



UNIVERSIDADE D
COIMBRA

Sérgio Duarte Correia

ALGORITMOS META-HEURÍSTICOS PARA APLICAÇÃO A
LOCALIZAÇÃO BASEADOS EM SISTEMAS ACÚSTICOS

Tese no âmbito do Doutoramento em Engenharia Eletrotécnica e de Computadores, Ramo de Especialização em Telecomunicações, orientada pelo Professor Doutor Luís Alberto da Silva Cruz e Professor Doutor Marko Beko e apresentada à Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Novembro de 2019

Algoritmos Meta-Heurísticos para Aplicação a Localização Baseada em Sistemas Acústicos



Sérgio Duarte Correia
Faculdade de Ciência e Tecnologia
Universidade de Coimbra

A thesis submitted for the degree of
Doctor of Philosophy

Coimbra 2019

A todos que direta ou indiretamente,
contribuíram para a construção deste trabalho.
A todos que acompanharam e sentiram ...

Agradecimentos

A realização da presente dissertação contou com importantes apoios de um vasto conjunto de pessoas, sem as quais não teria sido possível a realização de todos os passos associados a um trabalho de investigação.

Ao Prof. Luís A. da Silva Cruz, pela sua orientação, apoio e disponibilidade para solucionar dúvidas e problemas que normalmente surgem no decurso do longo caminho a percorrer. Os seus valiosos conselhos permitiram atingir o culminar de anos de trabalho.

Ao Prof. Marko Beko, pela sua orientação, motivação e apoio. A sua orientação do ponto de vista científico, foi fundamental para conduzir este trabalho e chegar ao que se espera, ser o início de uma carreira de investigação. A sempre incansável disponibilidade e conselhos, quer a nível profissional, quer a nível pessoal, ficarão na memória.

Um especial agradecimento ao Prof. Slavisa Tomic, pelo seu apoio ao nível técnico-científico, pelas sugestões, discussão e debate de ideias, que em muito enriqueceram o presente trabalho.

Ao Instituto Politécnico de Portalegre, à Escola Superior de Tecnologia e Gestão e ao Departamento de Tecnologias, por sempre contribuírem com o apoio possível, no sentido de providenciar as melhores condições de trabalho. Por considerarem a formação dos docentes nas suas políticas, ao nível da gestão de uma instituição de ensino superior.

Ao Instituto de Telecomunicações (Pólo de Coimbra) e ao COPELABS - Cognitive and People-Centric Computing, pelo apoio prestado ao nível de infraestruturas físicas e computacionais.

Resumo

A presente dissertação aborda o problema da localização no espaço de uma fonte acústica com o recurso a algoritmos meta-heurísticos. Tradicionalmente, as metodologias adotadas neste contexto baseiam-se em aproximações obtidas a partir de um estimador estatístico (não convexo), aplicando-se posteriormente métodos de procura determinísticos. Tais métodos implicam normalmente soluções eficientes à custa de um elevado poder de computação podendo, no entanto, apenas fornecer soluções subótimas.

Através da aplicação de métodos meta-heurísticos - nomeadamente métodos baseados em enxames de partículas - o problema de localização é neste trabalho abordado de forma direta (sem recurso a aproximações), recorrendo-se a uma metodologia de procura global no espaço. Embora conceptualmente estes algoritmos sejam desenvolvidos e propostos no contexto de problemas genéricos, sem explorar particularidades da aplicação alvo de estudo, o presente trabalho analisa, desenvolve, e valida novas metodologias, de modo a obter resultados ótimos e/ou subótimos com reduzido erro mas, sobretudo, acelerando consideravelmente a convergência dos mesmos.

Inicialmente são estudados métodos de otimização global baseados em enxames de partículas, procurando ajustar-se os seus parâmetros, de modo a serem obtidas soluções comparáveis ao atual estado da arte. Considera-se um largo espectro contextual, nomeadamente ao nível do ruído das medições, do número de sensores e da posição da fonte acústica no espaço. Os resultados obtidos certificam a utilização da metodologia, quer em termos do erro da solução final, quer em termos do tempo de computação. Estes são ainda validados com uma implementação em ambiente real, com medições de campo, obtendo-se resultados em concordância com as simulações computacionais.

A segunda parte da dissertação propõe novas técnicas para a inicialização da população normalmente considerada aleatória neste tipo de cenários. Para esse fim, dois métodos são desenvolvidos e propostos: (1) usando uma estimativa de distância obtida através das observações ruidosas do modelo, gerando pontos aleatórios em torno das interceções em relação aos sensores; (2) pela geração de Cadeias de *Markov* a partir do algoritmo de *Metropolis-Hastings*, tendo como ponto inicial o resultado obtido em (1). Em simultâneo, implementa-se uma procura local baseada

no gradiente discreto do modelo. Como complemento, através da aplicação de uma nova condição de paragem dos métodos, que normalmente consideram um valor pré-definido de gerações de população, é demonstrado que as propostas no âmbito desta dissertação aceleram consideravelmente a convergência dos mesmos. Tal situação implica um nível de computação consideravelmente reduzido em relação ao número de gerações de população e, conseqüentemente, do tempo de processamento.

O presente trabalho introduz diversas inovações importantes no campo de estudo: (1) incorporação de informação do modelo objeto de estudo nos métodos de otimização baseados em enxames de partículas; (2) abordagens inovadoras para inicialização de populações aplicadas em métodos baseados em enxames de partículas; (3) um critério de paragem mais eficiente, de modo a interromper o processo iterativo normalmente considerado com valores pré-determinados empiricamente; (4) uma estratégia de procura local, sem a necessidade do cálculo analítico do gradiente do modelo.

Tais inovações resultam em novas metodologias com um desempenho mais eficiente ao nível da localização em termos de precisão, complexidade computacional, e taxa de convergência. As inovações introduzidas no estado da arte para resolução do problema de localização de uma fonte acústica abrem caminho para implementações "*embedded*", em processadores de baixa complexidade, reduzido fator de forma, e consumo. Deste modo, surgem novas possibilidades de investigação e desenvolvimento ao nível da computação de borda ou em nuvem (do inglês *edge* ou *fog computing*), assim como a implementação de métodos de localização distribuídos ou sequenciais, quer pela simplicidade de implementação, quer pela menor exigência a nível computacional.

Abstract

The present dissertation addresses the problem of localization of an acoustic source using metaheuristic algorithms. Typically, existing methodologies adopted for the localization problem are based on a sequence of approximations and/or relaxations of a (non-convex) statistical estimator. These solutions are used later on as starting points for local search, applying deterministic search methods. Although solutions entailed from such methods are relatively accurate, they come at the expense of high computational cost and are only sub-optimal.

In this thesis, through the use of metaheuristic, namely swarm-based methods, the localization problem is tackled directly (without resorting to any approximations/relaxations), using a global optimization methodology. Conceptually, these algorithms are intended to address generic problems, without exploiting particularities of the problem at hand, but the present thesis shows that they can be adopted for problem-specific applications. More particularly, the present work develops novel frameworks for metaheuristic algorithms, which enhance localization performance, both in terms of localization accuracy and algorithm convergence.

In the first part of the dissertation, global optimization methods based on particle swarms are studied, aiming to adjust their parameters in order to match the localization accuracy of state-of-the-art solutions. A wide range of settings of practical interest is considered, namely in terms of measurement noise power, number of sensors, as well as the position of the source in the search space. The simulation results validate the use of the particle swarm optimization methodology, both in terms of localization accuracy and execution time. These results are also validated through a real-world implementation using field measurements, which corroborate the results obtained with the computer simulations.

The second part of the thesis proposes new techniques for population initialization, which is performed entirely randomly in the traditional approach. To this end, two methods are developed: (1) by exploiting distance estimates from noisy model observations to form circles centered at the known reference locations with radii equal to the respective distance estimates to generate random points in the neighborhood of the intersections of the circles; (2) by generating Markov Chains and employing Metropolis-Hastings algorithm, taking the result obtained in (1) as the starting point. Simultaneously, a local search based on a discrete gradient of

the model is implemented. Moreover, by adopting a different stopping criterion in comparison with the traditional one (an empirical predefined value for the number of function evaluations), it will be shown that the presented approaches in this thesis considerably accelerate their convergence rates. This result implies a reduced number of population generations and, consequently, a reduced processing time.

Therefore, this work introduces several essential innovations in the field of study, namely: (1) incorporation of the problem particularity under study into global optimization methods, (2) a novel approach for population initialization for swarm-based methods, (3) more efficient stopping criteria to interpose the iteration process, traditionally considered as a predetermined value, and (4) a simple, but efficient local search methodology which does not require (tedious) analytic gradient calculations.

Such innovations result in new methodologies, with enhanced localization performance in terms of localization accuracy, computational complexity, and convergence rate. Therefore, the proposed innovations introduced for solving the localization problem of an acoustic source paves the way for embedded implementations on low complexity processors with small form factor and low consumption. This way, not only does this work open up new research paths at the level of edge or cloud computing, but it is also suitable for distributed implementation or sequential localization schemes. These new research and development possibilities are mainly due to the simplicity of implementing the proposed methodology, which comes at a low computational effort.

