesta monografia, demonstrando a eficácia dos algoritmos meta-heurísticos no contexto de sistemas de posicionamento por luz visível.

Palavras-chave: potência do sinal recebido; algoritmos meta-heurísticos; comunicação por luz visível; sistema de posicionamento.

ABSTRACT

This academic work presents the structures of visible light positioning (VLP) systems, which utilize the classical Newton-Raphson algorithm and metaheuristic algorithms such as particle swarm optimization, ant colony optimization, and k-nearest neighbors. The aim is to compare the impacts of the metaheuristic algorithms on the widely established classical method within the scientific community. All evaluated positioning methods leverage the received signal strength (RSS) information to eliminate the need for synchronization between the receiver and transmitter clocks. This enables to design a VLP system based on visible light communication (VLC) using orthogonal frequency-division multiplexing modulation (OFDM). This approach offers the advantage of distinguishing the received powers from the receiver, providing valuable information about RSS. Furthermore, this work implements the proposed three metaheuristic algorithms to optimize the VLP system. The parameters of these algorithms are tuned to obtain the most accurate estimates. The main results are analyzed in precision, accuracy, convergence rate, and execution time, comparing the metaheuristic algorithms with the classical method. These results validate the proposals presented in this thesis, demonstrating the effectiveness of the metaheuristic algorithms in the context of visible light positioning systems.

Keywords: received signal strength; metaheuristic algorithm; visible light communications; positioning system.

LISTA DE ALGORITMOS

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Siglas

ACO	Otimização por colônia de formiga – *ant colony optimization*
AOA	Ângulo de chegada – *angle of arrival*
AWGN	Ruído aditivo branco gaussiano – *additive white gaussian noise*
FFT	Transformação rápida de Fourier – *fast Fourier transform*
FOV	Campo de visão – *field of view*
GPS	Sistema de posicionamento global – *global positioning system*
KNN	K-vizinhos mais próximos – *K-nearest neighbor*
LED	Diodo emissor de luz – *light-emitting-diode*
ML	Máxima verossimilhança – *maximum likelihood*
NLLS	Minimização não linear de mínimos quadrados – *nonlinear least square*
OFDM	Multiplexação por divisão de frequências ortogonais – *orthogonal frequency division multiplexing*
PSO	Otimização de enxame de partículas – *particle swarm optimization*
RF	Radiofrequência
RSS	Potência do sinal recebido – *received signal strength*
RSS-MLAT	Multilateração baseada na potência do sinal recebido – *received signal strength based multilateration*
RSS-TL	Trilateração baseada na potência do sinal recebido – *received signal strength based trilateration*
TDOA	Diferença do tempo de chegada – *time difference of arrival*
TIA	Amplificador de transimpedância – *transimpedance amplifier*
TOA	Tempo de chegada – *time of arrival*
VLC	Comunicação por luz visível – *visible light communication*

VLP	Posicionamento por luz visível – *visible light positioning*

LISTA DE SÍMBOLOS

Letras Latinas

$\boldsymbol{r}$	Vetor de posição	[m]
$\boldsymbol{n}$	Versor de orientação	[m]
$\boldsymbol{v}$	Vetor que representa a distância entre o transmissor eo receptor	[m]
n_L	Ordem de distribuição lambertiana	
A_{PD}	Área do fotodiodo	[m^2]
D	Distância euclidiana entre o LED e o fotorreceptor	[m]
P	Potência óptica do LED	[W]
p	Potência óptica do LED amostrada pelo receptor	[W]
$\boldsymbol{s}$	Sinal de potência amostrado pelo fotodiodo	[W]
$\boldsymbol{p}$	Vetor que contém a estimativa da intensidade da potência óptica de cada LED recebida no fotorreceptor	[W]
$\boldsymbol{n}_s$	Vetor que contém as informação de potência do ruído AWGN	[W]
$\boldsymbol{J}$	Matriz jacobiana de p_m em relação a $\boldsymbol{r}_R$	
d_E	Distância dos dados	
d_{room}	Dimensões do quarto analisado	
dc	Constante de decaimento (*decay constant*)	
K	Número de distâncias euclidianas obtidas	
L	Localização da partícula	
v	Velocidade da partícula	
c_1	Peso do comportamento cognitivo da partícula	
c_2	Peso do comportamento social da partícula	
rand	Valor aleatório entre 0 e 1	
pbest	Valor mínimo individual	

gbest	Valor mínimo global	
w	Peso, chamado também como constante de inércia ou ponderamento em alguns algorítimos	
R_{KNN}	Amplitude máxima do raio da vizinhança	$[\text{m}]$
M	Quantidade máxima de transmissores (LEDs) contidas no sistema	
$\boldsymbol{W}$	Matriz diagonal com a inversa do quadrado das potencias estimadas recebidas pelo receptor	$[\text{W}^{-2}]$

Letras Gregas

π	Pi (constante circular)	$[\text{rad}]$
ϕ	Fi (ângulo entre o versor $\boldsymbol{n}_m$ e o vetor $\boldsymbol{v}_m$)	$[\text{rad}]$
θ	Teta (ângulo entre o versor $\boldsymbol{n}_R$ e o vetor $\boldsymbol{v}_m$)	$[\text{rad}]$
Ω	Ômega (perda da potência óptica de percurso do transmissor)	
η	Eta (tamanho do passo, no método recursivo de Newton-Raphson)	
τ	Tau (total de feromônio no caminho)	
ρ	Rô (constante que permite a definição dos níveis de evaporação)	
σ	Sigma (parâmetro que controla a importância relativa ao feromônio na trilha)	
β	Beta (parâmetro que controla a importância relativa ao qualidade na trilha)	
ζ	Zeta (qualidade do caminho)	
ψ	Psi (probabilidade de escolha de caminho feita pela formiga)	
ϵ	Épsilon (critério de parada)	

Sobrescritos

T	Matriz transposta
i	Índice de iterações em métodos recursivos
$\dagger$	Forma generalizada inversa da matriz
t	Índice do caminho
k	Índice do vizinho

Subscritos

m	Índice do LED

R	Receptor
i	Índice da partícula
max	Valor máximo que uma variável pode obter
min	Valor mínimo que uma variável pode obter
x, y	Coordenadas do caminho em um plano

Notações

$\hat{\boldsymbol{r}}$	Estimativa do vetor de posição
$\Delta\tau$	Quantidade de feromônio que uma formiga deixa

SUMÁRIO

1 INTRODUÇÃO

O sistema de posicionamento global (GPS – *global positioning system*) vem sendo usado de maneira predominante em cenários externos (*outdoor*), mas falha em ambientes internos (*indoor*), pois o sinal dos satélites são atenuados pela estrutura predial, acarretando em uma grande margem de erro e falhas em seu funcionamento. Neste cenário, sistemas de posicionamento interno com base em luz visível vem se tornando um tema popular para trabalhos acadêmicos, como visto em Elgala, Mesleh e Haas (2011) e Liu, Makino e Maeda (2008). Isto se dá por conta da alta precisão de estimação no posicionamento, licença de operação gratuita, sem interferência de radiofrequência (RF) e etc.

Dessa forma, sistemas de localização interno por meio de comunicação por luz visível (VLC – *visible light communication*), vem utilizando LEDs (diodo emissor de luz, do inglês, *light-emitting-diode*) como transmissores, enquanto que um receptor óptico é instalado em um dispositivo móvel a fim de detectar o sinal recebido.

Esse é o fundamento dos algoritmos de posicionamento com base no sinal recebido, sendo utilizados em sistemas de posicionamento por luz visível (VLP – *visible light positioning*). Neste contexto, alguns dos algoritmos mais utilizados são:

- O de potência do sinal recebido (RSS – *received signal strength*);
- Pelo ângulo de chegada (AOA – *angle of arrival*);
- Com base no tempo de chegada (TOA – *time of arrival*);
- E pelo cálculo de diferença de tempo de chegada (TDOA – *time difference of arrival*).

Todos esses podem ser vistos em Keskin, Sezer e Gezici (2018), sendo empregados para estimar a distância entre o receptor e o transmissor. No entanto, para sistemas baseados em TOA ou TDOA é essencial uma ótima sincronização entre os relógios do receptor e transmissor. Um funcionamento inadequado do algoritmo de sincronização pode provocar um grande erro na estimativa de localização. Portanto, algoritmos com TOA ou TDOA se demonstram mais complexos, já que dependem de algoritmos de sincronização robustos para seu correto funcionamento (ZHANG; CHOWDHURY; KAVEHRAD, 2014; WANG *et al.*, 2013).

Nestas conformidades, o algoritmo que mais vem sendo usado pela sua simplicidade e eficiência é o RSS, baseando-se na atenuação da potência no canal VLC. Dessa maneira é estabelecida as coordenadas do posicionamento do receptor, sem a necessidade de sincronização entre os relógios (ZHANG; CHOWDHURY; KAVEHRAD, 2014; ZHOU; KAVEHRAD; DENG, 2012; MOHAMMED; ELKARIM, 2015).

Métodos como o Newton–Raphson podem ser utilizados em conjunto com o RSS para estimar a posição do receptor. Se considerada as informações de todas as fontes luminosas, temos um algoritmo de máxima verossimilhança (ML – *maximum likelihood*), como visto em Şahin *et al.* (2015a).

Apesar de sua alta precisão, o algoritmo de Newton–Raphson apresenta problemas de convergência para certos casos. Şahin *et al.* (2015b) tentam mitigar este problema utilizando, como primeira etapa, um algoritmo AOA para estimar o melhor ponto inicial de busca. A segunda etapa consiste em utilizar o método de Newton-Raphson a partir do ponto inicial encontrado na primeira etapa. Dessa forma, é possível obter menores erros no algoritmo de localização e maiores taxas de convergência para um resultado válido.

Outro método clássico bem estabelecido no meio acadêmico é o algoritmo de trilateração que é considerado um algoritmo sub-ótimo (LIN *et al.*, 2017). Pensando em uma maior taxa de exatidão, é possível de forma similar, utilizar mais de três transmissores. Este é conhecido como algoritmo de multilateração (ZHUANG *et al.*, 2018).

Contudo, nos métodos de trilateração baseado em RSS (RSS-TL – *received signal strength based trilateration*) e multilateração baseado em RSS (RSS-MLAT – *received signal strength based multilateration*), a exatidão da localização está diretamente associada com a precisão da estimativa das distâncias euclidianas entre o receptor e os transmissores. Esta pode ser degradada devido aos ruídos, interferência, e/ou problemas de falta de campo de visão (FOV – *field of view*) do fotorreceptor para com o sinal de luz transmitido (HUANG *et al.*, 2020). Diante deste cenário, métodos meta-heurísticos podem melhorar a eficiência destes estimadores.

Visando validar novas soluções de VLP, este trabalho acadêmico propõe desenvolver métodos meta-heurísticos para a estimativa de posicionamento baseado em RSS. A fim de conduzir de forma comparativa, com os algoritmos clássicos já estabelecidos, o desempenho em relação à exatidão, acurácia, taxa de convergência de resultado útil e tempo de execução.

1.1 Objetivo do Trabalho

Este trabalho tem como objetivo geral desenvolver, de modo comparativo, algoritmos clássicos e meta-heurísticos para um sistema de localização por luz visível. Neste contexto, verificando suas eficiências em relação à precisão, acurácia, taxa de convergência de resultado útil e tempo de execução.

Os objetivos específicos são listados abaixo:

- Desenvolver o algoritmo clássico de localização pela potência do sinal recebido, sendo este com base em Newton-Raphson;
- Desenvolver algoritmos meta-heurísticos de localização pela potência do sinal recebido, sendo estes com base em otimização por enxame de partículas (PSO – *particle swarm optimization*), k-vizinhos mais próximos (KNN – *k-nearest neighbor*) e otimização por colônia de formigas (ACO – *ant colony optimization*);
- A partir dos resultados numéricos obtidos, analisar os principais impactos e diferenças entre os métodos aplicados.

1.2 Justificativa

A utilização do GPS para ambientes *indoor* acaba resultando em uma estimativa com uma grande margem de erro. Isso se dá pela forte atenuação causada no sinal dos satélites. Este desvanecimento no sinal é acarretado pelas paredes, tetos, janelas e portas do ambiente. Se predominantemente metálico, o ambiente interno se comporta como uma gaiola de Faraday, trazendo vários obstáculos que suprimem o sinal dos satélites do sistema GPS, levando a um erro considerável na aproximação da localização (LIN *et al.*, 2020).

Neste contexto, o desenvolvimento de um VLP se demonstra mais apropriado, trazendo consigo melhores resultados. Portanto, este trabalho acadêmico visa o desenvolvimento de métodos de VLP utilizando VLC.

Ao desenvolver um esquema VLP é necessário escolher qual método de posicionamento com base no sinal recebido será utilizado. Desse modo, a aplicação do mecanismo de RSS facilita a implementação do algoritmo de localização, dado o fato de que não há a necessidade de elaborar o algoritmo de sincronismo entre os relógios do receptor e dos transmissores (De Gante; SILLER, 2013). Em virtude disso, é utilizado ao decorrer desta monografia o algoritmo de RSS para os métodos de localização.

Dessa forma, é conduzido de forma comparativa a implementação entre algoritmos clássicos e meta-heurísticos, a fim de validar o desempenho dos mesmos. Os algoritmos meta-heurísticos podem apresentar resultados sub-ótimos, porém com menor complexidade computacional e/ou maior taxa de convergência. Métodos como Newton–Raphson, apesar de ser ML e apresentar resultados ótimos, possui problemas de convergência e sensível degradação da acurácia e precisão quando na presença de ruído e interferência, bem como problemas de linha de visada como visto anteriormente. Dessa maneira, é desenvolvido um estudo de caso com algoritmos meta-heurísticos, definindo seus impactos em sistemas de VLP.

1.3 Organização do Texto

Esta monografia está organizada da seguinte maneira:

- O Capítulo 2 demonstra o estado da arte, de maneira a apresentar o esquema de modulação do sinal óptico que pode ser utilizado, desenvolver o modelo do canal, o ruído presente no sistema, o estimador RSS, os algoritmos clássicos e os meta-heurísticos;
- No Capítulo 3 são demonstradas as metodologias adotadas para efetuar os testes e extração dos resultados a partir de simulação, de forma a validar o projeto proposto;
- Assim, o Capítulo 4 traz as análises e interpretações dos resultados obtidos;
- O Capítulo 5 apresenta as principais conclusões e possibilidades de trabalhos futuros.

2 REFERENCIAL TEÓRICO

Para o desenvolvimento correto da proposta feita nesta monografia, é necessário o conhecimento de alguns fundamentos básicos. Em virtude disso, este capítulo visa conduzir de forma explanatória os conceitos do modelo matemático do canal VLC, assim como a estirpe de ruído presente na comunicação do sistema, e estruturar os algoritmos clássicos e metaheurísticos com base na intensidade do sinal recebido, para desta forma efetuar a estimativa de localização.