Índice

Lista de Figuras	xiii
Lista de Tabelas	xvii
Lista de Abreviações	xix
1 Introdução	1
1.1 Enquadramento	1
1.2 Conceitos Introdutórios	3
1.3 Organização do Documento	5
1.4 Publicações	7
2 Métodos de Otimização Meta-Heurísticos para Solução do Problema de Localização de uma Fonte Acústica	9
2.1 Motivação e Objetivos	9
2.2 Enquadramento do Problema	11
2.3 Formulação do Problema	14
2.4 Algoritmos de Otimização Meta-Heurísticos	17
2.4.1 Resultados Preliminares	18
2.4.2 Algoritmo <i>Elephant Herding Optimization (EHO)</i>	22
2.4.3 Sintonização dos Parâmetros do Método EHO	24
2.5 Análise de Resultados	29
2.5.1 Análise de Complexidade	29
2.5.2 Resultados de Simulação	31
2.6 Validação Experimental	33
2.6.1 Simulação em Ambiente Real	33
2.6.2 Resultados Experimentais	36
2.7 Conclusões	46
3 Desenvolvimento de Metodologias para o Aperfeiçoamento de Algoritmos Meta-Heurísticos	49
3.1 Motivação e Objetivos	49
3.2 Inicialização Inteligente da População	50

3.3	Métodos Baseados no Gradiente	57
3.4	Algoritmo Híbrido	59
3.5	Simulações e Análise de Resultados	62
3.5.1	Aplicação ao Método Original	63
3.5.2	Aplicação à mais Recente Versão do Método EHO	72
3.5.3	Aplicação ao Método " <i>Coyote Optimization</i> <i>Algorithm</i> "	77
3.6	Conclusões	82
4	Inicialização Inteligente da População pelo Método de <i>Monte Carlo</i>	85
4.1	Enquadramento e Objetivos	85
4.2	Geração do Estado Inicial da Cadeia de Markov	86
4.3	Cadeias de Markov a partir do Algoritmo de Metropolis-Hastings	88
4.4	Resultados e Discussão	95
4.5	Conclusão	97
5	Conclusão	99
5.1	Conclusão	99
5.2	Trabalho Futuro	101
Anexos		
A	Demonstrações	105
A.1	Demonstração da Proposição 3.1	105
A.2	Demonstração da Proposição 3.2	108
B	Histogramas e Indicadores Estatísticos (Cap. 3)	111
B.1	Histogramas Correspondes ao Caso de Estudo	111
C	Histogramas e Indicadores Estatísticos	117
C.1	Extensão dos Resultados a $N=6$ e $N=15$	117
Referências		123

Lista de Figuras

1.1	Taxonomia das tecnologias de localização	4
2.1	Representação gráfica da função de custo do modelo. (a) Configuração de sensores e fonte. (b) Gráfico de superfície do modelo	16
2.2	Resultados de simulação para PSO, ABC, EFWA, SOCP e EHO, com $N = 9$ sensores	20
2.3	Resultados de simulação para PSO, ABC, EFWA, SOCP e EHO, com $N = 12$ sensores	20
2.4	Análise de variabilidade do parâmetro α . (a) Representação da solução final no espaço de procura; "•" representa a posição verdadeira. (b) Dependência de α na função de custo.	24
2.5	RMSE função de α , com $\sigma_v^2 = -25$ dB e $\beta = 0.1$	25
2.6	RMSE função de α para diferentes valores de β	26
2.7	Dependência do RMSE com o número de clãs e n_{c_i}	27
2.8	Dependência do RMSE com o número de clãs e n_{c_i} , mantendo a população de tamanho constante.	28
2.9	Distribuição aleatória de 10000 fontes ($N = 9$)	31
2.10	RMSE em função de σ_v^2 (dB) para $N = 9$	32
2.11	RMSE em função de σ_v^2 (dB) para $N = 12$	33
2.12	Condições de implementação experimental. (a) Condições de simulação para $N = 12$ sensores e 1 fonte . (b) Distribuição das 10000 fontes posicionadas aleatoriamente.	34
2.13	Resultados de simulação para $N = 3, 6, 9$ e 12 sensores e σ_v^2 (dB) = $-80 : 5 : -45$	35
2.14	Diagrama de blocos da plataforma de aquisição de dados	37
2.15	Campo de medições para validação do modelo	39
2.16	Circuito de polarização do microfone	40
2.17	Plataforma de aquisição (circuito do microfone e placa Arduino UNO)	40
2.18	Sinais medidos em três distâncias diferentes	41
2.19	FFT do sinal medido a 5m	42
2.20	Projeto do filtro digital (<i>Filter Designer</i> -MATLAB®)	43
2.21	Modelo teórico e experimental do modelo de decaimento acústico	43

2.22	Medidas de campo para o algoritmo de localização EHO	44
2.23	Resultado do algoritmo EHO com dados reais. (a) $N = 3$ sensores e 1 fonte (b) $N = 6$ sensores e 1 fonte (c) $N = 9$ sensores e 1 fonte (d) $N = 12$ sensores e 1 fonte	45
3.1	Possíveis configurações segundo a distância. (a) Circunferências secantes (b) Circunferências externas	52
3.2	Diferentes combinações, dependentes do ruído das observações. (a) Uma interceção (b) Duas interceções	54
3.3	Exemplo de uma geração de clãs, $N = 9$, $X = (30, 70) m$ e $\sigma_v^2 = -40 dB$. (a) Clãs e população gerada (b) Visualização no espaço alargado	56
3.4	Fluxograma do algoritmo proposto perante o algoritmo original	61
3.5	Resultados de simulação considerando $N = 9$ sensores	63
3.6	Resultados de simulação considerando $N = 12$ sensores	64
3.7	Resultados de simulação considerando: (a) $N = 6$ sensores. (b) $N = 15$ sensores	64
3.8	Histograma do número de avaliações para o algoritmo EHO ($N = 9$) com critério de paragem	65
3.9	Histograma do número de avaliações para o algoritmo EHO ($N = 12$) com critério de paragem	66
3.10	Histograma do número de avaliações para o algoritmo iEHO ($N = 9$)	67
3.11	Histograma do número de avaliações para o algoritmo iEHO ($N = 12$)	67
3.12	Histograma do número de avaliações para $N = 6$. (a) Algoritmo EHO (b) Novo Algoritmo (iEHO)	68
3.13	Histograma do número de avaliações para $N = 15$. (a) Algoritmo EHO (b) Novo Algoritmo (iEHO)	68
3.14	Novos operadores do algoritmo EEHO [184]	73
3.15	Resultados de simulação para o algoritmo EEHO ($N = 9$ sensores)	74
3.16	Resultados de simulação para o algoritmo EEHO ($N = 12$ sensores)	75
3.17	Histograma do número de avaliações para o algoritmo EEHO ($N = 9$) com critério de paragem	75
3.18	Histograma do número de avaliações para o algoritmo EEHO ($N = 12$) com critério de paragem	76
3.19	Histograma do número de avaliações para o algoritmo EEHO ($N = 9$) com inicialização de população	76
3.20	Histograma do número de avaliações para o algoritmo EEHO ($N = 12$) com inicialização de população	77
3.21	Resultados de simulação para o algoritmo COA ($N = 9$ sensores)	79
3.22	Resultados de simulação para o algoritmo COA ($N = 12$ sensores)	79

3.23	Histograma do número de avaliações para o algoritmo COA ($N = 9$) com critério de paragem	80
3.24	Histograma do número de avaliações para o algoritmo COA ($N = 12$) com critério de paragem	80
3.25	Histograma do número de avaliações para o algoritmo COA ($N = 9$) com inicialização de população	81
3.26	Histograma do número de avaliações para o algoritmo COA ($N = 12$) com inicialização de população	81
3.27	Resumo dos algoritmos ao nível do erro ($N = 9$ sensores)	83
3.28	Resumo dos algoritmos ao nível do erro ($N = 12$ sensores)	83
4.1	Solução do problema de localização. (a) Situação ideal sem ruído nas medições (b) Situação real	86
4.2	Ponto de interesse perante as duas interseções entre a reta e a circunferência.	88
4.3	Condições de simulação ($N = 9$ sensores)	90
4.4	Ponto de partida para a geração da Cadeia de <i>Markov</i> . (a) Raios obtidos pela estimação da distância (b) Espaço convexo obtido por aplicação do Algoritmo (3.1)	91
4.5	Pontos gerados da sequência de Metropolis-Hastings. (a) $\sigma_M^2 = 0.01$, $T_{Accept} = 85.40\%$ (b) $\sigma_M^2 = 0.1$, $T_{Accept} = 73.10\%$ (c) $\sigma_M^2 = 0.20$, $T_{Accept} = 56.90\%$ (d) $\sigma_M^2 = 0.4$, $T_{Accept} = 38.60\%$ (e) $\sigma_M^2 = 1.00$, $T_{Accept} = 29.80\%$ (f) $\sigma_M^2 = 10$, $T_{Accept} = 0.70\%$	92
4.6	Varição da taxa de aceitação com σ_M^2	93
4.7	iEHO com inicialização Metropolis-Hastings, $N = 9$ sensores	96
4.8	Número de avaliações (MCMC+iEHO), $N = 9$ sensores	97
A.1	Circunferências secantes (Proposição 3.1)	106
A.2	Circunferências secantes (Proposição 3.2)	108
B.1	Histogramas e dados estatísticos para $\sigma_\nu^2 = -40$ dB. (a) $N = 9$ sensores (b) $N = 12$ sensores	111
B.2	Histogramas e dados estatísticos para $\sigma_\nu^2 = -35$ dB. (a) $N = 9$ sensores (b) $N = 12$ sensores	112
B.3	Histogramas e dados estatísticos para $\sigma_\nu^2 = -30$ dB. (a) $N = 9$ sensores (b) $N = 12$ sensores	112
B.4	Histogramas e dados estatísticos para $\sigma_\nu^2 = -25$ dB. (a) $N = 9$ sensores (b) $N = 12$ sensores	113
B.5	Histogramas e dados estatísticos para $\sigma_\nu^2 = -20$ dB. (a) $N = 9$ sensores (b) $N = 12$ sensores	113

B.6	Histogramas e dados estatísticos para $\sigma_v^2 = -15 \text{ dB}$. (a) $N = 9$ sensores (b) $N = 12$ sensores	114
B.7	Histogramas e dados estatísticos para $\sigma_v^2 = -10 \text{ dB}$. (a) $N = 9$ sensores (b) $N = 12$ sensores	114
B.8	Histogramas e dados estatísticos para $\sigma_v^2 = -5 \text{ dB}$. (a) $N = 9$ sensores (b) $N = 12$ sensores	115
C.1	Histogramas e dados estatísticos para $\sigma_v^2 = -40 \text{ dB}$. (a) $N = 6$ sensores (b) $N = 15$ sensores	117
C.2	Histogramas e dados estatísticos para $\sigma_v^2 = -35 \text{ dB}$. (a) $N = 6$ sensores (b) $N = 15$ sensores	118
C.3	Histogramas e dados estatísticos para $\sigma_v^2 = -30 \text{ dB}$. (a) $N = 6$ sensores (b) $N = 15$ sensores	118
C.4	Histogramas e dados estatísticos para $\sigma_v^2 = -25 \text{ dB}$. (a) $N = 6$ sensores (b) $N = 15$ sensores	119
C.5	Histogramas e dados estatísticos para $\sigma_v^2 = -20 \text{ dB}$. (a) $N = 6$ sensores (b) $N = 15$ sensores	119
C.6	Histogramas e dados estatísticos para $\sigma_v^2 = -15 \text{ dB}$. (a) $N = 6$ sensores (b) $N = 15$ sensores	120
C.7	Histogramas e dados estatísticos para $\sigma_v^2 = -10 \text{ dB}$. (a) $N = 6$ sensores (b) $N = 15$ sensores	120
C.8	Histogramas e dados estatísticos para $\sigma_v^2 = -5 \text{ dB}$. (a) $N = 6$ sensores (b) $N = 15$ sensores	121

Lista de Tabelas

2.1	Algoritmos baseados em enxames de partículas	11
2.2	Parâmetros dos algoritmos de otimização simulados	21
2.3	Tempo de computação dos algoritmos implementados	22
2.4	Condições de simulação para validação de resultados com população constante.	28
2.5	Resumo dos algoritmos analisados	30
2.6	Resultados de simulação para $N = 3, 6, 9$ e 12 sensores e σ_v^2 (dB) = $-80 : 5 : -45$	35
2.7	Resultado da aplicação do método EHO	46
3.1	Indicadores estatísticos ($\sigma_v^2 = -40$ dB)	70
3.2	Indicadores estatísticos ($\sigma_v^2 = -35$ dB)	70
3.3	Indicadores estatísticos ($\sigma_v^2 = -30$ dB)	70
3.4	Indicadores estatísticos ($\sigma_v^2 = -25$ dB)	70
3.5	Indicadores estatísticos ($\sigma_v^2 = -20$ dB)	71
3.6	Indicadores estatísticos ($\sigma_v^2 = -15$ dB)	71
3.7	Indicadores estatísticos ($\sigma_v^2 = -10$ dB)	71
3.8	Indicadores estatísticos ($\sigma_v^2 = -5$ dB)	71
4.1	Variação da taxa de aceitação com σ_M^2	93

Lista de Abreviações

ABC	Artificial Bee Colony
ADC	Analog to Digital Converter
AFWA	Adaptive Fireworks Algorithm
AGC	Armijo-Goldstein Condition
AoA	Angle of Arrival
BLS	Backtracking Line Search
CNG	Chaotic Number Generator
COA	Coyote Optimization Algorithm
CoFFWA	Cooperative Framework for Fireworks Algorithm
CPU	Central Processing Unit
DC	Direct Current
dynFWA	Dynamic Search in Fireworks Algorithm
dynFWACM	Dynamic Search Fireworks Algorithm with Covariance Mutation
eddynFWA	Exponentially Decreased Dimension NumberStrategy in Dynamic Search Fireworks Algorithm
EEHO	Enhanced Elephant Herding Optimization
EFWA	Enhanced Fireworks Algorithm
EHO	Elephant Herding Optimization
HEHO	Heuristic Elephant Herding Optimization
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FWA	Fireworks Algorithm
FWADM	Fireworks Algorithm with Covariance Mutation
GPS	Global Positioning System
GWO	Grey Wolf Optimizer

IoT	Internet of Things
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
ML	Máxima Verosimilhança
PET	Tomografia por Emissão de Positrões
PDF	Probability Density Function
PGA	Programmable Gain Amplifier
PRNG	Pseudo-Random Number Generator
PSO	Particle Swarm Optimization
RAM	Random Access Memory
RMSE	Root-Mean-Square Error
RSSI	Received Signal Strength Indicator
SDG	Steepest Descent Gradient
SDP	Semidefinite Programming
SI	Sistema Internacional
SNR	Signal-to-Noise Ratio
SOCP	Second-Order Cone Programming
SPL	Sound Pressure Level
TDoA	Time Difference of Arrival
TGA	Tree Growth Algorithm
ToA	Time of Arrival
WDC	Weighted Direct Least-Squares with Correction
WSN	Wireless Sensor Network

1

Introdução

Conteúdos

1.1	Enquadramento	1
1.2	Conceitos Introdutórios	3
1.3	Organização do Documento	5
1.4	Publicações	7

1.1 Enquadramento

Atualmente, o conceito de localização está estreitamente relacionado com a plataforma de GPS (Global Positioning System): um sistema de posicionamento global constituído por um conjunto de satélites que orbitam a Terra, e equipamentos recetores que calculam a sua posição baseada nos sinais de radiofrequência transmitidos pelos referidos satélites. Criado em 1973 pelo Departamento de Defesa dos Estados Unidos para fins militares, este consiste numa constelação de 31 satélites numa órbita de aproximadamente 20000 *km* (dados de 2018 [1]), de modo a que em qualquer ponto da Terra, tenhamos em média 9 satélites visíveis [2].

Tal situação implica que a investigação científica relacionada se tenha concentrado sobretudo ao nível das aplicações e arquiteturas computacionais. Uma das principais áreas de investigação é ao nível da computação sensível ao contexto. Este conceito

inicialmente proposto por Mark Weiser [3], em 1991, tem como objetivo uma total integração do utilizador com a infraestrutura tecnológica que o rodeia, de modo a integrar as tecnologias de informação com as ações e comportamentos das pessoas. Os computadores seriam, assim, sistemas inteligentes permanentemente conectados, tornando-se omnipresentes.

A georreferenciação do utilizador é então uma componente fundamental deste tipo de aplicações [4, 5]. Um dos principais objetivos consiste em desencadear um conjunto de ações, dependendo do contexto do utilizador, ou seja, da sua localização geográfica num determinado espaço. Também neste contexto, os equipamentos móveis de última geração vieram permitir o desenvolvimento de todo um conjunto de novas aplicações, sobretudo devido ao aumento da sua capacidade de processamento com o recurso a plataformas abertas, além da crescente integração de sensores como acelerómetros, giroscópios, câmaras, microfones entre outros [6]. Por outro lado, o desenvolvimento de redes de sensores sem fios e novos conceitos no âmbito da internet das coisas, aliado às limitações dos sistemas de GPS, importam os conceitos de localização para novas plataformas e aplicações [7, 8].

A presente dissertação aborda o problema de localização acústica, usando os métodos meta-heurísticos, mitigando as suas limitações. Pretende criar-se e validar-se metodologias que permitam aumentar a eficiência dos atuais métodos, quer ao nível da sua complexidade, quer ao nível da precisão da posição final determinada, quer em termos no número de sensores necessários.

Os algoritmos determinísticos apresentam-se em geral com resultados satisfatórios em termos de precisão, mas tipicamente com uma complexidade computacional, e consequentemente um tempo de processamento não desprezável. Os algoritmos meta-heurísticos, por seu lado, apresentam-se como algoritmos ideais para a realização de procuras globais. Assumem-se como algoritmos baseados em operadores simples, portanto, de baixa complexidade e baixo tempo de execução, sendo a sua parametrização e critério de paragem sobretudo empíricos.

Do ponto de vista aplicacional, encontramos exemplos na área da robótica [9], redes sensoriais aquáticas [10–12], sistemas de vigilância [13], localização de veículos

[14], detecção de animais [15], entre outros. Por outro lado, também os algoritmos determinísticos são alvo de aplicações práticas [16–18], implementados em contextos complexos, nomeadamente em situações de medição que não tenham linha de vista direta entre o emissor e o receptor [19–22].

1.2 Conceitos Introdutórios

Os algoritmos de localização podem ser classificados como *Range-free* e *Range-Based Algorithms* [23].

Os primeiros apenas usam informação acerca da conectividade entre uma estação móvel ou nó, acerca da qual se pretende determinar a sua localização, e um conjunto de estações de base ou âncoras, fixas no espaço, sobre as quais se conhece as suas coordenadas. Esta topologia de algoritmos apresenta a vantagem de as estações base e as estações móveis não necessitarem de realizar processamento complexo. Os métodos de cálculo são determinísticos e baseados em modelos geométricos, pelo que a arquitetura do hardware envolvido pode ser bastante simples. Normalmente apenas é necessário um emissor de rádio que emite sinais de curta duração com a sua identificação. Deve apenas salvaguardar-se o desfasamento entre as transmissões das várias estações móveis, evitando-se colisões dos pacotes transmitidos [24–27].

No segundo caso, os algoritmos dependem do pressuposto de que é realizada uma medição de distância absoluta entre um emissor e um receptor, normalmente baseada em medição de potência [28, 29], tempo [30–32], ou ângulo de transmissão/receção [33, 34]. No sentido de abranger as vantagens de diferentes grandezas de medição, são ainda consideradas técnicas que agregam as diferentes variáveis medidas [35–39]. Estes podem ainda ser classificados como cooperativos, em que é utilizada informação de distância entre as estações móveis, e não cooperativos [40, 41], onde apenas se considera a distância entre estações móveis e estações de base [42–45].

Do ponto de vista de computação, os mesmos podem ser classificados como centralizados, distribuídos ou sequenciais.