2.1 Discriminação dos Sinais Capturados no Receptor

O fotodetector do receptor captura um amálgama de todos os sinais dos LEDs transmissores e das demais fontes luminosas do ambiente que possuem linha de visada. Entretanto, os algoritmos de localização por RSS necessitam da informação da intensidade de luz capturada de cada LED transmissor. Assim, é necessário utilizar alguma técnica para discriminar estes sinais e estimar as suas intensidades.

Dessa forma, para efetuar a comunicação entre transmissor e receptor é necessário escolher um método de modulação. Estes se constituem principalmente na manipulação das seguintes características da onda senoidal portadora:

- Da amplitude, como por exemplo a demonstrada em Faruque (2017);
- Também há a modulação por fase, demonstrada em Anderson, Aulin e Sundberg (2013);
- Por fim, há a modulação por frequência (RODER, 1931).

Além da comunicação de dados, essa estratégia de modulação com base em multiplexação por divisão de frequências ortogonais (OFDM – *orthogonal frequency division multiplexing*) permite, como vantagem, a possibilidade de diferenciar as potências captadas pelo receptor (VAPPANGI; MANI, 2019). Esse mecanismo faz com que os sinais transmitidos pelos LEDs sejam codificados em subportadoras ortogonais, permitindo que o receptor receba múltiplos sinais de potência distintos.

Entretanto, uma solução mais expedita, mas que não permite a comunicação de dados, também permite a discriminação de potências (ASSAKAWA; MATHIAS, 2022; CORDEIRO; MATHIAS; ESTEVES, 2021). Nestes trabalhos, no transmissor, a intensidade de luz emitida é modulada de forma senoidal, no qual cada lâmpada possui uma frequência distinta. Seria como se cada lâmpada tivesse sua própria impressão digital. Isto permite no receptor, mediante filtragem com um algoritmo de FFT (*fast Fourier transform* – transformada rápida de Fourier), separar o sinal emitido de cada transmissor.

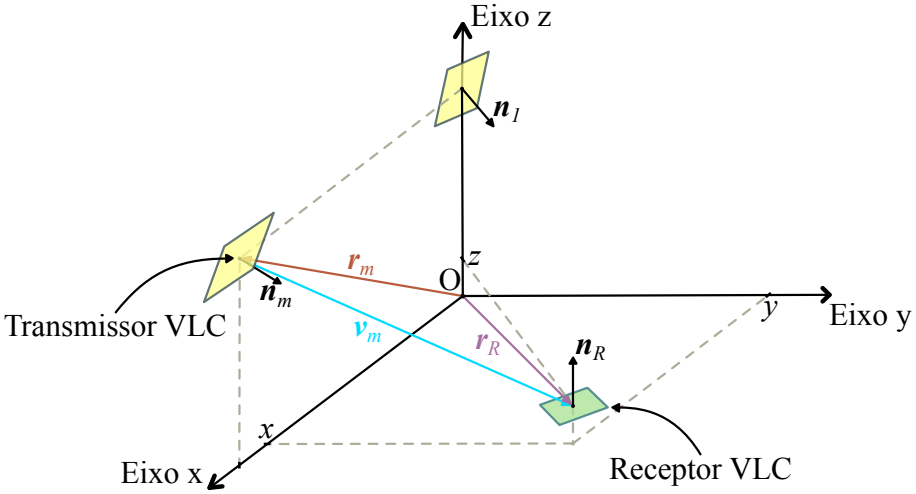
Outra possibilidade é o acionamento sequencial das lâmpadas (SAAB; SAAB, 2016). Entretanto, este dificulta as aplicações de iluminação, além de exigir um algoritmo robusto para exercer a correta sincronização entre os LEDs e receptor.

Neste contexto, como o foco deste trabalho é o desenvolvimento de algoritmos sub-ótimos para a localização, consideramos inicialmente que as potências já estão discriminadas e estimadas no dispositivo receptor móvel.

2.2 Modelo Matemático do Canal VLC

Para modelar o canal VLC pode-se considerar um sistema com M transmissores. Cada transmissão de dados é feita por um LED. Sendo assim, a Figura 1 demonstra a topologia adotada em um espaço cartesiano, de maneira a explanar os vetores de posição e os versores normais de cada transmissor e do receptor.

Figura 1 – Esquemático do modelo do sistema de localização por luz no espaço cartesiano



Fonte: Autoria própria (2023).

$\boldsymbol{r}_m$ e $\boldsymbol{n}_m$ representam respectivamente o vetor de posição e o versor de orientação de um único LED qualquer, sendo este um transmissor de índice m. Para o receptor é adotado $\boldsymbol{r}_R$ e $\boldsymbol{n}_R$, para representarem seu vetor de posição e versor de orientação.

Nestas conformidades, de acordo com Şahin *et al.* (2015b), o vetor que representa a distância entre o transmissor e o receptor, assim como seu ângulo de chegada, pode ser representado por:

$$\boldsymbol{v}_m = \boldsymbol{r}_R - \boldsymbol{r}_m. \tag{1}$$

Assim, ao considerar que a luz emitida da LED está no campo de visão do fotorreceptor, é possível estimar a perda de potência óptica durante o percurso transmissor/receptor. Portanto

se tem (KOMINE; NAKAGAWA, 2004):

$$\mathbf{\Omega}_m = \frac{(n_L + 1) \cdot A_{PD}}{2 \cdot \pi} \cdot \frac{\cos^{n_L}(\phi_m) \cdot \cos(\theta_m)}{D_m^2}; \tag{2}$$

onde A_{PD} é a área do fotodiodo, medida em metros quadrados; ϕ_m é o ângulo entre o versor de orientação do transmissor ($\boldsymbol{n}_m$) e a distância entre transmissor/receptor ($\boldsymbol{v}_m$), também chamado de ângulo de irradiância; θ_m é o ângulo entre o versor de orientação do receptor ($\boldsymbol{n}_R$) e $\boldsymbol{v}_m$, conhecido como ângulo de incidência; D_m se trata da distância euclidiana entre o LED e o fotorreceptor. Por fim, n_L representa a ordem de distribuição Lambertiana. Este é definido pelo semi-ângulo na meia potência do LED ($\phi_{1/2}$):

$$n_L = -\frac{\ln(2)}{\ln\left(\cos\left(\phi_{1/2}\right)\right)}. \tag{3}$$

Ao desenvolver a Equação 2, utilizando o conceito de produto interno de vetores, é possível concluir a seguinte equivalência vetorial (ŞAHIN *et al.*, 2015b):

$$\mathbf{\Omega}_m = -\frac{(n_L + 1) \cdot A_{PD}}{2 \cdot \pi} \cdot \frac{\left(\boldsymbol{v}_m^\intercal \cdot \boldsymbol{n}_m\right)^{n_L} \cdot \boldsymbol{v}_m^\intercal \cdot \boldsymbol{n}_R}{\|\boldsymbol{v}_m\|_2^{n_L+3}}. \tag{4}$$

Em casos ideais, onde a potência óptica (P) de cada LED é constante, tendo a orientação do receptor imutável, e com a posição e orientação dos transmissores permanente, é possível estimar a potência óptica de cada LED capturada pelo fotorreceptor de tal maneira:

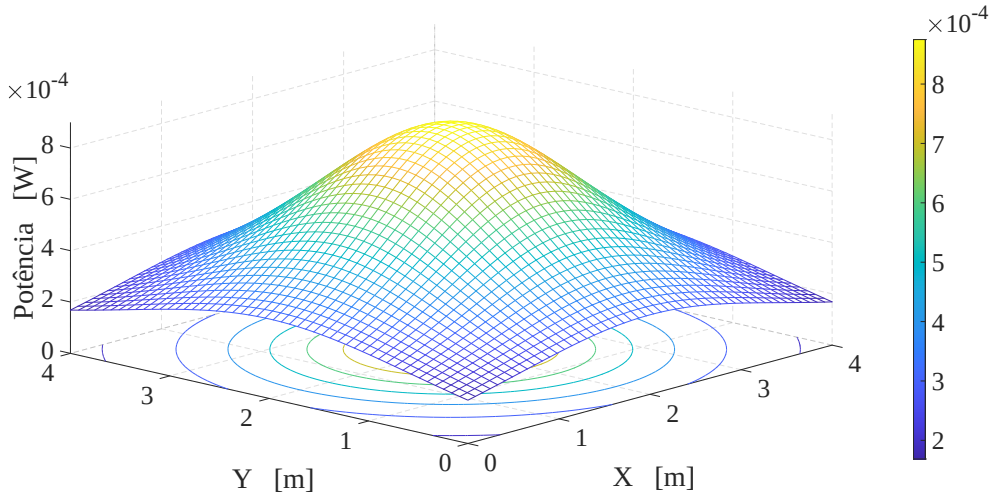
$$p_m(\boldsymbol{r}_R) = \mathbf{\Omega}_m \cdot P. \tag{5}$$

De maneira similar ao desenvolvido por Nguyen *et al.* (2010), é realizada uma simulação a fim de validar o modelo proposto. Com auxílio do *software* MATLAB R2022b é elaborado um programa para calcular a distribuição luminosa de um ou mais LEDs transmissores em uma sala. Considerando um cômodo com as dimensões 4×4×3 metros, com o transmissor VLC posicionado no centro da sala ($\boldsymbol{r}_1 = [2; 2; 3]$ m) com orientação para baixo ($\boldsymbol{n}_1 = [0; 0; -1]$) e potência óptica de 1 Watt. Com o receptor VLC a um metro do chão com orientação para cima ($\boldsymbol{n}_R = [0; 0; 1]$), utilizando ordem de distribuição Lambertiana de $n_L = 1$ e $n_L = 10$ são obtidos os resultados demonstrados na Figura 2 e na Figura 3.

Analisando a Figura 2 e Figura 3, é possível notar que a ordem de distribuição Lambertiana é diretamente proporcional a densidade de potência em torno do foco de orientação do LED transmissor. Ou seja, quanto maior o valor de n_L, mais diretiva é a luz.

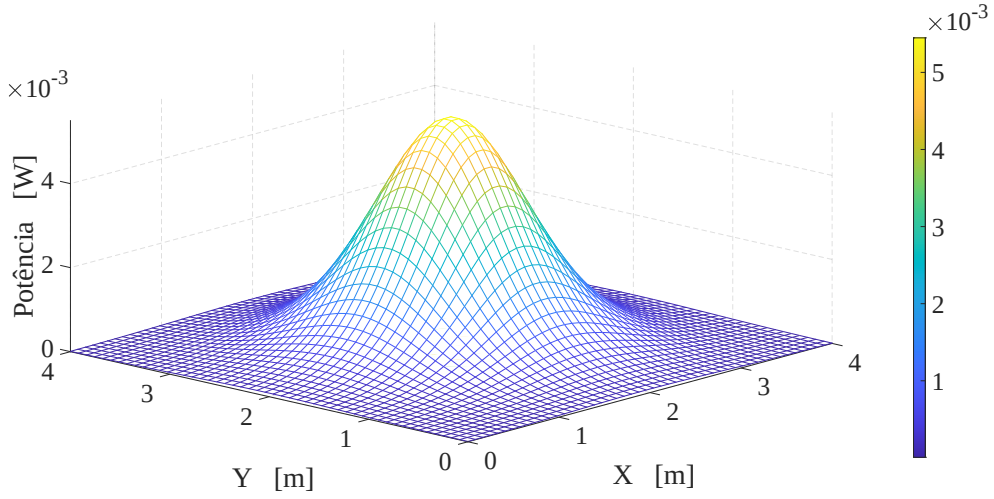
A partir da conclusão obtida com base na comparação entre Figura 2 e a Figura 3, é efetuado um estudo de caso a fim de corrigir o comportamento relacionado a uma alta ordem de distribuição Lambertiana. Neste sentido, é conduzido uma simulação, que é demonstrada na

Figura 2 – Distribuição do fluxo luminoso por unidade de área em um plano, com distância entre receptor e transmissor de 2 m, gerado por um único transmissor com distribuição $n_L = 1$



Fonte: Autoria própria (2023).

Figura 3 – Distribuição do fluxo luminoso por unidade de área em um plano, com distância entre receptor e transmissor de 2 m, gerado por um único transmissor com distribuição $n_L = 10$

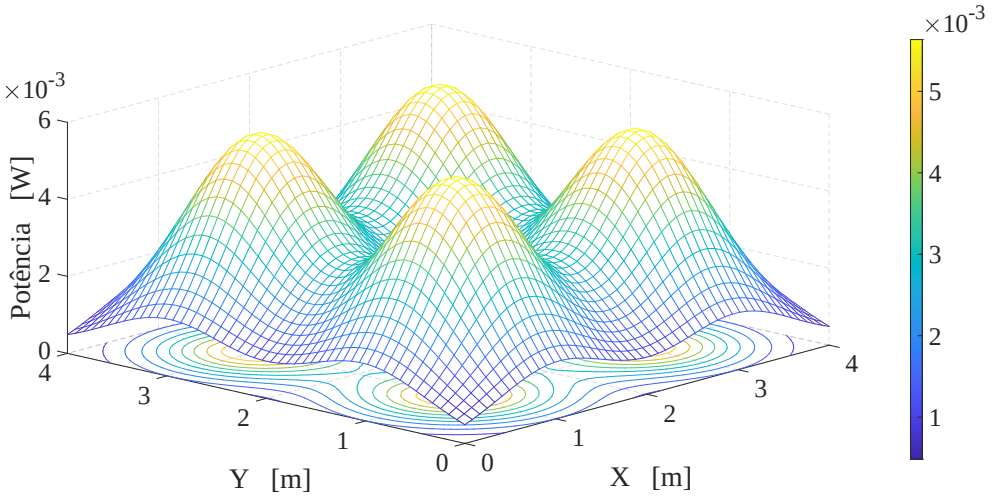


Fonte: Autoria própria (2023).

Figura 4, nesta são utilizadas quatro LEDs, localizadas em $\boldsymbol{r}_1 = [1; 1; 3]$ m, $\boldsymbol{r}_2 = [1; 3; 3]$ m, $\boldsymbol{r}_3 = [3; 1; 3]$ m e $\boldsymbol{r}_4 = [3; 3; 3]$ m.

Dessa forma, é possível validar que uma solução para o comportamento da densidade de potência, relacionado a uma maior magnitude do n_L, é a utilização de mais LEDs transmissores no ambiente. Em virtude disso a distribuição espacial do fluxo luminoso se demonstra mais uniforme.

Figura 4 – Distribuição do fluxo luminoso por unidade de área em um plano, com distância entre receptor e transmissor de 2 m, gerado por quatro transmissores com distribuição $n_L = 10$



Fonte: Autoria própria (2023).

2.2.1 Ruído AWGN

Em um sistema real, existem ruídos no sinal. Na literatura esses são considerados no amplificador de transimpedância (TIA – *transimpedance amplifier*), componente este que compõe o receptor VLC e é responsável por converter o sinal de fotocorrente capturada pelo fotodetector em um sinal de tensão. Os ruídos são compostos por duas partes. Uma parcela se dá pelos ruídos térmicos, causados pelo grau de agitação das moléculas presentes nos componentes do TIA.