O processamento centralizado caracteriza-se por concentrar a informação e o cálculo num único processador central, recolhendo todos os dados individuais da

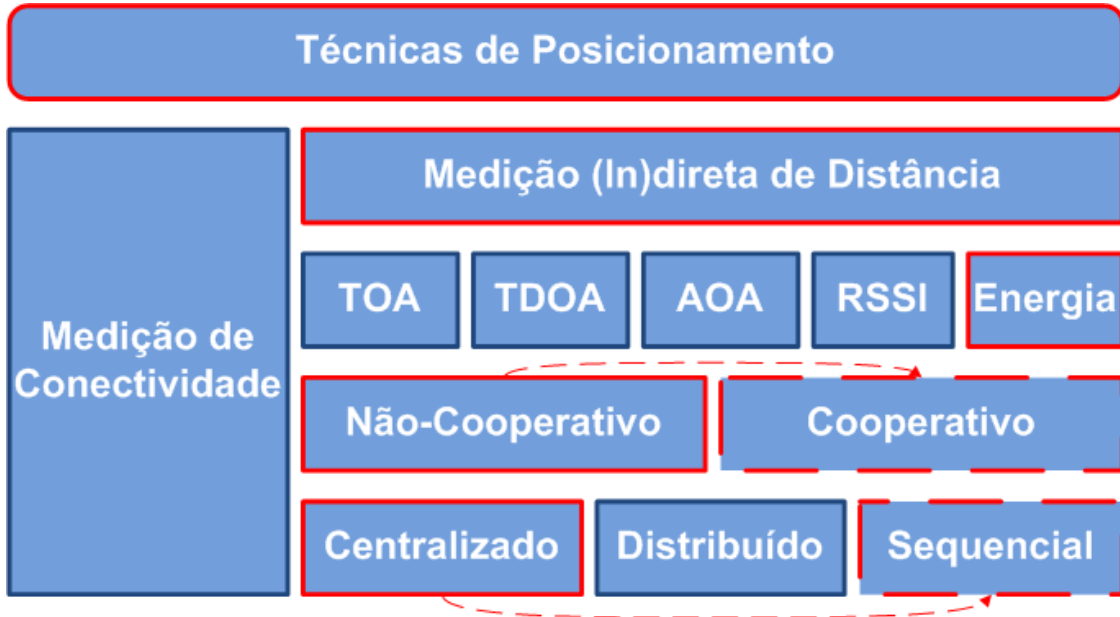


Figura 1.1: Taxonomia das tecnologias de localização

grandeza sob medição entre sensores, produzindo um mapa de toda a rede. O resultado é obtido em simultâneo para toda a topologia da rede [46, 47].

No caso de computação distribuída, os algoritmos dependem da auto localização de cada nó na rede de sensores, usando as medidas de distâncias que o nó mede entre os seus vizinhos e, portanto, as informações locais que ele consegue alcançar [48–51].

Por último, no que diz respeito à computação sequencial, esta baseia-se no conceito de que após uma posição ser estimada para um ou mais nós da rede, esses nós passem a ser usados como tendo posição conhecida. Perante esta abordagem, são necessárias várias iterações para concluir o processo de localização [52, 53].

A figura 1.1 ilustra um esquema que resume os vários conceitos apresentados, com ênfase nas metodologias abordadas no presente trabalho.

No caso da presente dissertação, será considerada a medição da energia acústica como métrica indireta de distância, num contexto não cooperativo e centralizado. No entanto, como se poderá observar com o decorrer do trabalho, perante os avanços alcançados, os algoritmos desenvolvidos terão as condições necessárias para uma implementação cooperativa e sequencial, devido sobretudo à sua simplicidade de implementação e baixo custo em termos de computação.

1.3 Organização do Documento

A presente dissertação está organizada em 5 capítulos. Apresenta-se seguidamente um resumo dos conteúdos abordados nos vários capítulos, nomeadamente a questão científica em análise, a metodologia de abordagem, e os principais resultados obtidos.

Capítulo 2: Métodos de Otimização Meta-Heurísticos para solução do Problema de Localização de uma Fonte Acústica:

No segundo capítulo correspondente ao trabalho de investigação desenvolvido, são analisados algoritmos baseados em enxames de partículas enquadrados no atual estado da arte. A hipótese sob escrutínio consiste em considerar que, apesar da simplicidade dos métodos envolvidos, e sobretudo das suas características de aleatoriedade, os mesmos possam apresentar resultados ao nível de precisão, superiores ou equivalentes, em relação aos seus oponentes determinísticos.

Através de simulações computacionais, considerando um modelo acústico e um estimador de Máxima Verosimilhança (ML), os vários métodos são comparados, considerando diversas tipologias ao nível do número de sensores e do ruído das observações.

Perante os resultados obtidos, o método de otimização baseado no pastoreio de elefantes ou *Elephant Herding Optimization* (EHO) é escrutinado, tendo em conta uma análise de sensibilidade aos seus parâmetros de sintonização, no contexto do estimador ML e do modelo acústico.

Com a informação obtida da análise de sensibilidade, pretende concluir-se que o algoritmo utilizado demonstra que a nova abordagem supera as soluções existentes em ambientes ruidosos, incentivando melhorias e testes adicionais destes métodos meta-heurísticos.

O capítulo termina com uma implementação do algoritmo com sinais obtidos em ambiente real para validação das simulações.

Capítulo 3: Desenvolvimento de Metodologias para o Aperfeiçoamento de Algoritmos Meta-heurísticos.

Perante os resultados obtidos no capítulo 2, formula-se a hipótese de, perante características intrínsecas ao problema de localização acústica, ser possível desenvolver metodologias que acelerem a convergência dos métodos, mantendo ou melhorando a precisão do erro de estimação.

O presente capítulo propõe uma nova abordagem para aplicação ao problema de localização acústica. A metodologia proposta representa uma versão melhorada do método de EHO (extensível a outros métodos baseados em enxames de partículas), na qual são introduzidas duas contribuições. Em primeiro lugar, em vez da inicialização aleatória da população de elefantes, exploram-se as particularidades do problema em questão para desenvolver um esquema de inicialização inteligente. Mais precisamente, as estimativas de distância obtidas em cada sensor são usadas para determinar as regiões nas quais uma fonte mais provavelmente poderá estar localizada. Em segundo lugar, em vez de permitir que os elefantes evoluam a sua posição pela atualização direta do método, é proposto uma procura local baseada na implementação de um método de gradiente discreto.

Pretende-se que os resultados de simulação demonstrem que a metodologia implementada acelera a convergência do algoritmo e tem um custo computacional muito baixo, pois a discretização permite evitar os cálculos reais do gradiente. Assim, espera-se que a nova metodologia supere o método EHO original em situações de baixo ruído, mantendo-se o seu desempenho para ruído elevado ao nível de precisão da localização. Além disso, pretende demonstrar-se que a versão proposta requer um número menor de iterações para convergir.

Para uma validação mais abrangente e demonstração da extensão a outros algoritmos meta-heurísticos, são implementados outros métodos para simulação e análise das metodologias desenvolvidas.

Capítulo 4: Inicialização Inteligente da População pelo Método de Monte Carlo, baseado em cadeias de Markov.

O presente capítulo pretende dar sequência ao método proposto anteriormente para inicialização de uma população candidata à aplicação de um método meta-heurístico. Num segundo estágio, o algoritmo Metropolis-Hastings é considerado para gerar uma sequência de *Monte Carlo* na cadeia de *Markov*. Com esses dois procedimentos, a região de interesse e a sequência de *Monte Carlo* são correlacionadas com o modelo de localização, evitando a geração populacional aleatória. Os resultados das simulações pretendem validar a eficácia da técnica proposta, que combina com o desempenho do estado da arte com um número de avaliações da função de custo reduzido, portanto, com uma taxa de convergência mais rápida.

Capítulo 5: Conclusões e Trabalho Futuro.

Este capítulo conclui a dissertação, sumariando os principais resultados obtidos e listando futuras linhas de investigação com o potencial de desenvolvimento, gerado a partir do presente trabalho.

1.4 Publicações

Os resultados do trabalho de investigação desenvolvidos foram objeto de publicações científicas em revistas da especialidade, assim como, apresentações em conferências da área. Apresenta-se, de seguida, uma listagem dos vários trabalhos de disseminação científica, onde são providenciadas hiperligações para as principais métricas de avaliação:

- a) S. D. Correia, M. Beko, L. A. Da Silva Cruz, S. Tomic, "***Elephant Herding Optimization for Energy-Based Localization***", *Sensors*, vol. 18, no. 2849, 2018.
DOI: 10.3390/s18092849

Scopus → CiteScore 3.72, Rank #102/661, Percentile 84th

Scimago → Q2 on Electrical and Electronic Engineering, SJR (2018) 0.59

WoS → Fator de Impacto (2018) 3.031, Fator de Impacto (5 Anos) 3.302

- b) S. D. Correia, M. Beko, L. A. da Silva Cruz, S. Tomic, "**Implementation and Validation of Elephant Herding Optimization Algorithm for Acoustic Localization**", 26th Telecommunications Forum (TELFOR), Belgrade, 2018, pp. 1–4.

DOI: 10.1109/TELFOR.2018.8611919

Proceedings Indexed by [IEEE](#)

- c) S. D. Correia, M. Beko, S. Tomic, L. A. Da Silva Cruz, "**Energy-based Localization by Enhanced Elephant Herding Optimization**", IEEE Access, vol. 8, pp. 28548–28559, 2020.

DOI: 10.1109/ACCESS.2020.2971787

Scopus → CiteScore 4.96, Rank #10/206, Percentile 95th

Scimago → Q1 on Computer Science, SJR (2018) 0.61

WoS → Fator de Impacto (2018) 4.098, Fator de Impacto (5 Anos) 4.540

Questões sob Investigação: Os algoritmos baseados em enxames de partículas têm sido amplamente apresentados na literatura científica, modelando o comportamento das mais variadas espécies de animais. De entre os vários algoritmos encontrados no atual estado da arte, qual, ou quais, se podem tornar mais promissores para aplicação ao problema da localização, baseado na medição de energia acústica? Qual a influência dos seus parâmetros no resultado final do problema? Qual o seu comportamento em ambiente real?

2

Métodos de Otimização Meta-Heurísticos para Solução do Problema de Localização de uma Fonte Acústica

Conteúdos

2.1	Motivação e Objetivos	9
2.2	Enquadramento do Problema	11
2.3	Formulação do Problema	14
2.4	Algoritmos de Otimização Meta-Heurísticos	17
2.4.1	Resultados Preliminares	18
2.4.2	Algoritmo <i>Elephant Herding Optimization (EHO)</i>	22
2.4.3	Sintonização dos Parâmetros do Método EHO	24
2.5	Análise de Resultados	29
2.5.1	Análise de Complexidade	29
2.5.2	Resultados de Simulação	31
2.6	Validação Experimental	33
2.6.1	Simulação em Ambiente Real	33
2.6.2	Resultados Experimentais	36
2.7	Conclusões	46

2.1 Motivação e Objetivos

O presente capítulo aborda o problema de localização de uma fonte acústica com base na medição da sua energia, enquadrada numa rede de sensores sem fios. Em vez

de contornar o problema da máxima verosimilhança (ML), aplicando relaxações ou aproximações convexas, este é abordado diretamente pelo uso de meta-heurísticas.

No presente contexto científico, tanto quanto se pode verificar pela análise do atual estado da arte, é a primeira vez que métodos meta-heurísticos são aplicados a esse tipo de problema. Mais especificamente, o algoritmo de otimização de pastoreio de elefantes ou, *Elephant Herding Optimization* (EHO), é analisado, estudado e sintonizado, para abordagem ao problema em estudo.

No que diz respeito aos métodos de otimização meta-heurísticos, são muitas e variadas as opções encontradas na literatura. De facto, o tema é abordado pela comunidade científica, de tal modo que são propostos modelos matemáticos que representam o comportamento dos mais variados seres vivos que vivem em comunidade. O primeiro algoritmo proposto com o objetivo de modelizar o comportamento de seres vivos é atribuído a Kennedy e Eberhart [54]. Este, intitulado *Particle Swarm Optimization* (PSO), representa um movimento organizado no contexto de um bando de pássaros. Recentemente, Bonyadi e Michalewicz [55] publicaram uma revisão abrangente sobre os trabalhos teóricos e experimentais acerca do algoritmo, nomeadamente versões que propõem melhorias e adaptações a problemas específicos. Outra abordagem consiste na junção de dois algoritmos, de modo a procurar as melhores características de cada um. Uma justificação usual consiste em utilizar um algoritmo para uma procura global, utilizando outro para um refinamento da solução obtida pelo primeiro. Uma visão global deste tipo de métodos pode ser consultada em [56]. Note-se, no entanto, que estes métodos tipicamente apelados como híbridos consistem na implementação de dois algoritmos, com o consequente custo computacional.

Por outro lado, os métodos continuam a surgir na literatura científica. Já no decorrer da escrita da presente dissertação, foram publicados algoritmos, nomeadamente baseados no comportamento de esquilos voadores e do seu modo de locomoção [57], no comportamento de peixes com particularidades biológicas [58], no comportamento de traças [59–61], ou borboletas [62–64].

Do ponto de vista de originalidade, muitos dos algoritmos propostos têm características comuns do ponto de vista dos modelos matemáticos adotados, o que torna complexo a análise do estado da arte e uma avaliação completa e rigorosa dos mesmos. Na tabela 2.1 apresenta-se um levantamento de alguns dos métodos presentes na literatura científica. O trabalho apresentado em [65] pode ser considerado como uma das mais completas e atuais revisões no que diz respeito a métodos de otimização baseados em enxames de partículas.

Tabela 2.1: Algoritmos baseados em enxames de partículas

Algoritmo	Autor	Referência
Ant colony optimization	Dorigo	[66]
Artificial bee colony	Karaboga and Basturk	[67]
Bat algorithm	Yang	[68]
Bee colony optimization	Teodorivić and Dell’Orco	[69]
Wolf search	Tang et al.	[70]
Cat swarm	Chu et al.	[71]
Cuckoo search	Yang and Deb	[72]
Firefly algorithm	Yang	[73]
Monkey search	Mucherino and Seref	[74]

Perante o cenário traçado, considera-se pertinente e de relevância científica, uma abordagem à aplicação desta tipologia de algoritmos, ao problema concreto da localização de uma fonte acústica. A principal característica considerada para justificar a pertinência de tal aplicação reside sobretudo na simplicidade de implementação dos métodos em termos computacionais. Assim, considera-se que o sucesso da aplicação destes últimos ao problema em questão implicará soluções com baixos tempos de computação e de rápida convergência, características fundamentais no atual contexto das ciências da computação. A título de exemplo, a computação no âmbito do conceito da internet das coisas (do inglês *Internet of Things*), ou *IoT*, será um forte candidato à integração dos métodos desenvolvidos no presente trabalho.

2.2 Enquadramento do Problema

O problema de localização baseada na medição da energia acústica de um sinal tem assumido bastante protagonismo na comunidade científica, sobretudo ao nível da

temática do processamento digital de sinal [75–81]. Esta abordagem considera a média das amostras de energia dos sinais acústicos adquiridos [82]. A localização de fontes acústicas baseada em medição de energia, quando considerada para alvos como objetos em movimento, tem a propriedade de variar lentamente com o tempo. Assim, o sinal de energia acústica pode ser amostrado a uma taxa muito menor. Como consequência, o consumo de energia para transmissão de dados em nós de sensores individuais será reduzido, e a necessidade de largura de banda de comunicação através de comunicações sem fio também será menor [83].

A implementação de um modelo de energia de um sinal acústico, transmitido através de uma rede sem fios, ou *Wireless Sensor Network* (WSN), com uma ou mais fontes implica um problema de otimização não convexo. Para lidar com a não-convexidade, vários métodos foram propostos na literatura.

Em [78] foi apresentado o método dos mínimos quadrados diretos, ponderados e com correção (WDC). Esse método é sujeito a uma técnica de correção que implica ganhos adicionais de desempenho, sendo no entanto, deteriorado em ambientes com elevado ruído, pois os termos de ruído de segunda ordem são ignorados.

Wang e Yang mostraram em [81] que o problema não convexo pode ser aproximado através de programação semidefinida convexa (SDP). Da mesma forma, Beko mostrou em [76] que o problema originalmente não convexo pode ser solucionado através da aplicação de relaxações de programação convexa de segunda ordem ou *Second-Order Cone Programming* (SOCP). Embora os métodos [81] e [76] tenham bom desempenho mesmo em ambientes ruidosos, a sua principal desvantagem consiste na elevada complexidade computacional, que aumenta significativamente com o tamanho da rede.

Os algoritmos anteriormente mencionados são baseados na aplicação de determinadas aproximações ao problema original, causando discrepâncias entre a solução estimada e a solução real. Essas disparidades podem ser elevadas no caso em que as aproximações aplicadas não sejam suficientemente rigorosas, resultando em elevados erros da estimativa em relação à solução real. Para contornar esse problema, o presente trabalho propõe uma abordagem baseada numa metodologia oposta.

Em vez de aproximar o problema de localização original, este é abordado diretamente e solucionado com recurso a métodos meta-heurísticos, nomeadamente o método de otimização EHO.

Este método foi proposto inicialmente por Wang et. al [84, 85] aplicado a várias funções tipificadas no contexto de otimização. Essencialmente, este é um método de pesquisa meta-heurística, baseado em enxames de partículas, para resolver problemas de otimização. O algoritmo simula o comportamento de elefantes em grupo no seu ambiente natural, tendo em conta que, na natureza, os elefantes pertencentes a diferentes clãs vivem juntos sob a liderança de uma matriarca, e os elefantes machos deixam seu grupo familiar ao atingir a adolescência.

O algoritmo de EHO foi aplicado a diversas funções de teste no contexto de várias tipologias de problemas de otimização [86–88], assim como a problemas concretos de engenharia, revelando resultados promissores na busca de soluções ótimas ou sub-ótimas [16, 89–101]. O algoritmo assumiu alguma popularidade devido à sua simplicidade de implementação e, portanto, ao nível da rapidez de computação, quando comparado com outros algoritmos baseados em enxames de partículas. No entanto, o mesmo não tem aplicações conhecidas no contexto da localização em redes de sensores, nomeadamente, ao nível do problema de localização acústica. Portanto, no presente trabalho, o algoritmo de EHO é ajustado e aplicado à localização de fontes acústicas através da medição da energia. No presente, o algoritmo continua a despertar interesse na literatura científica, continuando a surgir publicações que abordam a sua aplicação ou melhoria [102–106].

Preservando-se a estrutura proposta do algoritmo procura-se, numa primeira fase, o ajuste ideal de parâmetros por meio de simulações extensivas, de modo a alcançar taxas de convergência mais eficientes, juntamente com soluções de menor erro, perante várias gamas de ruído das mesmas. Como o algoritmo de EHO não recorre a nenhum tipo de simplificações mas resolve o problema de localização original diretamente, o seu desempenho é menos vulnerável ao ruído. Assim, coloca-se como hipótese que o algoritmo supere outros métodos em ambientes de ruído elevado.

Tendo em conta a elevada proliferação deste tipo de algoritmos (meta-heurísticos baseados em enxames de partículas), o mesmo será comparado com outras soluções, considerando-se os algoritmos presentes na literatura e que tenham demonstrado a sua eficácia, tanto em problemas de teste como em aplicações concretas.

2.3 Formulação do Problema

Considera-se uma rede de sensores bidimensional (a extensão para um cenário tridimensional não será considerada por ser de aplicação direta) composta por N sensores e uma fonte acústica. A real posição da fonte (desconhecida) é denotada por \mathbf{X} e a posição (conhecida) dos i sensores por \mathbf{S}_i , onde $i = 1, \dots, N$. O objetivo consiste em determinar a localização desconhecida da fonte, explorando as observações de energia acústica emitida por uma fonte omnidirecional. Para tal, considera-se o modelo de decaimento de um sinal acústico proposto em [82], que tem como fundamento o facto de a energia acústica emitida de modo omnidirecional de uma fonte sonora ser atenuada a uma taxa inversamente proporcional ao quadrado da distância [82, 83, 107]. Em alguns modelos é proposto um taxa de decaimento β_E , com o propósito de modelar fatores externos. No entanto, o mesmo é normalmente assumido no intervalo $\beta_E \in [2, 4]$ [108], 2 em campo aberto e 4 em ambientes interiores. Cada sensor realiza M medições perturbadas com ruído, considerando uma janela de tempo $T = M/f_s$, onde f_s corresponde à frequência de amostragem. Assim, considera-se a assinatura de energia média ao longo da janela de tempo $[t - T/2, t + T/2]$. De acordo com [82, 83, 107], o sinal amostrado no sensor i pode ser representado como

$$z_i(t) = \frac{\sqrt{g_i}\varphi(t - \tau_i)}{\|\mathbf{x} - \mathbf{s}_i\|^{\beta_E/2}} + \omega_i(t), \quad i = 1, \dots, N. \quad (2.1)$$

Na equação 2.1, tratando-se de uma onda mecânica, $\varphi(t)$ corresponde à intensidade acústica medida em relação a uma distância padrão. O parâmetro τ_i corresponde ao atraso de tempo devido à propagação da onda entre a fonte e o sensor i , $\omega_i(t)$ representa o erro de medição, modelado como ruído Gaussiano com média zero e

variância $\sigma_{\omega_i}^2$. A $\sqrt{g_i}$ representa o ganho do sensor i [107]. No presente trabalho será assumido $\beta_E = 2$. O modelo apresentado em 2.1 será predominantemente analisado, implementado, e usado num estimador para aplicação em algoritmos meta-heurísticos, de um ponto de vista numérico.