A outra parcela é constituída pelos ruídos do tipo *shot* que se dá por meio do movimento randômico dos fótons. Quando um fóton encontra uma região de depleção, sua energia potencial aumenta, até que ele tenha energia o suficiente para transpassá-la. A energia potencial então é convertida, de forma abrupta, para energia cinética. O fenômeno citado é conhecido como o ruído *shot* e ocorre na junção semicondutora do fotodiodo.

Dessa maneira, as parcelas de ruídos *shot* e térmico compõem os ruídos presentes no sistema, mas como elucidado por Şahin *et al.* (2015b) e Mathias e Abrão (2019), os ruídos presentes nos mecanismos de VLC podem ser representados, de maneira equivalente, por um ruído aditivo branco gaussiano (AWGN – *additive white Gaussian noise*). Neste contexto, será utilizada esta aproximação durante o desenvolvimento do projeto proposto.

2.2.2 Potência do Sinal Recebido (RSS)

Para realizar a estimativa do posicionamento do receptor, é vital que se defina qual estimador com base no sinal recebido será utilizado. Como dito no Capítulo 1, o escolhido para o desenvolvimento deste trabalho acadêmico é o por RSS.

O estimador RSS consiste no conhecimento das posições e orientações dos transmissores VLC, como visto em Şahin *et al.* (2015b). Ou seja, é necessário o conhecimento a priori da infraestrutura de VLC do ambiente. Em virtude disso, é possível induzir que o sinal de potência amostrado pelo fotodiodo é:

$$\boldsymbol{s} = \boldsymbol{p}\left(\boldsymbol{r}_R\right) + \boldsymbol{n}_s; \tag{6}$$

sendo $\boldsymbol{s}$ um vetor que contém as potências captadas pelo receptor, $\boldsymbol{p}\left(\boldsymbol{r}_R\right)$ é o vetor que contém a estimativa da intensidade da potência óptica de cada LED recebida no fotorreceptor (este é demonstrada na Equação 5). Por fim, $\boldsymbol{n}_s$ representa um vetor que contém as informações de potência do ruído AWGN, de maneira que $\boldsymbol{s}$, $\boldsymbol{p}\left(\boldsymbol{r}_R\right)$ e $\boldsymbol{n}_s \in \mathbb{R}^{M\times 1}$.

2.3 Algoritmo Clássico de Newton Raphson

Para o desenvolvimento desta monografia, se demonstra necessário entender os conceitos básicos de alguns algoritmos. Visando isso, nessa seção são explicadas as estruturas básicas que serão utilizadas para a elaboração dos algoritmos de RSS clássicos para que na seção subsequente, sejam apresentados os algoritmos meta-heurísticos.

Na literatura, há algoritmos clássicos que com uma alta taxa de frequência são utilizados. Um deles será desenvolvido, de forma que seus resultados sejam aplicados de forma comparativa com os resultados obtidos dos algoritmos meta-heurísticos que serão concebidos.

É importante ressaltar que para ambos os métodos meta-heurísticos ou clássicos, é necessário a utilização do RSS, para estimar a posição do receptor. Nas simulações numéricas arbitra-se uma posição exata, aplica-se o modelo do sistema VLC, e baseado neste modelo aplicam-se os algoritmos que utilizam métodos de aproximação para estimar a posição do receptor.

Assim, será elucidado o fundamento por trás destes algoritmos de estimação clássicos. Sendo assim, é visto mais adiante o funcionamento do método de descida íngreme (Newton-Raphson).

Como descrito por Mathias, Melo e Abrão (2019), o problema de estimativa ML da posição $\boldsymbol{r}_R$ pode ser expresso como um problema de minimização não linear de mínimos quadrados (NLLS – *nonlinear least squares*):

$$\hat{\boldsymbol{r}}_R = \arg\min_{\boldsymbol{r}_R} \left(\| \boldsymbol{s} - \boldsymbol{p}\left(\boldsymbol{r}_R\right) \|_2^2 \right); \tag{7}$$

sendo $\|.\|_2$ o operador norma euclidiana.

Assim, é possível minimizar as distâncias entre o vetor de potência amostrado $\boldsymbol{s}$ e o vetor de potências exatas $\boldsymbol{p}\left(\boldsymbol{r}_R\right)$. Dessa forma, tendo como resultado na Equação 7, o vetor que representa a posição do receptor calculada é $\hat{\boldsymbol{r}}_R$.

Com o intuito de resolver o problema de minimização demonstrado pela Equação 7, isto é, estimar a localização do receptor com fotodiodo, é empregado uma técnica conhecida como Newton-Raphson multivariado (ŞAHIN *et al.*, 2015b; KAY, 1993).

Considerando que $\boldsymbol{r}_R^i \in \mathbb{R}^{3\times 1}$, onde o expoente i representa o índice de iteração, ao aplicar o método de descida íngreme, é possível obter a seguinte equação:

$$\boldsymbol{r}_R^{i+1} = \boldsymbol{r}_R^i - \eta \cdot \boldsymbol{J}^{\dagger} \cdot \left(\boldsymbol{s} - \boldsymbol{p}\left(\boldsymbol{r}_R^i\right)\right); \tag{8}$$

onde η é o tamanho do passo com $\eta \in (0,1]$. $\boldsymbol{J}^{\dagger}$ representa a forma generalizada inversa da matriz Jacobiana ($\boldsymbol{J}$) de $p_m\left(\boldsymbol{r}_R\right)$, demonstrada logo a seguir:

$$\boldsymbol{J}^{\dagger} = \left(\boldsymbol{J}^{\intercal} \cdot \boldsymbol{J}\right)^{-1} \cdot \boldsymbol{J}^{\intercal}. \tag{9}$$

A Equação 10 representa a matriz Jacobiana de $p_m\left(\boldsymbol{r}_R\right)$, de maneira que cada coluna representa um eixo geométrico da posição do receptor, onde $\boldsymbol{r}_R = [x, y, z]$.

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial p_1}{\partial x} & \frac{\partial p_1}{\partial y} & \frac{\partial p_1}{\partial z} \\ \frac{\partial p_2}{\partial x} & \frac{\partial p_2}{\partial y} & \frac{\partial p_2}{\partial z} \\ \vdots & \vdots & \vdots \\ \frac{\partial p_M}{\partial x} & \frac{\partial p_M}{\partial y} & \frac{\partial p_M}{\partial z} \end{bmatrix}. \tag{10}$$

Dessa forma, as linhas da matriz jacobiana representam a taxa de variação do RSS de cada transmissor em relação ao movimento do receptor ao longo de seus eixos geométricos.

2.4 Algoritmos Meta-heurísticos

Algoritmos meta-heurísticos vem tomando grande proporção no cenário atual do mundo, demonstrando cada vez mais sua ampla aplicação e eficiência em problemas de otimização, se tornando um dos campos de estudo mais fomentados na área de sistemas inteligentes.

Neste contexto, aqui é desenvolvido, de forma inicial, os fundamentos por trás dos algoritmos de otimização que serão aplicados neste trabalho acadêmico. Tendo como o intuito serem utilizados para fins comparativos com o algoritmo clássico já pré-estabelecido, a fim de validar o desempenho destes e suas complexidades.

2.4.1 K-Vizinhos Mais Próximos (KNN)

O KNN é uma técnica que considera a distância entre dados para a realização de predições, de maneira a ser usado para classificação e regressão de dados (FIX; HODGES, 1989; COVER; HART, 1967).

Este algoritmo se consolida com a hipótese de que dados semelhantes tendem a estar mais agrupados em uma mesma região no espaço de dispersão dos dados. Como dito anteriormente, o KNN funciona por meio da distância dos dados:

$$d_E^k = \boldsymbol{s} - \boldsymbol{p}\left(\boldsymbol{r}_{R\_KNN}^k\right). \tag{11}$$

Dessa maneira, utilizando a Equação 11, é possível calcular essa distância dos dados (d_E), para o caso do problema VLP com VLC deste trabalho. É importante ressaltar que d_E pode ser qualquer tipo de informação relevante, que possa ser expressa de forma quantitativa, para a otimização do problema. Para este caso seria o erro entre a potência amostrada e a estimada. O sobrescrito k contido em $\boldsymbol{r}_{R\_KNN}$ e d_E significa o índice do vizinho, ou seja, representa a posição em que o vizinho se encontra e o erro da potência do vizinho, respectivamente. Note que a posição do vizinho $(\boldsymbol{r}_{R\_KNN})$, para este caso, corresponde a uma possível localização estimada do receptor.

Em virtude disso, uma possível maneira para estimar a posição do fotodiodo pode ser utilizando a média das coordenadas dos vizinhos. Neste contexto é possível realizar a seguinte operação:

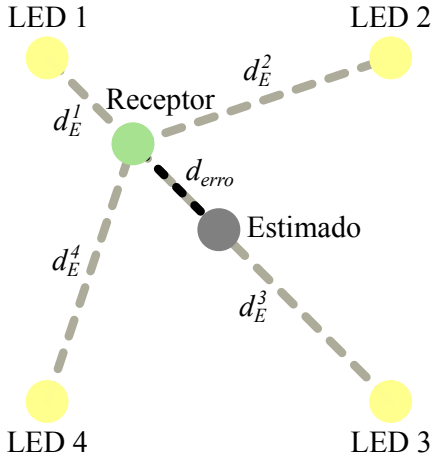
$$\hat{\boldsymbol{r}}_R = \frac{\sum_{k=1}^{K}\left(\frac{\sum_{m=1}^{M}\left(\boldsymbol{r}_m - \boldsymbol{r}_{R\_KNN}^k\right)}{M}\right)}{K}. \tag{12}$$

Utilizando a Equação 12 é possível estimar as coordenadas utilizando o KNN, onde K expressa o número total de vizinhos utilizados para compor a população deste algoritmo, e M representa o número total de LEDs utilizados. Porém, problemas geométricos tendem a não trazer um bom resultado médio.

Por exemplo, com $K = 1$ e $M = 4$, tendo $\boldsymbol{r}_1 = [2; 4; 3]$ m, $\boldsymbol{r}_2 = [4; 4; 3]$ m, $\boldsymbol{r}_3 = [4; 6; 3]$ m, $\boldsymbol{r}_4 = [2; 6; 3]$ m, o vizinho iniciando em $\boldsymbol{r}_{R\_KNN}^1 = [1; 3; 1]$ m, e tomando $\boldsymbol{r}_R = [1; 3; 1]$ m como a posição exata do receptor. É possível notar que a posição estimada para o receptor seria $\hat{\boldsymbol{r}}_R = [2; 2; 2]$ m, o que por sua vez é o centro da sala. Mas como visto o receptor não se encontra neste local. A Figura 5 demonstra o exemplo citado.

Para evitar este cenário, é muito comum a utilização de uma constante de ponderamento. Desta maneira, é necessário analisar o problema e elaborar a melhor equação que define essa ponderação do problema, a fim de encontrar a constante e conseguir dados de melhor qualidade. Isto será melhor desenvolvido na subseção 3.1.2.

Figura 5 – Erro ao estimar receptor utilizando KNN sem constante de ponderamento.



Fonte: Autoria própria (2023).

Além disso, também pode ser utilizada a Equação 11 para ajudar na estimativa, utilizando d_E juntamente com algum critério de seleção para assim utilizar no cálculo da estimativa apenas os vizinhos que passem nesse critério, como por exemplo $d_E < 0{,}1$. Assim só seriam selecionados os vizinhos em que o erro das potências é menor que 0,1.

Sendo assim, de forma sucinta, para efetuar a otimização por KNN basta seguir os passos:

1. Calcular a distância das informações de maior relevância para o seu problema;
2. Ordenar esse conjunto de dados de forma crescente;
3. Elaborar uma solução para definir o melhor vizinho possível com base no problema;
4. Projetar uma solução para definir a melhor constante de ponderação;
5. Calcular uma média ponderada da distância com os k-vizinhos mais próximos.

2.4.2 Otimização por Enxame de Partículas (PSO)

Outra forma de efetuar uma otimização é com a utilização do PSO. Este foi feito com base na análise do comportamento de cardumes de peixes e bandos de pássaros. Desta forma foi desenvolvido um algoritmo que se comporta de maneira similar a de um cardume de peixe, ou bando de pássaros, para a resolução de problemas irrestritos de otimização não-linear (KENNEDY; EBERHART, 1995).

É importante ressaltar que ao imitar o comportamento desses grupos de animais, isso faz com que cada elemento do exame aprenda com o comportamento de seus demais componentes, pois cardumes de peixes e bandos de pássaros possuem uma consciência coletiva. Neste contexto, o algoritmo apresenta comportamento dinâmico em deslocamentos relativa-

mente complexos, onde cada partícula tem acesso a um número limitado de informações do coletivo, como a posição e a velocidade de seus vizinhos mais próximos.

Dessa maneira, a fim de emular o comportamento de um cardume de peixes, Clerc (1999) propõe a seguinte equação para a localização de uma partícula:

$$L_i\left[k\right] = L_i\left[k-1\right] + v_i\left[k-1\right]; \tag{13}$$

sendo que o subscrito i indica o índice da partícula, o parâmetro k descreve sua instância, L é a posição da partícula e v é sua velocidade. Também segundo Clerc (1999), a equação que demonstra a velocidade da partícula pode ser definida por:

$$v_i\left[k\right] = w \cdot v_i\left[k-1\right] + c_1 \cdot \text{rand}_1 \cdot \left(\text{pbest}_i\left[k-1\right] - L_i\left[k-1\right]\right) + c_2 \cdot \text{rand}_2 \cdot \left(\text{gbest}_i\left[k-1\right] - L_i\left[k-1\right]\right); \tag{14}$$

sendo que w é a constante de inércia; c_1 e c_2 representam o peso do comportamento cognitivo e social da partícula, respectivamente; rand_1 e rand_2 são números aleatórios entre 0 e 1. Por fim, o valor mínimo individual é representado por pbest, e gbest é o valor mínimo global.

Com base em análises da Equação 14, é possível concluir que há uma parcela de comparativo individual e uma parcela de comparativo do grupo. Dessa forma, é validado o comportamento simulado de cardume de peixes.

É importante dizer que não se tem garantia de encontrar o mínimo global. Também há o fato de que o algoritmo emprega o parâmetro de v_{max} e v_{min}, para limitar a velocidade do movimento das partículas, de forma a evitar a lentidão ou aceleração exacerbada do sistema. Dessa mesma forma, na posição há um alcance máximo de $-L_{\text{max}}$ à L_{max}.