Para obter as observações de energia em relação ao sensor i , calcula-se a média de M leituras correspondentes às medições do sinal, segundo 2.1, obtendo-se

$$y_i = \frac{1}{M} \sum_{m=0}^{M-1} z_i^2\left(t_s + \frac{m}{f_s}\right), \quad (2.2)$$

onde t_s representa o instante inicial. Deste modo, tendo em consideração as equações 2.1 e 2.2, temos

$$y_i = \frac{\mathbf{g}_i P}{\|\mathbf{x} - \mathbf{s}_i\|^2} + \nu_i, \quad i = 1, \dots, N \quad (2.3)$$

onde P corresponde à potência transmitida e ν_i representa o ruído de medição [109].

De acordo com o Teorema do Limite Central, para M suficientemente elevado ($M \gg 30$), o ruído de medição de energia do sensor i , ν_i obedece aproximadamente a uma distribuição gaussiana, ou seja, $\nu_i \sim \mathcal{N}(\sigma_{\omega}^2, 2\sigma_{\omega}^4/M)$ em que σ_{ω}^2 corresponde ao nível de ruído. Consequentemente, podemos subtrair a média σ_{ω}^2 de 2.3 e assumir $\nu_i \sim \mathcal{N}(0, \sigma_{\nu_i}^2)$, em que $\sigma_{\nu_i}^2 = 2\sigma_{\omega}^4/M$ [77]. Considerando as observações em 2.3, o estimador de máxima verossimilhança (ML) de \mathbf{X} pode ser formulado como [110]:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^N \left(y_i - \frac{\mathbf{g}_i P}{\|\mathbf{x} - \mathbf{s}_i\|^2} \right)^2. \quad (2.4)$$

O estimador proposto em 2.4 é claramente não convexo, com singularidades nas coordenadas dos sensores. Em relação à variância da potência do ruído, considera-se que a mesma se mantém constante entre os vários sensores, pelo que, $\sigma_{\nu_i}^2 = \sigma_{\nu}^2$.

Uma ilustração de uma possível realização de 2.4 é apresentada na figura 2.1b, onde o modelo é simulado em MATLAB®, considerando-se para tal um conjunto de parâmetros numéricos, de modo a interpretar graficamente o seu comportamento. De modo a obter o gráfico de superfície correspondente à expressão 2.4, que passará a ser denominada a função de custo em relação ao problema de minimização, considera-se

a topologia de rede conforme a figura 2.1a. Esta é constituída por uma WSN de 9 sensores e uma fonte, limitada por uma região quadrada de $100\text{ m} \times 100\text{ m}$. Os sensores são distribuídos de modo uniforme em torno do ponto $[0, 0]^T$ com o raio igual a metade do comprimento da região. A fonte, podendo ser considerada num ponto aleatório do espaço, terá como verdadeira posição o ponto $[-28, 28]^T$. Os outros parâmetros do modelo foram definidos como $g_i = 1$ para $(i = 1, \dots, 9)$ e $P = 500$. Foi ainda adicionado às medições ruído branco onde $10\log_{10}(\sigma_v^2) = -25\text{ dB}$.

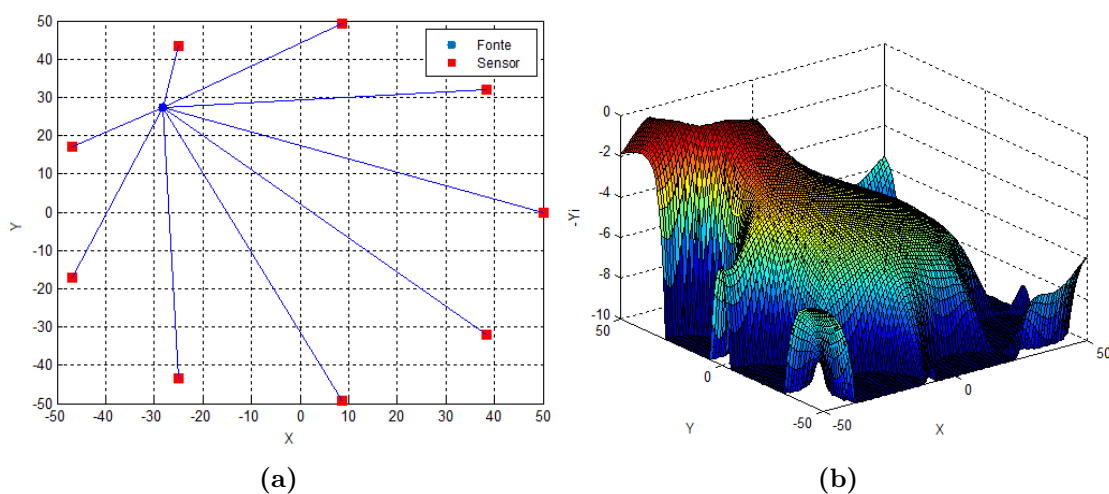


Figura 2.1: Representação gráfica da função de custo do modelo. (a) Configuração de sensores e fonte. (b) Gráfico de superfície do modelo

Com o objetivo de melhorar a visualização e interpretação da função de custo 2.4, a figura 2.1b representa a função simétrica de 2.4. Tal implica que o problema de minimização seja equivalente à procura de um máximo, ao invés de um mínimo da função. Na figura 2.1b pode observar-se que a superfície resultante é não convexa, compreendendo vários máximos locais e zonas de superfície plana. Além disso, observa-se que o máximo global está próximo da posição do ponto real. Como as singularidades implicariam um valor infinito nas coordenadas dos sensores, foi estipulado um valor limite de -10 para a representação gráfica. Devido às características referidas, nomeadamente à não-convexidade de 2.4, encontrar o seu mínimo global é uma tarefa extremamente complexa. Assim, nos capítulos seguintes, desenvolve-se um estimador subótimo que fornecerá uma solução precisa para o problema objeto de estudo, especialmente em ambientes ruidosos.

2.4 Algoritmos de Otimização Meta-Heurísticos

Os algoritmos genéticos podem ser considerados como a primeira metodologia proposta para resolução de problemas de otimização global baseados em meta-heurística. Inspirados no processo de evolução das espécies de Darwin [111], esta classe de métodos (Algoritmos Evolutivos ou *Evolutionary Computation*) aplica um conjunto de operadores inspirados na biologia das espécies, nomeadamente a seleção, recombinação e mutação [112, 113]. Os trabalhos de David E. Goldberg e M. Mitchell atraíram muita atenção na comunidade científica, com 85444 e 14622 citações (Fonte: Google Scholar, consultado em 23/11/2019).

Diversos desenvolvimentos e aplicações foram publicados com base nos algoritmos genéticos. No entanto, outros métodos surgiram, nomeadamente o algoritmo de *Simulated Annealing*, baseado nas propriedades termodinâmicas dos materiais [114–116]. A versão original publicada em 1983 foi citada 45450 vezes (Fonte: Google Scholar, consultado em 23/11/2019), sobretudo evidenciando-se com uma taxa de convergência superior em relação aos algoritmos genéticos [117]. Inspirados nos mesmos algoritmos genéticos, são propostos algoritmos estocásticos, sobretudo com uma abordagem de procura local. Procurando evoluir a partir de uma solução atual, de modo iterativo, baseados numa análise dimensional do problema em estudo, obtendo melhorias ao nível da taxa de convergência para a solução final [118, 119]. Aplicado à otimização de trajetórias, surge em 2000 o algoritmo de Descida Canónica ou *Dynamic Canonical Descent*, onde mais uma vez são conseguidos avanços em termos de convergência dos métodos [120–122]. Em simultâneo, é proposta uma nova classe de algoritmos que, não tendo por base os operadores de seleção, recombinação e mutação, não são classificados como algoritmos evolutivos. Inicialmente proposto por Kennedy e Eberhart em 1995 [54], o algoritmo baseado em enxames de partículas (PSO), é inspirado pelo comportamento social e cooperativo exibido por várias espécies de forma a evoluir de um modo iterativo no espaço de procura. Este é considerado o primeiro passo no conjunto de métodos denominados de *Swarm Computation*. O algoritmo foi alvo de várias propostas de variantes e

aplicações em várias campos da ciência, atingindo 55569 citações (Fonte: Google Scholar, consultado em 23/11/2019).

2.4.1 Resultados Preliminares

O número de algoritmos meta-heurísticos publicados no contexto da computação baseada em enxames de partículas tem sofrido um acréscimo exponencial ao longo dos anos. São propostas, na literatura, novas metodologias, algoritmos baseados em novos fenômenos biológicos, sendo que muitos dos quais, em termos matemáticos, não aportam valor acrescentado (modelos matemáticos muito semelhantes). Deste modo, no sentido de obter um conjunto de algoritmos para análise aplicada ao problema de localização acústica, consideram-se determinados critérios seguidamente descritos.

O primeiro algoritmo analisado será o método original de Kennedy e Eberhart (PSO) [54], sendo um dos métodos mais citado na literatura, quer em termos das variações de melhoria propostas, quer em termos de campos de aplicação.

Em segundo lugar, considera-se um algoritmo meta-heurístico, que não assenta em modelos biológicos, o algoritmo de *Fireworks* (FWA) [123]. Este foi alvo de várias propostas de melhoria, nomeadamente *Enhanced Fireworks Algorithm* (EFWA) [124], *Dynamic Search in Fireworks Algorithm* (dynFWA) [125], *Adaptive Fireworks Algorithm* (AFWA) [126], *Exponentially Decreased Dimension Number Strategy in Dynamic Search Fireworks Algorithm* (eddynFWA) [127], *Dynamic Search Fireworks Algorithm with Covariance Mutation* (dynFWACM) [128], *Fireworks Algorithm with Covariance Mutation* (FWADM) [129] e *Cooperative Framework for Fireworks Algorithm* (CoFFWA) [130]. Dadas as suas diversas variantes, também foi amplamente sujeito a aplicações em vários campos da engenharia, nomeadamente ao nível das WSNs [131–137].

O terceiro algoritmo sujeito a escrutínio será o "Artificial Bee Colony" (ABC), um algoritmo de otimização baseado no comportamento inteligente de enxames de abelhas, proposto por Derviş Karaboğa em 2005 [67]. O modelo destaca-se por considerar uma colónia com grupos de abelhas (partículas): empregadas, observadoras e batedoras. Cada uma apresenta um comportamento diferente

no modelo, recriando os conceitos de procura local e global. O algoritmo será abordado na sua forma original, embora o mesmo possua variações e implementações híbridas [138–142].

Por fim, considera-se um algoritmo mais recente (do ponto de vista da elaboração do presente trabalho), *Elephant Herding Optimization* [84, 85], mas com aplicações propostas em vários campos da engenharia. O algoritmo EHO foi testado em várias funções de teste, obtendo resultados promissores [84, 143]. Além disso, também foi considerado em aplicações como o controle proporcional–integral–derivativo (PID) [89], a qualidade da composição do serviço da web [90], entre outros [92]. Recentemente propostos, podemos ainda encontrar o *Squirrel Search Algorithm* [57], ou o *Selfish Herd Optimizer* [58, 144], como alguns dos algoritmos mais publicados em revistas de 1º ou 2º quartil. Os algoritmos selecionados foram implementados na plataforma MATLAB®R2009b. Todas as simulações foram realizadas numa arquitetura com os seguintes componentes: processador Intel®Core®I7-4700HQ CPU, 2.4GHz, 16 GB de RAM e Windows®8 (64 bits).

Em relação aos parâmetros de simulação dos algoritmos (resumidos na tabela 2.2, onde N_{eval} representa o número de avaliações), são tidos em conta os valores propostos nos seus artigos de origem, nomeadamente o algoritmo EHO ajustado com os parâmetros $\alpha = 0.5$ e $\beta = 0.1$. Considera-se ainda $c_i = 5$ clãs com $n_{ci} = 10$ elefantes cada. Foi utilizado um critério de paragem, dependente do número de avaliações da função de custo, de 2500 avaliações. Tal implica um máximo de 50 gerações, tendo em conta um tamanho populacional de 50 elefantes. O algoritmo PSO é simulado com $n_p = 60$ partículas e um máximo de 10000 iterações. O algoritmo ABC com $n = 20$ abelhas para o tamanho da colónia, e um máximo de 5000 iterações. O algoritmo de EFWA é parametrizado com um coeficiente de explosão de $A = 40$ e coeficiente de faíscas $m = 50$, tendo como número máximo e mínimo de faíscas $S_{Max} = 40$ e $S_{min} = 2$, respetivamente, e um número máximo de 10000 avaliações.

As simulações foram realizadas com duas configurações, sendo considerada apenas 1 fonte e $N = 9$ ou $N = 12$ sensores, distribuídos conforme a figura

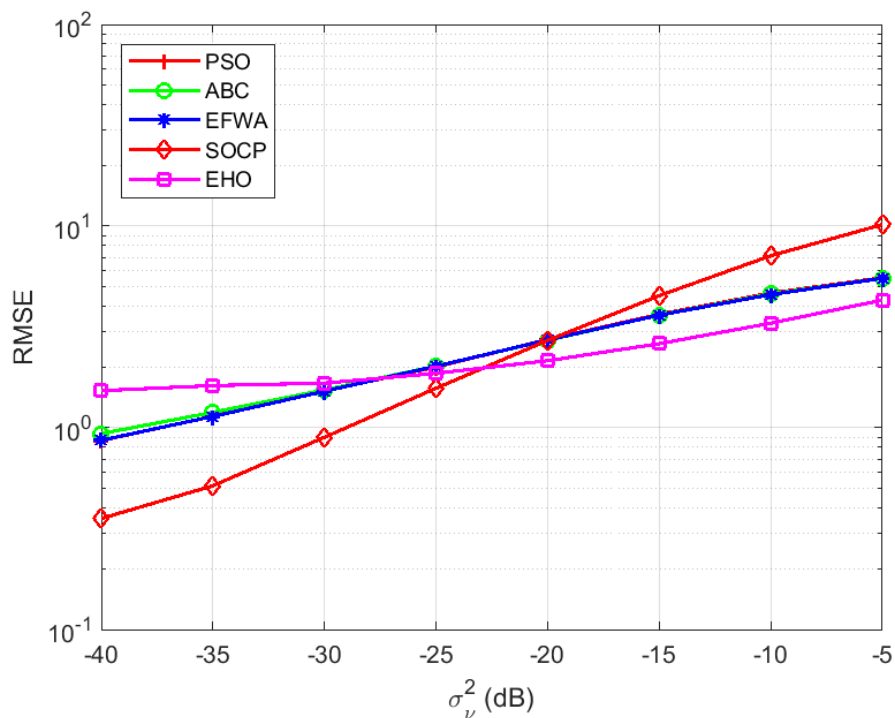


Figura 2.2: Resultados de simulação para PSO, ABC, EFWA, SOCP e EHO, com $N = 9$ sensores

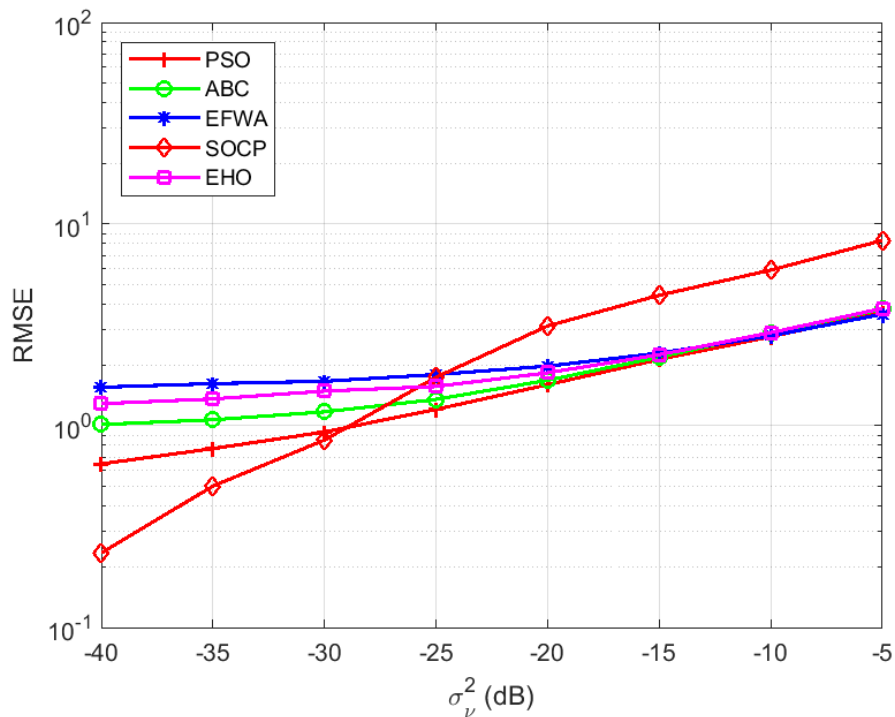


Figura 2.3: Resultados de simulação para PSO, ABC, EFWA, SOCP e EHO, com $N = 12$ sensores

Tabela 2.2: Parâmetros dos algoritmos de otimização simulados

	EHO	PSO	ABC	EFWA
Parâmetros Específicos	$\alpha = 0.5$ $\beta = 0.1$ $c_i = 5$ $n_{ci} = 10$	$n_p = 60$	$n_a = 20$	A=40 $m = 2$ $S_{min} = 2$ $S_{max} = 40$
N_{eval}	2500	10000	5000	10000

2.1a. Em relação à potência do ruído, considera-se $-40 \text{ dB} < \sigma_\nu^2 < -5 \text{ dB}$, onde, tendo em conta a amplitude do intervalo, a variância é considerada numa escala logarítmica segundo a expressão $10\log_{10}(\sigma_\nu^2)$. Em relação a cada algoritmo são gerados $M_c = 10000$ pontos de *Monte Carlo* aleatoriamente distribuídos no espaço de procura. O algoritmo SOCP é considerado como referência ao nível determinístico, simulado com os parâmetros de [76]. Os resultados obtidos podem ser observados nas figuras 2.2 e 2.3, respetivamente, onde o eixo das ordenadas representa o erro médio quadrático em metros (RMSE) segundo,

$$RMSE = \sqrt{\sum_{i=1}^{M_c} \frac{\|\mathbf{x} - \hat{\mathbf{x}}_i\|^2}{M_c}}, \quad (2.5)$$

onde \mathbf{x} corresponde ao vetor dos pontos reais, desconhecidos para os algoritmos, e $\hat{\mathbf{x}}_i$ o vetor dos M_c pontos estimados.

Como podemos verificar na figura 2.2, para valores reduzidos da potência do ruído σ_ν^2 , o algoritmo SOCP supera os restantes. Quando a potência do ruído atinge os -25 dB , todos os restantes métodos apresentam melhores resultados. O método de EHO apenas melhora em aproximadamente 1 m , em relação ao algoritmo PSO, ABC e EFWA, e 5 m em relação ao algoritmo SOCP.

No caso da figura 2.3, os resultados configuram-se de forma muito semelhantes, embora com uma melhoria de cerca de 1 m para uma potência de ruído de -5 dB , face ao algoritmo de SOCP. Os resultados dos restantes métodos são equivalentes em comparação com PSO, ABC e EFWA. No entanto, a análise de complexidade apresentada em [76] refere-se ao método SOCP como sendo de ordem $O(N^{3.5})$, resultando em valores mais baixos no tempo de computação.

Em relação aos restantes algoritmos, a análise pode ser realizada do ponto de vista do número de iterações.