Com saber no que foi dito, para realizar o algoritmo de PSO basta seguir os passos:

1. Definir o tamanho da população, constante de inércia, velocidade inicial, máxima e mínima, peso do comportamento cognitivo e social, e localização máxima das partículas;
2. Calcular a função *fitness* (função a ser otimizada);
3. Calcular a melhor posição da partícula;
4. Atualizar a velocidade e a posição de cada partícula;
5. Atualizar a função *fitness*;
6. Armazenar a melhor partícula global (mínimo global);
7. Verificar se alcançou número máximo de interações, se sim, termine o programa, caso contrário volte para o passo 4.

2.4.3 Otimização por Colônia de Formigas (ACO)

De maneira muito semelhante ao do PSO, o ACO simula o comportamento de formigas buscando comida. Este algoritmo consiste em deixar traços de feromônio pelos caminhos que as formigas fazem, sendo os caminhos mais eficientes os com maiores traços de feromônio. Além disso também há a evaporação, o que faz com que traços antigos sejam eliminados, reforçando para as formigas o caminho mais eficiente, dado o fato de que a escolha de um caminho é probabilístico. Sendo assim, quanto mais caminhos, menor é a probabilidade da formiga escolher o correto, e quanto maior o feromônio em um caminho, maior a probabilidade da formiga escolhê-lo (DORIGO; MANIEZZO; COLORNI, 1991).

Visando simular este comportamento no algoritmo, Dorigo, Maniezzo e Colorni (1996) propõem o modelo matemático subsequente:

$$\tau_{x,y}^{t}\left[k\right] = (1-\rho)\cdot\tau_{x,y}^{t}\left[k-1\right] + \sum_{m=1}^{M}\left(\Delta\tau_{x,y}^{m}\right); \tag{15}$$

onde τ representa o total de feromônio no caminho, o sobrescrito t representa o índice do caminho, os subscritos x e y representam as coordenadas do caminho em um plano, e o parâmetro k representa a instância do feromônio. ρ é uma constante que permite a definição dos níveis de evaporação com $0 < \rho < 1$. A quantidade de feromônio que uma formiga deixa é representado por $\Delta\tau$ e este pode ser calculado com base na Equação 16:

$$\Delta\tau_{x,y}^{m} = \begin{cases} \frac{1}{d_m}\text{ , caso a formiga passe nesse caminho,} \\ 0\text{ , caso contrário;} \end{cases} \tag{16}$$

onde o sobrescrito m representa o índice de cada formiga e d_m é o comprimento do caminho feito pela formiga.

Como dito anteriormente, a escolha das rotas das formigas se dá de maneira probabilística, podendo ser expressa da seguinte forma:

$$\psi_{x,y}^{t} = \frac{\left(\tau_{x,y}^{t}\right)^{\sigma}\cdot\left(\zeta_{x,y}^{t}\right)^{\beta}}{\sum_{t=1}^{T}\left(\left(\tau_{x,y}^{t}\right)^{\sigma}\cdot\left(\zeta_{x,y}^{t}\right)^{\beta}\right)}; \tag{17}$$

onde σ e β são parâmetros que controlam a importância relativa ao feromônio e a qualidade na trilha, também há o fato de que $\sigma > 0$ e $\beta > 0$. ζ representa a qualidade do caminho, ou seja, quanto menor o caminho, maior sua qualidade, este se dá por:

$$\zeta_{x,y} = \frac{1}{d_{x,y}}; \tag{18}$$

em que com base nisso, é possível utilizar algoritmos como *roulette wheel selection* para simular a seleção de rota das formigas.

Por fim, para garantir que todos os caminhos sejam percorridos pelo menos uma vez, existe a lista de tabu. Esta é uma lista que guarda todos os caminhos percorridos pelas formigas numa instancia k e as proíbe de revisitá-los até que n interações sejam feitas.

Neste contexto, é possível traçar os seguintes passos para o algoritmo:

1. Definir o tamanho da população, constante de evaporação e constantes σ e β;
2. Calcular a probabilidade de escolha de cada caminho (regra de transição);
3. Atualizar lista de tabu;
4. Efetuar busca local;
5. Atualizar feromônio;
6. Verificar se alcançou número máximo de interações, se sim, termine o programa, caso contrário, volte para o passo 2.

3 MATERIAIS E MÉTODOS

Visando desempenhar com os objetivos apresentados e atestar os conceitos retratados, serão desenvolvidas simulações numéricas do tipo Monte Carlo. Isto se dá pelo fato de que há pelo menos uma variável aleatória no sistema, sendo essa o ruído AWGN.

3.1 Desenvolvimento dos Algoritmos

Para o desenvolvimento do projeto será utilizado o *software* MATLAB. Serão desenvolvidos em forma de *scripts* os *softwares* de estimação para o VLP, estes sendo o de Newton-Raphson, como algoritmo clássico; e os meta-heurísticos, sendo o KNN, o PSO e o ACO. Todos os algoritmos, sendo estes os meta-heurísticos ou o clássico, utilizam os conceitos do RSS.

3.1.1 Desenvolvimento do Algoritmo de Newton-Raphson

A partir do estimador recursivo definido na seção 2.3, foi desenvolvido o algoritmo recursivo, demonstrado no Algoritmo 1.

Algoritmo 1 – Estimador recursivo com base em Newton-Raphson.

requer valores necessários para a Equação 5, Equação 8 e para a Equação 10 (quantidade de transmissores, ponto inicial de busca, tamanho do passo de busca, critério de parada, etc.)
garantir gráfico mostrando o resultado de cada iteração do estimador.
$DCFunction\,()$ $\leftarrow$ função que efetua o cálculo de Ω_m da Equação 4 para cada transmissor, recebe como parâmetro os elementos necessários para a equação
$GFunction\,()$ $\leftarrow$ função que calcula a derivada parcial para a matriz Jacobiana, recebe como parâmetro os elementos necessários para a Equação 4 e a posição estimada do receptor
$scatter3\,()$ $\leftarrow$ função do MATLAB que plota ponto no gráfico nas coordenadas passadas como parâmetro
para $i = 1$ até 100 **faça**
  para $m = 1$ até M **faça**
    $\boldsymbol{p}\,[m,1] = DCFunction\,()$
    $\boldsymbol{J}\,[m,1] = GFunction\,(\text{parâmetros para cálculo de x})$
    $\boldsymbol{J}\,[m,2] = GFunction\,(\text{parâmetros para cálculo de y})$
    $\boldsymbol{J}\,[m,3] = GFunction\,(\text{parâmetros para cálculo de z})$
  finaliza para
  $\boldsymbol{r}_R^{i+1} = \boldsymbol{r}_R^i - \eta \cdot \boldsymbol{J}^\dagger \cdot \left(\boldsymbol{s} - \boldsymbol{p}\left(\boldsymbol{r}_R^i\right)\right)$
  $scatter3\left(\boldsymbol{r}_R^{i+1}\,[1], \boldsymbol{r}_R^{i+1}\,[2], \boldsymbol{r}_R^{i+1}\,[3]\right)$
  se $\left|\boldsymbol{r}_R^i - \boldsymbol{r}_R^{i+1}\right| < \epsilon$ **então**
    desvio incondicional (*break*)
  finaliza se
  $\boldsymbol{r}_R^i = \boldsymbol{r}_R^{i+1}$
finaliza para

Fonte: (ASSAKAWA; MATHIAS, 2022).

O algoritmo do pseudocódigo utilizado para o estimador por Newton-Raphson foi o mesmo desenvolvido por Assakawa e Mathias (2022). Também é importante dizer que ϵ é o critério de parada. Este é utilizado com os fins de otimizar o código, assim caso o algoritmo atinja esse critério, o mesmo terminará as iterações com antecedência. M é o número de transmissores, assim como apresentado neste trabalho. As funções $DCFunction\,()$ e $GFunction\,()$ presentes no Algoritmo 1 são as mesmas desenvolvidas por (ŞAHIN *et al.*, 2015b).

3.1.2 Definição da Métrica de Qualidade

Os métodos meta-heurísticos envolvem várias iterações para encontrar uma solução. Assim, para avaliar a qualidade das soluções, pode-se adotar como função de aptidão (*fitness*) a distância euclidiana entre o vetor de potências captadas no receptor e o vetor de potência calculado a partir do modelo do sistema, juntamente com a solução potencial, ou seja,

$$f\left(\boldsymbol{s},\hat{\boldsymbol{p}}\right) = \left(\boldsymbol{s} - \hat{\boldsymbol{p}}\right)^{\mathsf{T}} \cdot \left(\boldsymbol{s} - \hat{\boldsymbol{p}}\right); \tag{19}$$

onde $\hat{\boldsymbol{p}} = \left[\hat{p}_1\left(\hat{\boldsymbol{r}}_R\right); \hat{p}_2\left(\hat{\boldsymbol{r}}_R\right); \cdots; \hat{p}_M\left(\hat{\boldsymbol{r}}_R\right)\right]$ é o vetor que contém as potências estimadas. Quanto menor o valor de $f\left(\boldsymbol{s},\hat{\boldsymbol{p}}\right)$, menor é o erro e melhor é a qualidade da solução. Deste modo o estimador VLP pode ser definido como um problema de minimização. Entretanto, o problema de VLP considerado consiste no fato de que será muito maior a potência capturada de um transmissor bem próximo se comparado com a potência capturada de um transmissor bem mais distante do receptor. Este efeito *near-far* implica em um enviesamento nos estimadores embasados na Equação 19, pois tende a otimizar o resultado para a maior potência capturada em detrimento das demais. Uma forma de corrigir este problema é normalizar estas diferenças das potências:

$$f_w\left(\boldsymbol{s},\hat{\boldsymbol{p}}\right) = \left(\boldsymbol{s} - \hat{\boldsymbol{p}}\right)^{\mathsf{T}} \cdot \boldsymbol{W} \cdot \left(\boldsymbol{s} - \hat{\boldsymbol{p}}\right); \tag{20}$$

onde $\boldsymbol{W} = \operatorname{diag}\left(\left[\frac{1}{\hat{p}_1(\hat{\boldsymbol{r}}_R)^2}; \frac{1}{\hat{p}_2(\hat{\boldsymbol{r}}_R)^2}; \cdots; \frac{1}{\hat{p}_M(\hat{\boldsymbol{r}}_R)^2}\right]\right)$ é uma matriz diagonal. Deste modo, considerando que a origem está em um canto inferior da sala, o problema (já com as restrições) pode ser definido:

$$\hat{\boldsymbol{r}}_R = \arg\min_{\hat{\boldsymbol{r}}_R}\left(f_w\left(\boldsymbol{s},\hat{\boldsymbol{p}}\right)\right); \tag{21}$$

tal que

$$\begin{aligned} p_m\left(\hat{\boldsymbol{r}}_R\right) &> 0; \\ 0 \leq \hat{r}_{R_x} &\leq d_{room_x}; \\ 0 \leq \hat{r}_{R_y} &\leq d_{room_y}; \\ 0 \leq \hat{r}_{R_z} &\leq d_{room_z}. \end{aligned} \tag{22}$$

onde $\boldsymbol{d}_{room} = \left[d_{room_x}; d_{room_y}; d_{room_z}\right]$ é o vetor que contém as dimensões do ambiente a ser analisado. Com isso foi elaborado o Algoritmo 2 que representa o pseudocódigo do método que será utilizado pelos métodos meta-heurísticos para calcular o *fitness* de maneira ponderada.

Algoritmo 2 – Função para calcular o *fitness* ponderado.

requer $\boldsymbol{r}_R$, $\boldsymbol{n}_R$, $\boldsymbol{r}_m$, $\boldsymbol{n}_m$, M, A_{PD}, θ_m, n_L, P, $n\_population$ que é o tamanho da população presente no algoritmo meta-heurístico, e $\boldsymbol{s}$.
garantir que o *fitness* está sendo calculado para todos os indivíduos da população do algoritmo meta-heurístico.

$sum\,() \leftarrow$ função do MATLAB que soma todos os elementos de uma matriz.
$normalized\_s = \frac{\boldsymbol{s}^{-1}}{sum(\boldsymbol{s}^{-1})}$
para $i = 1$ até $n\_population$ **faça**
  para $m = 1$ até M **faça**
    $\Omega = DCFunction\,()$
    $optical\_power\_population = \Omega \cdot P$
  finaliza para
  $weighted\_error = (optical\_power\_population\,[:, i] - \boldsymbol{s}) \cdot normalized\_s$
  $fitness\,[i,1] = weighted\_error^{\intercal} \cdot weighted\_error$
finaliza para
retorna $fitness$

Fonte: Autoria própria (2023).

3.1.3 Desenvolvimento do Algoritmo KNN

Ao desenvolver o algoritmo KNN para otimizar o problema de minimização proposto, conforme demonstrado na Equação 21, foi necessário fazer algumas adaptações, como é possível observar no Algoritmo 3, de forma a tornar o algoritmo mais adequado ao problema e permitir sua utilização para estimar a localização do receptor.

Com isso em mente, o algoritmo foi projetado para criar inicialmente uma vizinhança a partir de um ponto inicial, e aplicar o KNN para encontrar o melhor ponto dentro dessa vizinhança. Em seguida, é criada uma nova vizinhança em torno do novo melhor ponto encontrado. Esse processo se repete por algumas iterações, resultando no melhor ponto encontrado, chamado de $global\_best$ no algoritmo. É importante mencionar que a primeira iteração é realizada com pontos aleatórios e, a partir daí, de forma recursiva, os melhores pontos são encontrados.

Essa abordagem apresenta alguns problemas. Como são utilizados pontos aleatórios, é necessário garantir a restrição desses pontos dentro da sala, conforme demonstrado na Equação 22. Portanto, é preciso garantir esse comportamento no algoritmo. Além disso, a construção de uma nova vizinhança a cada iteração precisa ser uma distribuição uniforme, para garantir que a probabilidade de gerar qualquer vizinho em um intervalo contido no espaço amostral seja proporcional ao tamanho da vizinhança. Uma possível forma de garantir a distribuição uniforme é

dada pela Equação 23:

$$\hat{\boldsymbol{r}}_R^k = \hat{\boldsymbol{r}}_R^1 \cdot (1 + k^{dc} \cdot (2 \cdot \boldsymbol{rand} - 1) \cdot R_{\text{KNN}}); \tag{23}$$

onde k representa o índice do vizinho, dc é a constante de decaimento (*decay constant*), $\boldsymbol{rand}$ é um vetor coluna 3×1 com valores aleatórios de 0 a 1, e R_{KNN} é a amplitude máxima do raio da vizinhança. A constante dc é utilizada para otimizar o algoritmo, de forma que a vizinhança esteja mais dispersa no início e, a cada iteração, a vizinhança gerada esteja cada vez mais próxima, aumentando assim a possibilidade de encontrar um resultado com maior precisão.

A Equação 11 foi utilizada como parte do KNN para calcular a distância entre os dados e classificá-los. Como é possível observar no Algoritmo 2, esse resultado é retornado de forma ponderada, solucionando o problema mencionado na subseção 2.4.1. Portanto, essa equação é utilizada no algoritmo para auxiliar na estimativa. O Algoritmo 2 é chamado no Algoritmo 3 com o nome de $fitness\_function\,()$.

Além disso, a variável $n\_neighbors$ no Algoritmo 3 representa o tamanho da população utilizada pelo KNN, ou seja, o tamanho da vizinhança. Conforme mencionado na subseção 3.1.4, o tamanho da população para todos os algoritmos meta-heurísticos desenvolvidos nesta monografia é de 50.

Por fim, a escolha de criar uma vizinhança menor em torno do melhor ponto encontrado a cada iteração, em vez de criar uma grande vizinhança que abranja toda a região de dispersão dos dados e executar tudo em uma única iteração, visa otimizar o código. Assim, foi projetado o Algoritmo 3.

Algoritmo 3 – Estimador recursivo com base em KNN.

requer a amplitude de raio da vizinhança, R_{KNN}, e a constante de decaimento dc
garantir que a melhor posição encontrada pelo estimador seja retornada.

$rand(A,B) \leftarrow$ função do MATLAB que retorna uma matriz $A \times B$ com valores aleatórios entre 0 e 1

para $i = 1$ até $n\_iterations$ **faça**
  para $neighbor = 2$ até $n\_neighbors$ **faça**
    $\hat{\boldsymbol{r}}_R[:,neighbor] = \hat{\boldsymbol{r}}_R[:,1] \cdot (1 + neighbor^{dc} \cdot (2 \cdot rand(3,1) - 1) \cdot R_{\text{KNN}})$
  finaliza para

  para $d = 1$ até $n\_dimensions$ **faça**
    se $\hat{\boldsymbol{r}}_R[d] > \boldsymbol{d}_{room}[d]$ **então**
      $\hat{\boldsymbol{r}}_R[d] = \boldsymbol{d}_{room}[d]$
    senão, se $\hat{\boldsymbol{r}}_R[d] < 0$ **então**
      $\hat{\boldsymbol{r}}_R[d] = 0$
    finaliza se
  finaliza para

  $fitness = fitness\_function()$

  para $neighbor = 1$ até $n\_neighbors$ **faça**
    se $fitness[neighbor] < individual\_best\_fitness[neighbor]$ **então**
      $individual\_best[:,neighbor] = \hat{\boldsymbol{r}}_R[:,neighbor]$
      $individual\_best\_fitness[neighbor] = fitness[neighbor]$
    finaliza se

    se $individual\_best\_fitness[neighbor] < global\_best\_fitness$ **então**
      $global\_best = individual\_best[:,neighbor]$
      $global\_best\_fitness = individual\_best\_fitness[neighbor]$
      $\hat{\boldsymbol{r}}_R[:,1] = global\_best$
    finaliza se
  finaliza para
finaliza para

retorna $global\_best$

Fonte: Autoria própria (2023).

3.1.4 Desenvolvimento do Algoritmo PSO

Utilizando os conceitos demonstrados na subseção 2.4.2, foi desenvolvido o Algoritmo 4. Devido à estrutura do PSO, sua aplicação para o problema de minimização proposto na Equação 21 mostrou-se expedita.

O PSO é amplamente utilizado para problemas de minimização. Portanto, ao chamar a função *fitness* para estimar o valor necessário para atingir o mínimo da função, basta utilizar a equação do problema proposto neste trabalho acadêmico. Ou seja, a função de aptidão utilizada pelo PSO é o Algoritmo 2.

Desta forma, devido ao seu comportamento natural, o PSO integra-se facilmente ao problema proposto, sem a necessidade de adaptação, como pode ser observado no Algoritmo 4.

Além disso, $n\_iterations$ representa o número total de iterações a serem executadas por todos os algoritmos meta-heurísticos, sendo utilizado $n\_iterations = 100$. $n\_particles$ representa o número total de partículas no sistema, ou seja, o tamanho da população do PSO. Para todos os algoritmos meta-heurísticos deste trabalho, o tamanho total da população é de 50. $n\_dimensions$ é o número de dimensões do espaço da sala. Portanto, para todos os algoritmos, temos $n\_dimensions = 3$.

Algoritmo 4 – Estimador recursivo com base em PSO.

requer coeficiente de inércia w, coeficiente cognitivo c_1 e coeficiente social c_2
garantir que a melhor posição encontrado pelo estimador seja retornada
$pbest = \hat{\boldsymbol{r}}_R$
$pbest\_fitness = fitness\_function\,()$
$gbest = \hat{\boldsymbol{r}}_R\,[:,1]$
$gbest\_fitness = pbest\_fitness\,[1]$
$v \leftarrow$ calcular velocidades atualizadas com base na Equação 14

para $i = 1$ até $n\_iterations$ **faça**
    $fitness = fitness\_function\,()$

    para $particle = 1$ até $n\_particles$ **faça**
        se $fitness\,[particle] < pbest\_fitness\,[particle]$ **então**
            $pbest\,[:, particle] = \hat{\boldsymbol{r}}_R\,[:, particle]$
            $pbest\_fitness\,[particle] = fitness\,[particle]$
        finaliza se

        se $pbest\_fitness\,[particle] < gbest\_fitness$ **então**
            $gbest = pbest\,[:, particle]$
            $gbest\_fitness = pbest\_fitness\,[particle]$
        finaliza se
    finaliza para

    $\hat{\boldsymbol{r}}_R \leftarrow$ calcular posição da partícula com base na Equação 13
    $v \leftarrow$ calcular velocidades atualizadas com base na Equação 14

    para $d = 1$ até $n\_dimensions$ **faça**
        se $\hat{\boldsymbol{r}}_R\,[d] > \boldsymbol{d}_{room}\,[d]$ **então**
            $\hat{\boldsymbol{r}}_R\,[d] = \boldsymbol{d}_{room}\,[d]$
        senão, se $\hat{\boldsymbol{r}}_R\,[d] < 0$ **então**
            $\hat{\boldsymbol{r}}_R\,[d] = 0$
        finaliza se
    finaliza para
finaliza para

retorna $gbest$

Fonte: Autoria própria (2023).

Por fim, $\hat{\boldsymbol{r}}_R$ presente na linha 1 do algoritmo, é inicializado com valores aleatórios contidos dentro das dimensões da sala a ser analisada, e $fitness\_function\,()$ representa o Algoritmo 2.

3.1.5 Desenvolvimento do Algoritmo ACO

Com base na teoria apresentada na subseção 2.4.3, foi desenvolvido o estimador baseado em ACO, demonstrado no Algoritmo 5. Nesse algoritmo, não foi implementada uma lista de tabu, devido ao fato de que as posições iniciais das formigas são escolhidas aleatoriamente em um espaço com infinitas possibilidades de posicionamento. Seria impossível, para o problema proposto, fazer com que as formigas passem por todos os caminhos possíveis antes de começar a selecionar os caminhos com maior feromônio de maneira probabilística. Portanto, assume-se que as posições aleatórias iniciais das formigas abrangem todos os caminhos existentes, e assim o algoritmo é iniciado.

Onde $n\_ants = 50$, sendo este o número total de formigas, ou seja, o tamanho da população deste algoritmo meta-heurístico, $\hat{\boldsymbol{r}}_R$, presente na linha 1 do algoritmo, é inicializado com valores aleatórios contidos dentro das dimensões do ambiente a ser analisado, e $fitness\_function\,()$ é expressado pelo Algoritmo 2.

Na Equação 15, há uma fração que representa o total de feromônio que uma formiga deixa no caminho, sendo este o somatório de $\Delta\tau$. Esse fator é utilizado no algoritmo com base no valor retornado pela função de aptidão (*fitness function*), conforme apresentado no Algoritmo 2.

Uma vez que a quantidade de feromônio deixada está diretamente relacionada à qualidade do caminho, e quanto menor o caminho, maior a qualidade, pode-se inferir que o valor retornado pela função de aptidão representa a qualidade do caminho. Portanto, o objetivo do algoritmo é otimizar essa função, buscando minimizar o valor retornado para obter um caminho de maior qualidade, e assim estimar a posição do receptor de luz.

Algoritmo 5 – Estimador recursivo com base em ACO.

requer σ, β e ρ
garantir que a melhor posição encontrada pelo estimador seja retornada.

$ant\_best = \hat{\boldsymbol{r}}_R$
$ant\_best\_fitness = fitness\_function\,()$
$global\_best = \hat{\boldsymbol{r}}_R\,[:,1]$
$global\_best\_fitness = ant\_best\_fitness\,[1]$
$pheromone \leftarrow$ inicializar matriz de feromônio com 1s

para $i = 1$ até $n\_iterations$ **faça**
  $pheromone \leftarrow$ atualizar matriz de feromônio com base na Equação 15

  para $ant = 1$ até n_ants **faça**
    $probabilities \leftarrow$ calcular probabilidades de seleção com base na Equação 17
    $next\_position \leftarrow$ selecionar próxima posição do receptor de luz com base no algoritmo *roulette wheel selection* como demonstrado por Lipowski e Lipowska (2012)

    $\hat{\boldsymbol{r}}_R\,[:, ant] = next\_position$
  finaliza para

  $fitness = fitness\_function\,()$

  para $ant = 1$ até n_ants **faça**
    se $fitness\,[ant] < ant\_best\_fitness\,[ant]$ **então**
      $ant\_best\,[:, ant] = \hat{\boldsymbol{r}}_R\,[:, ant]$
      $ant\_best\_fitness\,[ant] = fitness\,[ant]$
    finaliza se

    se $ant\_best\_fitness\,[ant] < global\_best\_fitness$ **então**
      $global\_best = ant\_best\,[:, ant]$
      $global\_best\_fitness = ant\_best\_fitness\,[ant]$
    finaliza se
  finaliza para

  para $ant = 1$ até n_ants **faça**
    para $m = 1$ até M **faça**
      se $ant\_best\,[:, ant]$ é igual a $\hat{\boldsymbol{r}}_R\,[:, ant]$ **então**
        $pheromone\,[m, ant] = pheromone\,[m, ant] + ant\_best\_fitness\,[ant]$
      finaliza se
    finaliza para
  finaliza para
finaliza para

retorna $global\_best$

Fonte: Autoria própria (2023).

4 RESULTADOS E DISCUSSÃO

Antes de comparar o algoritmo clássico com os meta-heurísticos, é necessário ajustar os algoritmos meta-heurísticos. Conforme visto na seção 3.1, os algoritmos meta-heurísticos dependem de parâmetros para seu funcionamento. Portanto, foi realizado um conjunto de testes utilizando uma simulação dos estimadores para ajustar os algoritmos meta-heurísticos, a fim de obter uma resposta mais precisa do sistema.

4.1 Sintonização dos Algoritmos Meta-heurísticos

Para realizar a sintonização dos algoritmos meta-heurísticos, foram considerados os parâmetros demonstrados na Tabela 1, onde $n\_population = n\_ants = n\_neighbors = n\_particles$.

Tabela 1 – Parâmetros utilizados na sintonização dos algoritmos

Parâmetros	Valor
M	4 LEDs
P	1 W
n_L	1
A_{PD}	10^{-4} cm^2
θ_m	90°
$n\_dimensions$	3
$n\_iterations$	100
$n\_population$	50
$\boldsymbol{d}_{room}$	$[5; 4; 3]$ m
$\hat{\boldsymbol{r}}_R$	posições iniciais aleatórias
posição exata do receptor	$[2; 1; 1]$ m
variância do ruído	$3{,}0660 \times 10^{-14}$ W^2

Fonte: Autoria própria (2023).

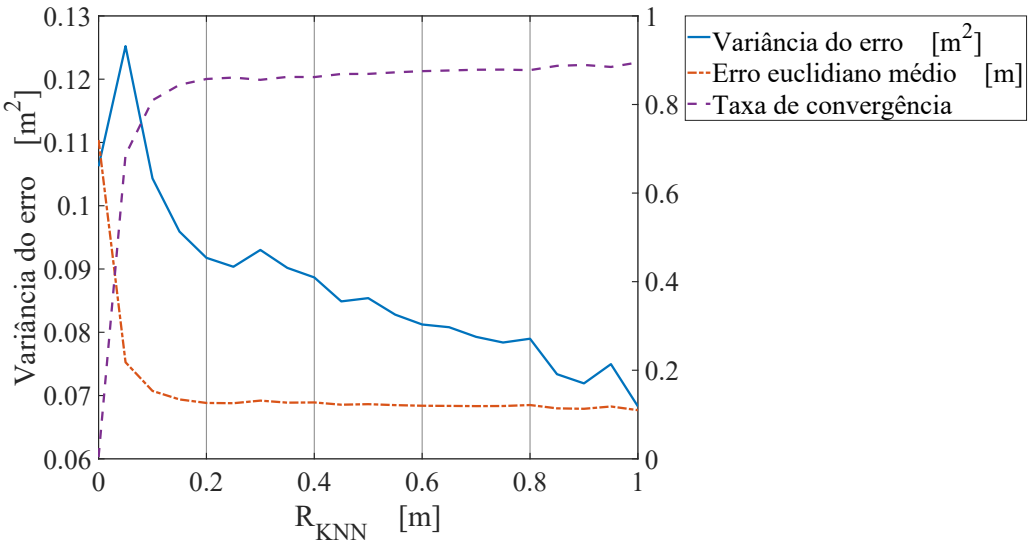
Além disso, os testes foram executados 10.000 vezes para cada ponto da abscissa de cada algoritmo. Nesse contexto, a pilha de testes avaliou a variância, o erro médio em metros e a taxa de convergência dos algoritmos. A taxa de convergência varia de 1 a 0, onde 1 representa 100% e 0 representa 0%. O critério de convergência adotado foi um erro menor que 10 cm. Ou seja, se o resultado apresentou um erro maior do que 10 cm, é considerada a realização como não convergente.

Com o objetivo de obter uma simulação precisa e sintonizar os parâmetros da melhor maneira possível, a posição inicial de $\hat{\boldsymbol{r}}_R$ foi definida de forma aleatória para cada iteração dos testes. Isso garante que os parâmetros sejam sintonizados para qualquer posição inicial de $\hat{\boldsymbol{r}}_R$, em vez de uma posição específica. Além disso, o ruído AWGN foi utilizado no sistema para fornecer um cenário mais robusto e realista. Assim, a variância do ruído é utilizada para garantir que cada amostra de cada iteração dos testes seja descorrelacionada das demais.

4.1.1 Otimizando Parâmetros do KNN

Seguindo a proposta da seção 4.1, são conduzidas uma série de testes no algoritmo KNN, conforme apresentado na Figura 6. Nessa primeira simulação, é utilizado um valor arbitrário para dc e variado o parâmetro R_{KNN} para determinar o melhor valor. Após análise dos resultados de variância, erro médio e taxa de convergência, o valor escolhido para R_{KNN} foi 1, pois apresenta os melhores resultados, como demonstrado na Figura 6.

Figura 6 – Variância, erro médio e taxa de convergência do estimador baseado em KNN com o parâmetro $dc = -0{,}5$ e R_{KNN} variável.



Fonte: Autoria própria (2023).

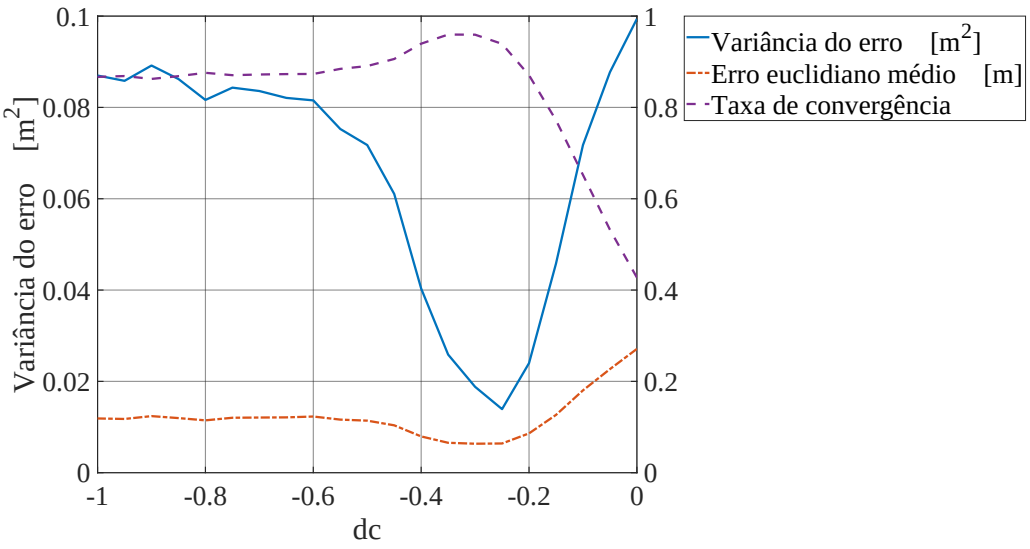
Na Figura 6, os valores representados à esquerda do eixo y correspondem à variância do erro médio euclidiano, enquanto os valores à direita no eixo y são o erro médio euclidiano e a taxa de convergência.

É importante mencionar que, nessa análise, o valor de R_{KNN} foi limitado a no máximo 1 metro. Isso ocorre porque se trata de um algoritmo de localização interna e, se o raio de vizinhança for muito grande, há o risco de o algoritmo se afastar da região de dispersão dos dados, que corresponde ao tamanho da sala ($\boldsymbol{d}_{room}$). Portanto, os valores escolhidos para a abscissa variam de 0 a 1.

Ao realizar os testes para o parâmetro dc, obtivemos a Figura 7. Os melhores valores encontrados estão representados na Tabela 2. Observando $dc = -0{,}25$, é possível notar que este tem a menor variância, enquanto que para $dc = -0{,}35$ é obtido a maior taxa de convergência. Considerando esses resultados, é escolhido $dc = -0{,}3$, pois apresenta um bom equilíbrio entre ambos os aspectos, com uma taxa de convergência satisfatória e uma variância baixa. Além disso, ele possui o menor erro médio.

Assim, a fim de garantir a convergência dos parâmetros para o seu melhor valor possível, é efetuado novamente um teste variando R_{KNN}, mas utilizando o melhor valor de dc encontrado,

Figura 7 – Variância, erro médio e taxa de convergência do estimador baseado em KNN com o parâmetro $R_{\text{KNN}} = 1$ e dc variável.



Fonte: Autoria própria (2023).

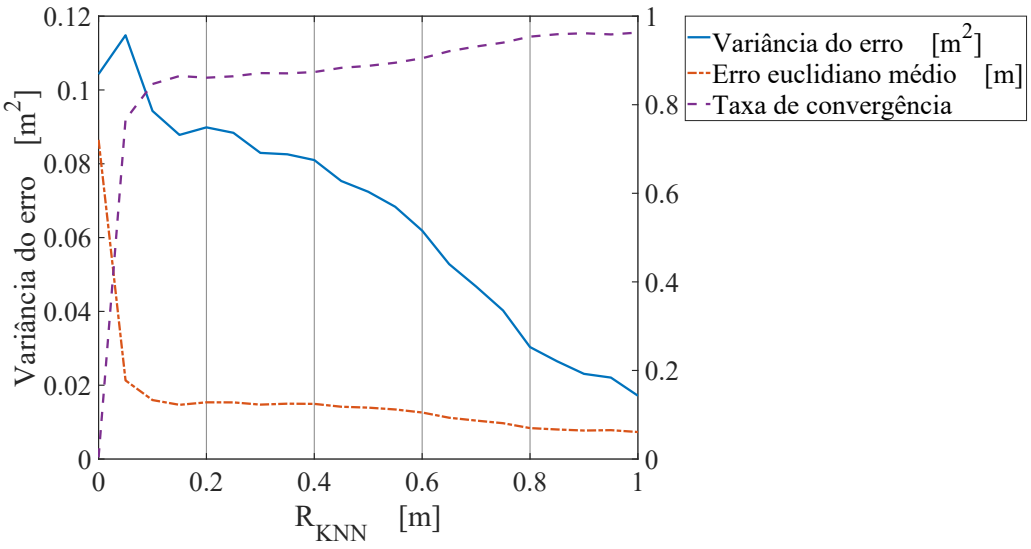
Tabela 2 – Melhores valores de dc encontrados na Figura 7

dc	Variância do erro [cm²]	Erro euclidiano médio [cm]	Taxa de convergência [%]
$-0{,}35$	258,696	6,55062	95,96
$-0{,}30$	188,102	6,34724	95,93
$-0{,}25$	139,225	6,40049	93,92

Fonte: Autoria própria (2023).

que pode ser visto na Figura 8. Analisando a figura é notório que de fato $R_{\text{KNN}} = 1$ se mantém como o melhor valor possível, então este é mantido.

Figura 8 – Variância, erro médio e taxa de convergência do estimador baseado em KNN com o parâmetro $dc = -0{,}3$ e R_{KNN} variável.

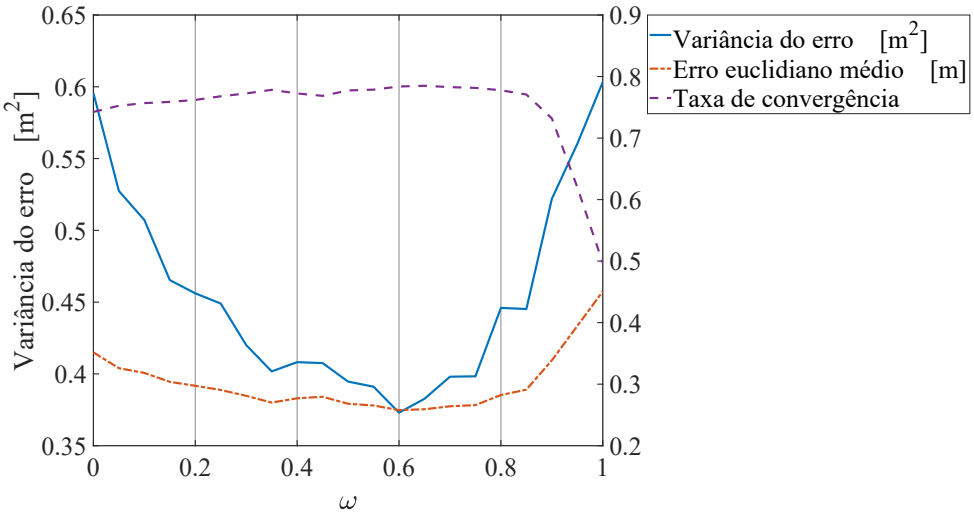


Fonte: Autoria própria (2023).

4.1.2 Otimizando Parâmetros do PSO

Por fim, basta sintonizar os parâmetros do Algoritmo 4. Como visto na Figura 9, inicialmente é utilizado $C_1 = C_2 = 1{,}5$, e então variado ω. Com isso é obtido o valor de 0,6 como melhor valor para ω.

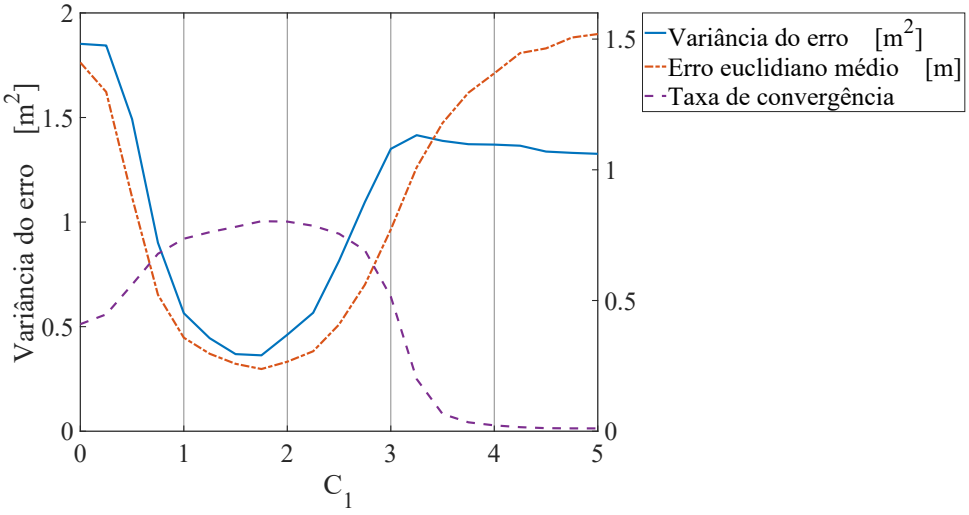
Figura 9 – Variância, erro médio e taxa de convergência do estimador baseado em PSO com os parâmetros $C_1 = 1{,}5$, $C_2 = 1{,}5$ e ω variável.



Fonte: Autoria própria (2023).

Com o melhor valor de ω encontrado, é então variado o valor de C_1, mantendo $C_2 = 1{,}5$, desta forma é aferido que o melhor valor para C_1 é 1,75, como visto na Figura 10.

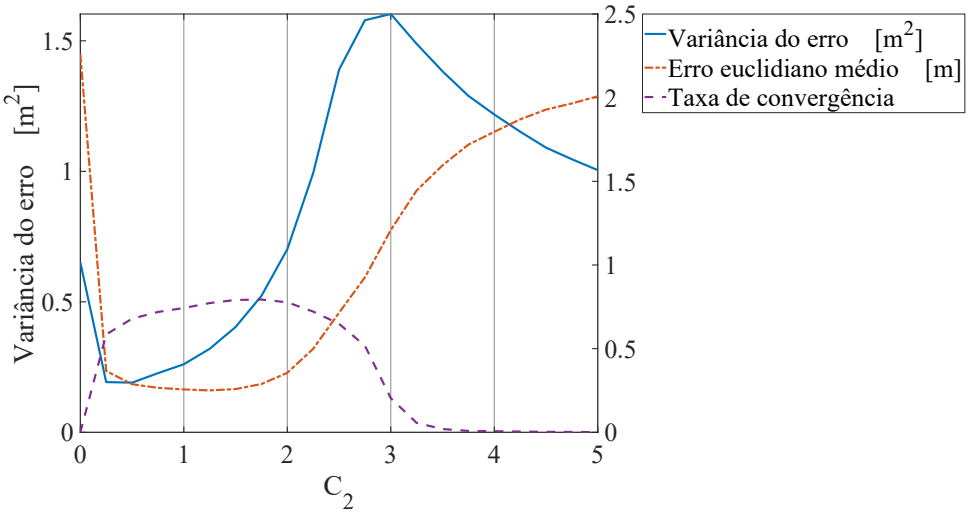
Figura 10 – Variância, erro médio e taxa de convergência do estimador baseado em PSO com os parâmetros $C_2 = 1{,}5$, $\omega = 0{,}6$ e C_1 variável.



Fonte: Autoria própria (2023).

Com ω e C_1 definidos, basta variar C_2, com base nisso é obtida a Figura 11. Nela é possível notar que os melhores valores para C_2 estão entre 0,5 e 1,5, sendo o primeiro o que possui a menor média de erros, e o segundo o que possui a maior taxa de convergência. Entretanto, o ponto de 1,5 traz consigo uma grande variância. Dessa forma é escolhido o valor de 1 para C_2.

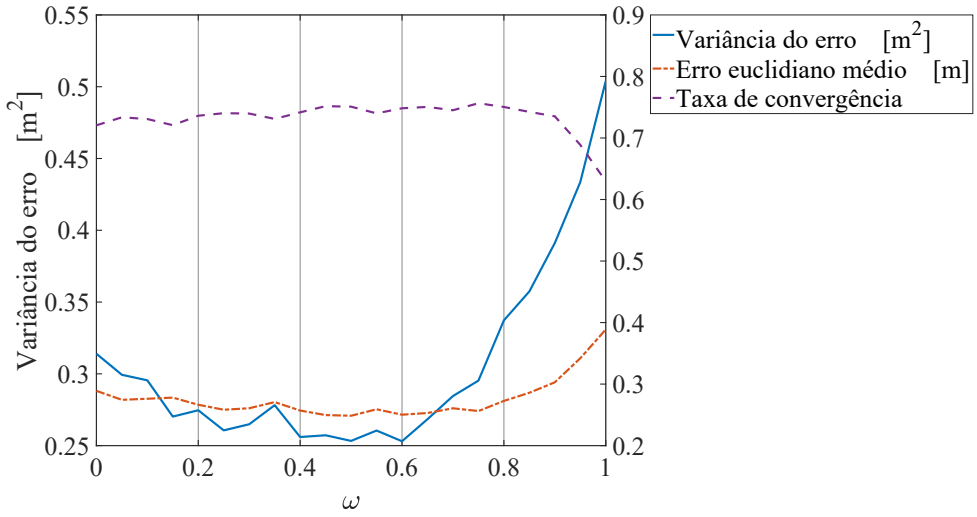
Figura 11 – Variância, erro médio e taxa de convergência do estimador baseado em PSO com os parâmetros $C_1 = 1{,}75$, $\omega = 0{,}6$ e C_2 variável.



Fonte: Autoria própria (2023).

A fim de garantir a convergência dos parâmetros para o melhor grupo de parâmetros possíveis, é conduzido um último teste utilizando o melhor C_1 e C_2 encontrado e variando o ω, como expresso na Figura 12, desta forma é possível notar de que 0,6 continua sendo o melhor valor passível para ω.

Figura 12 – Variância, erro médio e taxa de convergência do estimador baseado em PSO com os parâmetros $C_1 = 1{,}75$, $C_2 = 1$ e ω variável.

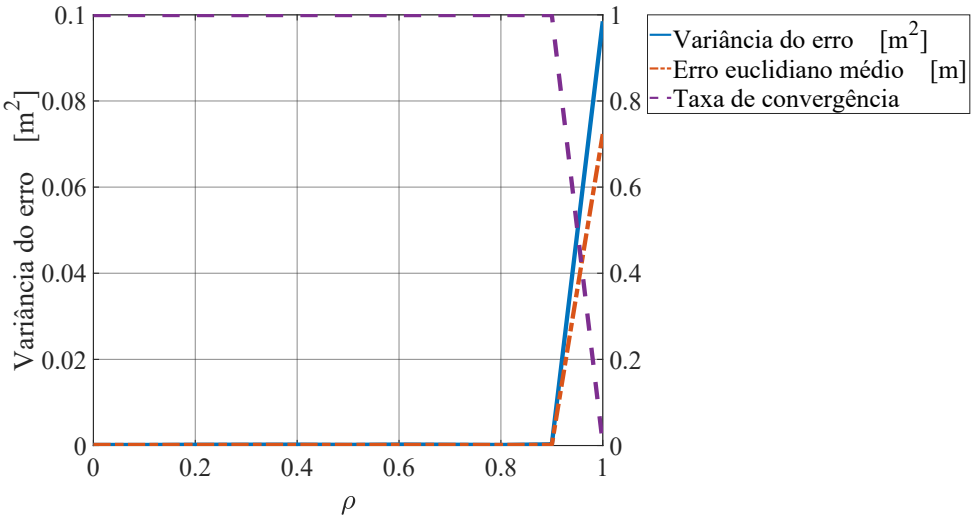


Fonte: Autoria própria (2023).

4.1.3 Otimizando Parâmetros do ACO

Como mostrado no Algoritmo 5, os parâmetros necessários para o funcionamento deste código são σ, β e ρ. Portanto, foi realizada uma primeira série de testes onde foram escolhidos valores arbitrários para σ e β, sendo estes 5 e 1, respectivamente. Com esses valores definidos, o valor de ρ foi variado, permitindo a escolha do melhor ρ com base na variância do erro médio euclidiano, no erro médio euclidiano e na taxa de convergência, como pode ser observado na Figura 13.

Figura 13 – Variância, erro médio e taxa de convergência do estimador baseado em ACO com os parâmetros $\sigma = 5$, $\beta = 1$ e ρ variável

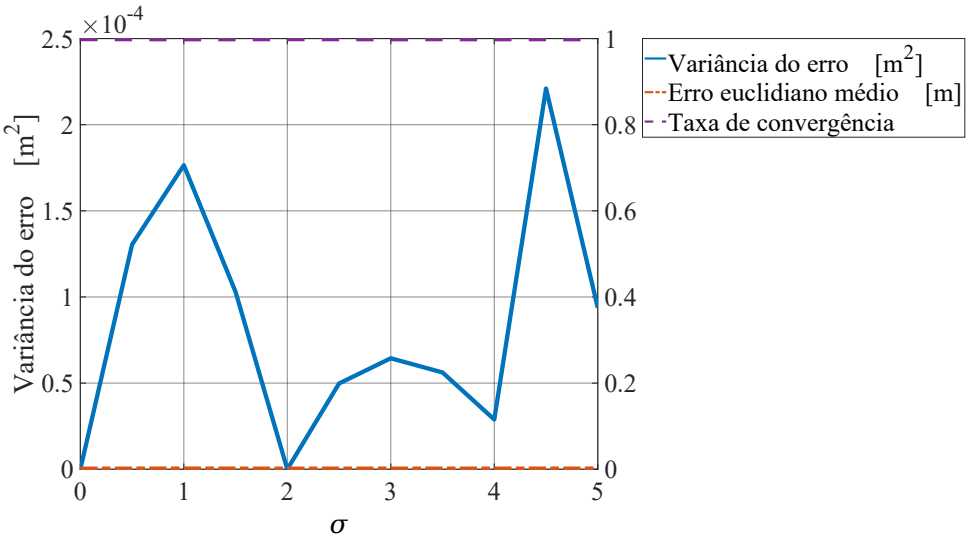


Fonte: Autoria própria (2023).

Ao analisar a Figura 13, pode-se observar que os valores tanto da variância quanto do erro são zero e a taxa de convergência é de 100% para ρ variando de 0 a 0,9. Portanto, foi escolhido o valor de 0,5 para ρ, por ser um valor intermediário na abscissa.

Em seguida, foi realizada uma série de testes semelhantes, variando o valor de σ, utilizando o melhor valor de ρ encontrado e um valor arbitrário para β, como mostrado na Figura 14. Nesse contexto, o valor escolhido para σ foi 2, pois foi o único valor que apresentou uma variância igual a zero, considerando que todos os valores de σ apresentaram taxa de convergência de 100% e erros próximos de zero.

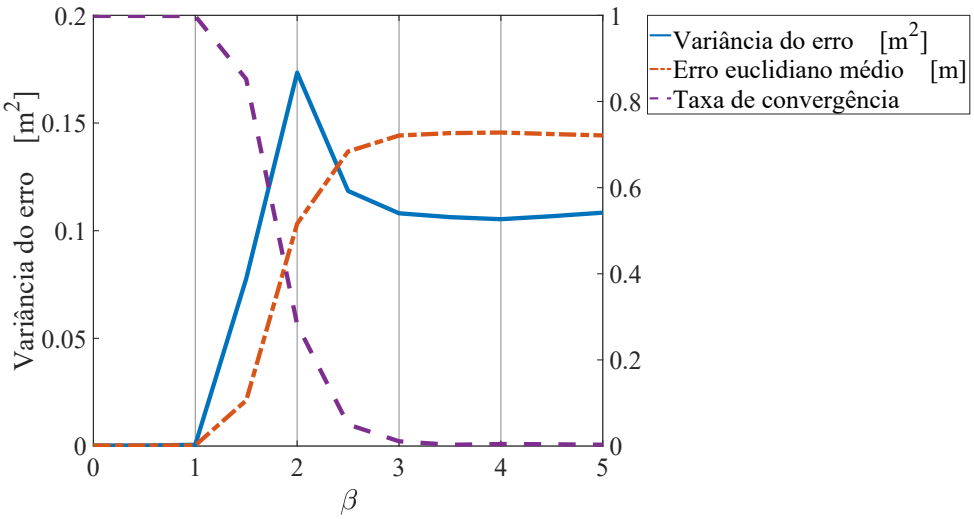
Figura 14 – Variância, erro médio e taxa de convergência do estimador baseado em ACO com os parâmetros $\beta = 1$, $\rho = 0{,}5$ e σ variável.



Fonte: Autoria própria (2023).

O mesmo procedimento é repetido, mas variando o β, utilizando $\sigma = 2$ e $\rho = 0{,}5$, como demonstrado na Figura 15, desta forma é escolhido para β o valor de 0,5, pois este se encontra dentro da região com os melhores valores.

Figura 15 – Variância, erro médio e taxa de convergência do estimador baseado em ACO com os parâmetros $\sigma = 2$, $\rho = 0{,}5$ e β variável.

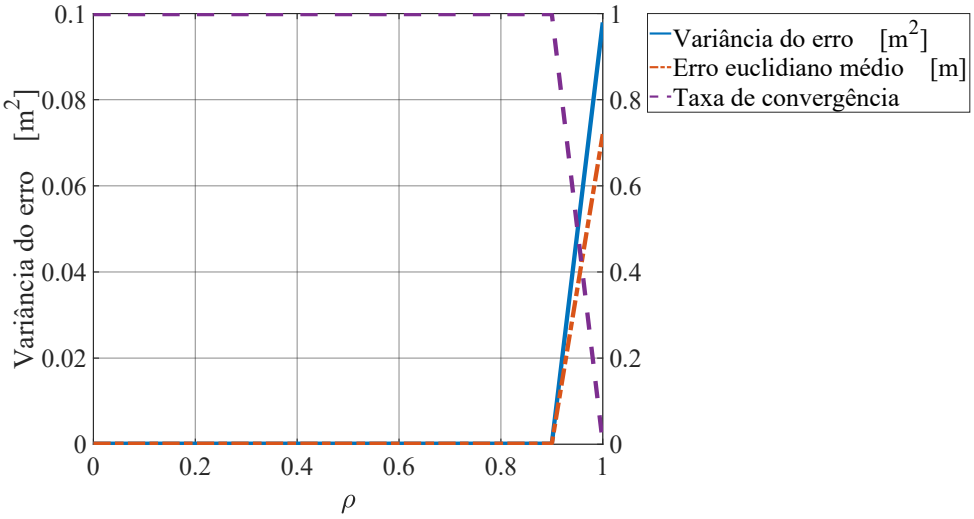


Fonte: Autoria própria (2023).

Para garantir que os parâmetros selecionados convergirão de fato para os melhores possíveis, é feito uma última bateria de testes, agora utilizando $\sigma = 2$, $\beta = 0{,}5$ e variando o ρ, como expresso na Figura 16.

Ao avaliar a Figura 16 é possível identificar que o valor selecionado para ρ continua dentro dos melhores resultados, logo este se manteve.

Figura 16 – Variância, erro médio e taxa de convergência do estimador baseado em ACO com os parâmetros $\sigma = 2$, $\beta = 0{,}5$ e ρ variável.



Fonte: Autoria própria (2023).

4.2 Validação dos Algoritmos

Com os valores dos parâmetros dos algoritmos meta-heurísticos sintonizados, é realizada uma comparação entre eles e o algoritmo clássico (Algoritmo 1). Para esta comparação, cada algoritmo é executado 10.000 vezes e então são calculadas a média dos erros euclidianos, a variância dos erros, a taxa de convergência, a média do tempo de execução e a variância da média do tempo de execução.

Ao executar os algoritmos meta-heurísticos, foram utilizados os parâmetros encontrados na seção 4.1, os quais, estes também podem ser vistos na Tabela 3.

Tabela 3 – Parâmetros dos algoritmos meta-heurísticos.

Algoritmos	Parâmetros
Algoritmo 3 (KNN)	$R_{\text{KNN}} = 1; \quad dc = -0{,}3$
Algoritmo 4 (PSO)	$C_1 = 1{,}75; \quad C_2 = 1; \quad \omega = 0{,}6$
Algoritmo 5 (ACO)	$\sigma = 2; \quad \beta = 0{,}5; \quad \rho = 0{,}5$

Fonte: Autoria própria (2023).

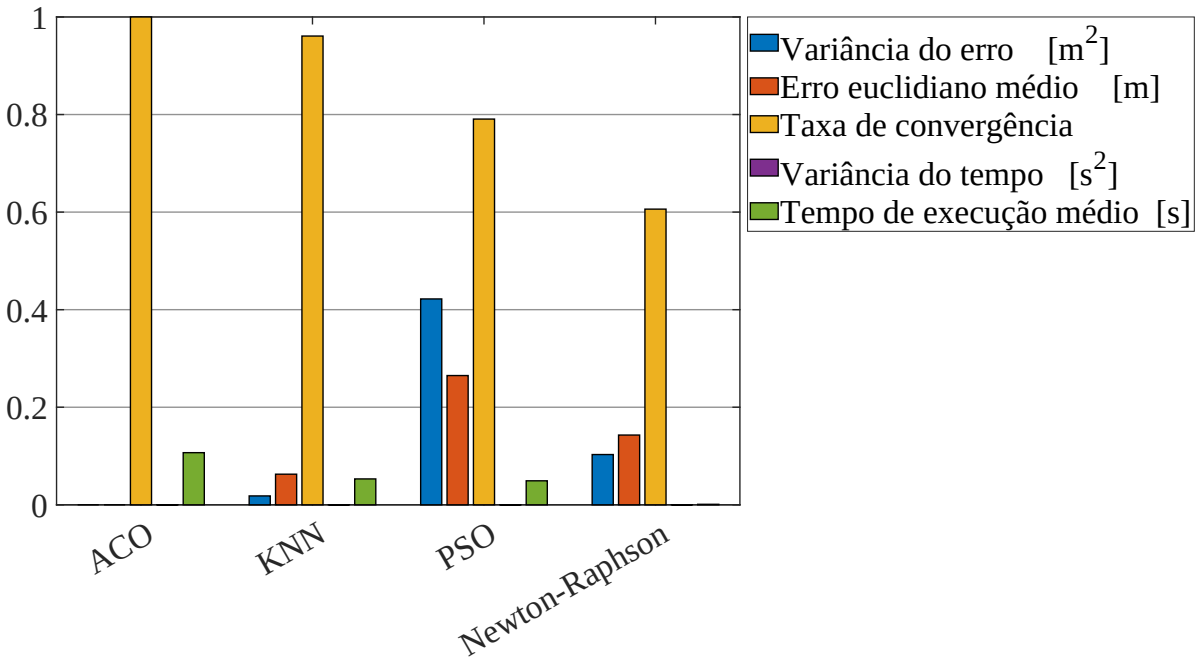
Para o estimador de Newton-Raphson, também foram utilizados os parâmetros demonstrados na Tabela 1, onde $n_{dimensions}$, $n_{iterations}$ e $n_{population}$ são desconsiderados, já que não são utilizados por esse algoritmo.

É importante mencionar, para critérios de medição de tempo, que o algoritmo de comparação utiliza os métodos *tic* e *toc* nativos no MATLAB para calcular o tempo de execução de cada algoritmo.

A máquina utilizada para essas simulações foi um *Intel Core i3-5015U* com 16 GB de memória RAM DDR3 de 1600 MHz, 1 TB de SSD SATA 3, placa gráfica *Intel Graphics 5500*,

sistema operacional *Windows 10*, e o *software* utilizado, como citado na seção 2.2, é o MATLAB R2022b.

Figura 17 – Comparação de variância, erro médio, taxa de convergência e tempo de execução médio entre os algoritmos de estimativa.



Fonte: Autoria própria (2023).

Nesse contexto, a comparação entre os algoritmos é realizada e os resultados podem ser visualizados na Figura 17. É importante ressaltar que, devido aos casos de divergência no algoritmo de descida íngreme, foi adotada uma convenção para evitar erros no software. Quando alguma estimativa ultrapassa o valor máximo permitido pelo MATLAB e/ou resulta em um cálculo indefinido, como a divisão por zero, o valor retornado é NaN (*Not-a-Number*), o que afeta o cálculo da média e variância. Portanto, quando ocorre um NaN, o valor é substituído por um erro de 100 metros. Esses valores de 100 metros são utilizados para calcular a taxa de convergência. No entanto, ao calcular o erro médio e a variância, apenas os valores menores que 100 são considerados. Esse problema ocorre exclusivamente no algoritmo de Newton-Raphson, que é o único entre os algoritmos comparados a apresentar essa questão. Por isso, é importante validar a aplicação dos algoritmos meta-heurísticos para o problema de minimização proposto. Os resultados obtidos são apresentados na Figura 17, e os valores correspondentes podem ser encontrados nas Tabelas 4, 5 e 6.

O valor escolhido para substituir o NaN é arbitrário e foi definido como 100 metros. Essa escolha foi feita porque um valor menor, como 10 metros, não representaria adequadamente o conceito de divergência no problema de localização interna. Embora um erro de 10 metros seja significativo em um sistema de posicionamento interno, provavelmente indicaria um cômodo diferente dentro da residência. Um erro a partir de 100 metros, por outro lado, indicaria algo fora

da maioria dos ambientes internos, não se limitando a um simples erro de cômodo, o que se aproxima mais do conceito de divergência do algoritmo de localização.

Tabela 4 – Comparação entres os algoritmos meta-heurísticos e clássico em relação ao erro.

Algoritmos	Variância do erro [cm²]	Erro euclidiano médio [cm]
Algoritmo 1 (Newton-Raphson)	$1030{,}39$	$14{,}299$
Algoritmo 3 (KNN)	$182{,}139$	$6{,}27628$
Algoritmo 4 (PSO)	$4219{,}94$	$26{,}496$
Algoritmo 5 (ACO)	0	0

Fonte: Autoria própria (2023).

Tabela 5 – Comparação entres os algoritmos meta-heurísticos e clássico em relação a taxa de convergência.

Algoritmos	Taxa de convergência [%]
Algoritmo 1 (Newton-Raphson)	$60{,}61$
Algoritmo 3 (KNN)	$96{,}09$
Algoritmo 4 (PSO)	$79{,}07$
Algoritmo 5 (ACO)	100

Fonte: Autoria própria (2023).

Tabela 6 – Comparação entres os algoritmos meta-heurísticos e clássico em relação ao tempo de execução.

Algoritmos	Variância do tempo [ms²]	Tempo de execução médio [ms]
Algoritmo 1 (Newton-Raphson)	$0{,}189172$	$1{,}03562$
Algoritmo 3 (KNN)	$24{,}2695$	$53{,}063$
Algoritmo 4 (PSO)	$30{,}1938$	$49{,}2014$
Algoritmo 5 (ACO)	$69{,}8373$	$106{,}881$

Fonte: Autoria própria (2023).

Se forem considerados os valores de erro maiores ou iguais a 100 no algoritmo de Newton-Raphson, obtém-se a Tabela 7. Como é possível observar, o algoritmo de Newton-Raphson apresenta uma certa sensibilidade dependendo de onde a estimativa vai se iniciar, como a posição inicial do receptor é feita de maneira aleatória, isso pode intensificar as divergências presente nesse estimador.

Tabela 7 – Comparação entres os algoritmos meta-heurísticos e clássico, considerando todos os dados amostral do clássico.

Algoritmos	Variância do erro [cm²]	Erro euclidiano médio [cm]
Algoritmo 1 (Newton-Raphson)	$9{,}9176 \times 10^{92}$	$3{,}3347 \times 10^{44}$
Algoritmo 3 (KNN)	$182{,}139$	$6{,}27628$
Algoritmo 4 (PSO)	$4219{,}94$	$26{,}496$
Algoritmo 5 (ACO)	0	0

Fonte: Autoria própria (2023).

De acordo com a Figura 17, o algoritmo clássico de Newton-Raphson apresentou a pior taxa de convergência em comparação aos outros algoritmos avaliados. Considerando que uma

taxa de convergência maior indica um estimador mais confiável, podemos afirmar com base nos resultados obtidos e apresentados, que o estimador mais confiável é o ACO, enquanto o menos confiável é o algoritmo clássico de Newton-Raphson. Além disso, o KNN, PSO e ACO apresentam taxas de convergência 58,54%, 30,45% e 64,99% maiores, respectivamente, em comparação ao Newton-Raphson.

Com exceção do PSO, os algoritmos meta-heurísticos também demonstraram um desempenho superior ao do Newton-Raphson em relação ao erro médio e à variância do erro médio. Quanto menor for o erro médio, mais acurado é o estimador. Quanto menor for a variância do erro médio, mais preciso é o algoritmo. Esta correlação pode ser conjecturada com base na Figura 18.

Figura 18 – Comparação entre acurácia e precisão.

Fonte: (GONZáLEZ, 2019).

Dessa forma, e conforme a Tabela 4, o PSO se mostrou menos acurado e preciso do que o estimador por descida íngreme, porém, mais confiável. Além disso, tanto o algoritmo clássico quanto o PSO possuem uma média de erro euclidiano em torno de 14 cm e 26 cm, respectivamente, o que representa uma magnitude maior do que o critério de convergência estabelecido (de 10 cm).

Ainda assim, os algoritmos meta-heurísticos apresentaram um tempo de execução maior do que o do estimador clássico. O ACO foi cerca de 103 vezes mais lento, seguido pelo KNN, que foi 51 vezes mais lento, e o PSO, que foi cerca de 47 vezes mais lento. Isso representa um problema, especialmente em um cenário que exige tempo real.

5 CONCLUSÕES

O presente trabalho desenvolveu algoritmos meta-heurísticos para um sistema de posicionamento por luz visível, sendo estes a otimização por enxame de partículas, a otimização por colônia de formigas e o k-vizinhos mais próximos, conforme apresentado na seção 3.1. Foram realizados testes para ajustar esses algoritmos com os melhores parâmetros possíveis, a fim de garantir uma maior acurácia, precisão e confiabilidade, conforme discutido na seção 4.1.

Para fins de comparação, foram conduzidas simulações com os algoritmos meta-heurísticos desenvolvidos e o clássico por Newton-Raphson. Com base nos resultados obtidos e apresentados na Figura 17 e nas Tabelas 4, 5, 6 e 7, pode-se observar que as soluções propostas neste trabalho acadêmico apresentam vantagens em relação ao algoritmo por descida íngreme, principalmente em termos de taxa de convergência. Os algoritmos PSO, KNN e ACO têm taxas de convergência 30,45%, 58,54% e 64,99% maiores, respectivamente, do que o Newton-Raphson, demonstrando assim uma maior confiabilidade.

Tanto o ACO quanto o KNN apresentam uma variância de erros euclidianos mais baixa do que o clássico, o que sugere uma maior exatidão desses algoritmos em comparação ao clássico. No entanto, como mencionado na seção 4.2, todos os algoritmos meta-heurísticos são significativamente mais lentos em termos de tempo de execução do que o algoritmo clássico. O ACO, KNN e PSO são aproximadamente 103 vezes, 51 vezes e 47 vezes mais lentos, respectivamente, do que o clássico.

Portanto, uma possibilidade para trabalhos futuros é implementar estimadores híbridos, utilizando um meta-heurístico e um clássico. Nesse caso, o clássico seria executado primeiro devido à sua velocidade. Caso o clássico não convirja, com base em um critério de convergência, o meta-heurístico seria acionado para garantir a convergência. Dessa maneira, seria possível obter um melhor tempo de execução e taxa de convergência. De forma inversa, outra possibilidade seria aplicar um algoritmo meta-heurístico primeiro, mas com poucas iterações para não consumir muito tempo, com o propósito de melhorar o ponto inicial de busca e, consequentemente, a convergência do algoritmo de Newton-Raphson. Esse processo seguiria uma abordagem semelhante à proposta por Şahin *et al.* (2015b), mas não exigiria uma luminária especial para o AoA.

Além disso, poderiam ser implementados algoritmos meta-heurísticos em um protótipo real para realizar testes empíricos e validar sua aplicabilidade, bem como avaliar os impactos reais dessa lentidão em um ambiente real.

Outra possibilidade de trabalho futuro é implementar algum mecanismo para impor uma condição de parada nos algoritmos meta-heurísticos ou reduzir o número de iterações de forma otimizada. Seria necessário realizar estudos para validar até que ponto é possível diminuir as iterações sem causar um grande impacto na qualidade dos algoritmos em termos de acurácia, precisão e confiabilidade. Isso seria feito para torná-los mais atrativos em comparação com o

Newton-Raphson, otimizando assim os algoritmos meta-heurísticos em relação ao tempo de execução.

Por fim, análises mais aprofundadas dos algoritmos em questão poderiam ser realizadas, como estudar o impacto do tamanho da população em cada algoritmo, a fim de utilizar um tamanho adequado que proporcione um equilíbrio melhor entre taxa de convergência, erro euclidiano médio e variância do erro euclidiano médio em relação ao tempo de execução, visando otimizar esses algoritmos.

REFERÊNCIAS

ANDERSON, J.; AULIN, T.; SUNDBERG, C. **Digital Phase Modulation**. New York: Springer US, 2013. (Applications of Communications Theory). ISBN 9781489920317.

ASIMOV, I. **I, Robot**. New York, NY, USA: Bantam Books, 1950. (Doubleday Science Fiction). ISBN 9780451012821.

ASSAKAWA, A. M.; MATHIAS, L. C. **Sistemas de localização interna em 3D utilizando infraestrutura de iluminação LED**. 2022. 16-56 p. Trabalho de Conclusão de Curso.

CLERC, M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. *In*: **Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)**. Washington: IEEE, 1999. v. 3, p. 1951–1957 Vol. 3.

CORDEIRO, M. C.; MATHIAS, L. C.; ESTEVES, M. P. **Vision Light: Um auxílio na locomoção de deficientes visuais utilizando VLC**. Guarapuava, 2021.

COVER, T.; HART, P. Nearest neighbor pattern classification. **IEEE Transactions on Information Theory**, v. 13, n. 1, p. 21–27, 1967.

De Gante, A.; SILLER, M. A survey of hybrid schemes for location estimation in wireless sensor networks. **Procedia Technology**, v. 7, p. 377–383, 2013. ISSN 2212-0173. 3rd Iberoamerican Conference on Electronics Engineering and Computer Science, CIIECC 2013.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. The ant system: An autocatalytic optimizing process. *In*: . [*S.l.*]: Technical report, 1991.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, v. 26, n. 1, p. 29–41, 1996.

ELGALA, H.; MESLEH, R.; HAAS, H. Indoor optical wireless communication: potential and state-of-the-art. **IEEE Communications Magazine**, v. 49, n. 9, p. 56–62, 2011.

FARUQUE, S. Amplitude modulation (AM). *In*: ____. **Radio Frequency Modulation Made Easy**. Cham: Springer International Publishing, 2017. p. 17–32. ISBN 978-3-319-41202-3.

FIX, E.; HODGES, J. L. Discriminatory analysis. nonparametric discrimination: Consistency properties. **International Statistical Review / Revue Internationale de Statistique**, [Wiley, International Statistical Institute (ISI)], v. 57, n. 3, p. 238–247, 1989. ISSN 03067734, 17515823. Disponível em: http://www.jstor.org/stable/1403797.

GONZáLEZ, M. **O que é acurácia? Entenda o conceito e sua importância**. 2019. BASE Aerofotogrametria. Disponível em: https://blog.idwall.co/o-que-e-acuracia/.

HUANG, C. *et al.* NLOS-aware VLC-based indoor localization: Algorithm design and experimental validation. *In*: **2020 IEEE Wireless Communications and Networking Conference (WCNC)**. Virtual Conference: IEEE, 2020. p. 1–7.

KAY, S. M. **Fundamentals of Statistical Signal Processing: Estimation Theory**. USA: Prentice-Hall, Inc., 1993. ISBN 0133457117.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. *In*: **Proceedings of ICNN'95 - International Conference on Neural Networks**. Perth: IEEE, 1995. v. 4, p. 1942–1948 vol.4.

KESKIN, M. F.; SEZER, A. D.; GEZICI, S. Localization via visible light systems. **Proceedings of the IEEE**, v. 106, n. 6, p. 1063–1088, 2018.

KOMINE, T.; NAKAGAWA, M. Fundamental analysis for visible-light communication system using LED lights. **IEEE Transactions on Consumer Electronics**, v. 50, n. 1, p. 100–107, 2004.

LIN, B. *et al.* Experimental demonstration of an indoor VLC positioning system based on OFDMA. **IEEE Photonics Journal**, v. 9, n. 2, p. 1–9, 2017.

LIN, P. *et al.* Real-time visible light positioning supporting fast moving speed. **Opt. Express**, Optica Publishing Group, v. 28, n. 10, p. 14503–14510, May 2020.

LIPOWSKI, A.; LIPOWSKA, D. Roulette-wheel selection via stochastic acceptance. **Physica A: Statistical Mechanics and its Applications**, v. 391, n. 6, p. 2193–2196, 2012. ISSN 0378-4371. Disponível em: https://www.sciencedirect.com/science/article/pii/S0378437111009010.

LIU, X.; MAKINO, H.; MAEDA, Y. Basic study on indoor location estimation using visible light communication platform. *In*: **2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society**. Vancouver: IEEE, 2008. p. 2377–2380.

MATHIAS, L. C.; ABRãO, T. **Localização, Pré-distorção, Pré-equalização e Mitigação do *Crosstalk* em Sistemas VLC-OFDM**. 2019. Tese (Doutorado) — Universidade Estadual de Londrina, 2019.

MATHIAS, L. C.; MELO, L. F. D.; ABRãO, T. 3-D localization with multiple LEDs lamps in OFDM-VLC system. **IEEE Access**, v. 7, p. 6249–6261, 2019.

MOHAMMED, N. A.; ELKARIM, M. A. Exploring the effect of diffuse reflection on indoor localization systems based on RSSI-VLC. **Opt. Express**, Optica Publishing Group, v. 23, n. 16, p. 20297–20313, Aug 2015.

NGUYEN, H. *et al.* A matlab-based simulation program for indoor visible light communication system. *In*: **2010 7th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP 2010)**. Newcastle upon Tyne: IEEE, 2010. p. 537–541.

RODER, H. Amplitude, phase, and frequency modulation. **Proceedings of the Institute of Radio Engineers**, v. 19, n. 12, p. 2145–2176, 1931.

SAAB, S. S.; SAAB, K. K. A positioning system for photodiode device using collocated LEDs. **IEEE Photonics Journal**, v. 8, n. 5, p. 1–14, 2016.

VAPPANGI, S.; MANI, V. V. Performance analysis of DST-based intensity modulated/direct detection (IM/DD) systems for VLC. **IEEE Sensors Journal**, v. 19, n. 4, p. 1320–1337, 2019.

WANG, T. Q. *et al.* Position accuracy of time-of-arrival based ranging using visible light with application in indoor localization systems. **Journal of Lightwave Technology**, v. 31, n. 20, p. 3302–3308, 2013.

ZHANG, W.; CHOWDHURY, M.; KAVEHRAD, M. Asynchronous indoor positioning system based on visible light communications. **Optical Engineering**, v. 53, p. 045105, 04 2014.

ZHOU, Z.; KAVEHRAD, M.; DENG, P. Indoor positioning algorithm using light-emitting diode visible light communications. **Optical Engineering**, v. 51, p. 5009–, 08 2012.

ZHUANG, Y. *et al.* A survey of positioning systems using visible LED lights. **IEEE Communications Surveys & Tutorials**, IEEE, Virtual Conference, v. 20, n. 3, p. 1963–1988, 2018.

ŞAHIN, A. *et al.* Accuracy of AOA-based and RSS-based 3d localization for visible light communications. *In*: **2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)**. [*S.l.*: *s.n.*], 2015. p. 1–5.

ŞAHIN, A. *et al.* Hybrid 3-D localization for visible light communication systems. **Journal of Lightwave Technology**, v. 33, n. 22, p. 4589–4599, 2015.