Note-se que o algoritmo de PSO necessita de quatro vezes mais gerações e avaliações de população, enquanto que o algoritmo de ABC necessita do dobro das gerações, o que implica um tempo de computação superior em relação ao método de EHO, que se apresenta com os valores mais reduzidos. Na tabela 2.3 pode observar-se os tempos médios de computação dos vários algoritmos nas M_c simulações.

Tabela 2.3: Tempo de computação dos algoritmos implementados

Algoritmo	$Tempo_{CPU}(s)$
PSO	11.21
ABC	4.73
SOCP	2.57
EFWA	0.41
EHO	0.23

2.4.2 Algoritmo *Elephant Herding Optimization (EHO)*

O algoritmo de EHO proposto por Wang et. al. [84, 85], conforme foi referido, consiste num método baseado em enxames de partículas [145], sendo um método de busca meta-heurística, e surge da modelação do comportamento de elefantes na natureza. Esse comportamento específico pode ser resumido da seguinte forma: existência de uma população de elefantes dividida em vários subgrupos conhecidos como clãs; cada clã move-se sob a liderança de uma matriarca; os elefantes machos deixam o clã ao qual pertencem e vivem em solidão na natureza.

Em termos do algoritmo, esses comportamentos são descritos com dois operadores: atualização do clã (que atualiza as posições dos elefantes e matriarcas em cada clã) e uma separação (que aprimora a diversidade da população na fase de busca posterior) [84].

Do ponto de vista matemático, o algoritmo EHO é definido da seguinte forma: toda a população de elefantes é inicialmente organizada em k clãs; depois de classificar os elefantes de acordo com sua aptidão (correspondente à avaliação de cada elefante de acordo com 2.4), o operador de atualização do clã é aplicado; cada

membro j do clã i move-se de acordo com a respetiva matriarca, correspondente ao valor mais baixo da avaliação da função de custo:

$$x_{new,c_i,j} = x_{c_i,j} + \alpha(x_{best,c_i} - x_{c_i,j})r. \quad (2.6)$$

Na equação 2.6 $x_{new,c_i,j}$ e $x_{c_i,j}$, correspondem à nova e antiga posição do elefante j no clã i , $\alpha \in [0, 1]$ diz respeito a um parâmetro de ajuste que determina a influência da matriarca i em $x_{new,c_i,j}$, x_{best,c_i} representa o indivíduo mais apto no clã c_i , e $r \sim \mathcal{U}[0, 1]$ [84].

A posição do elefante mais apto do clã é atualizada de acordo com:

$$x_{new,c_i} = \beta x_{center,c_i} \quad (2.7)$$

$$x_{center,c_i,d} = \frac{1}{n_{c_i}} \sum_{j=1}^{n_{c_i}} x_{c_i,j,d} \quad (2.8)$$

onde $\beta \sim \mathcal{U}[0, 1]$ realiza um ajuste estocástico para determinar a influência de x_{center,c_i} em relação a x_{new,c_i} , d é uma referência à dimensão do sistema de coordenadas, em que $1 \leq d \leq D$ onde D corresponde à dimensão do problema (no presente caso, $d = 2$, tendo em conta o caso bidimensional), e n_{c_i} o número de elefantes no clã i [84].

Em relação aos elefantes machos, aos valores mais reduzidos obtidos pela avaliação da função de custo, aplica-se o operador de separação em cada iteração, criando um movimento desses mesmos elefantes para novas posições, e sendo substituídos no respetivo clã i . Para tal, é usada a expressão

$$x_{worst,c_i} = x_{min} + (x_{max} - x_{min} + 1) \psi, \quad (2.9)$$

onde x_{max} e x_{min} são, respetivamente, o limite superior e inferior do espaço de procura, e $\psi \sim \mathcal{U}[0, 1]$ [84].

Deste modo, o algoritmo EHO consiste em aplicar iterativamente as equações 2.6, 2.7, 2.8 e 2.9, durante um número predefinido de iterações. Os parâmetros α , β , c_i , n_{c_i} e o número de avaliações da função custo são considerados fixos para uma determinada simulação. O tamanho da população corresponde ao conjunto formado pelos clãs e o número de elefantes que integram cada um.

2.4.3 Sintonização dos Parâmetros do Método EHO

Nesta secção, o algoritmo EHO é testado em relação à variabilidade dos valores dos seus principais parâmetros, de modo a determinar a sua influência na taxa de convergência e no erro de localização, definido como a diferença entre a posição real da fonte (desconhecida) e o ponto estimado segundo 2.4. Considera-se a mesma configuração apresentada na figura 2.1a, assim como α entre 0.3 e 0.8, com o objetivo de analisar as consequências das sua variação nos resultados finais do algoritmo EHO. Os resultados são apresentados na figura 2.4.

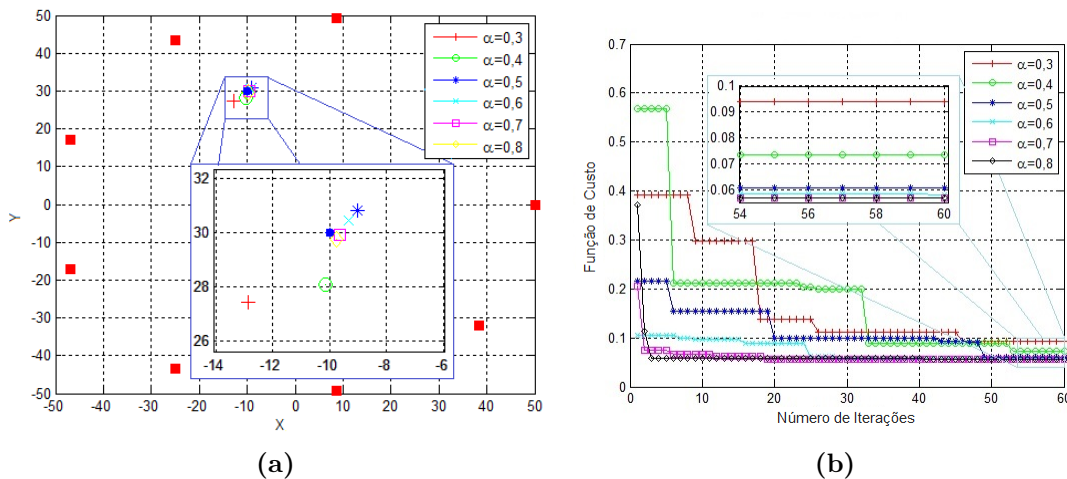


Figura 2.4: Análise de variabilidade do parâmetro α . (a) Representação da solução final no espaço de procura; "•" representa a posição verdadeira. (b) Dependência de α na função de custo.

A figura 2.4a representa as soluções da estratégia proposta para diferentes valores de α , enquanto que a figura 2.4b ilustra a dependência da taxa de convergência com o mesmo α . Na figura 2.4, pode observar-se que as soluções correspondentes a α mais elevados (por exemplo, $\alpha \geq 0.7$) resultam numa maior precisão da estimativa, sugerindo que esses valores sejam preferíveis para o problema em questão. Além disso, a figura 2.4 demonstra que o aumento de α melhora a taxa de convergência.

Como o algoritmo EHO tem um comportamento estocástico, tendo em conta as equações 2.6 e 2.9, um cenário específico pode não ser suficiente para permitir extrapolar os resultados, pelo que, uma abordagem de *Monte Carlo* (M_c simulações)

deve ser considerada. Portanto, $M_c = 10000$ simulações são executadas, e o RMSE é utilizado como a métrica de desempenho, segundo 2.5, definido previamente.

A figura 2.5 ilustra a relação entre o erro $RMSE$ em metros e α , para o caso em que $\sigma_v^2 = -25$ dB e $\beta = 0.1$. Esta evidencia, que à medida que o valor de α aumenta, o erro diminui, o que está de acordo com as previsões expostas anteriormente. Portanto, no algoritmo EHO, α é um parâmetro que determina a influência da matriarca e atua como um fator de escala, onde valores mais elevados de α devem ser predominantemente escolhidos. Tal situação implica uma predominância de $x_{c_i,j}$ na geração de $x_{new,c_i,j}$.

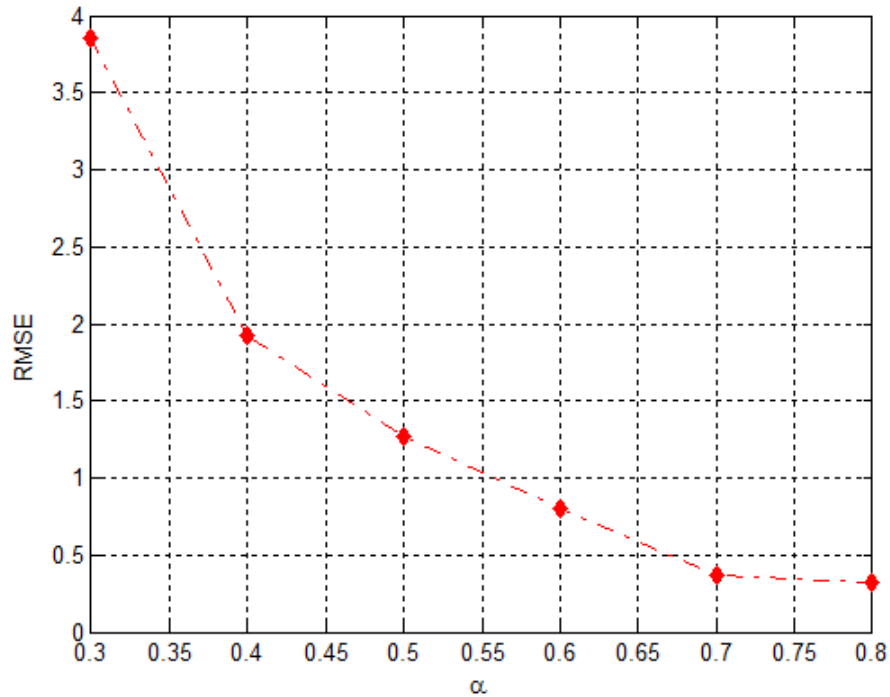


Figura 2.5: RMSE função de α , com $\sigma_v^2 = -25$ dB e $\beta = 0.1$.

O segundo parâmetro objeto de estudo corresponde a β , ou seja, a sua influência no RMSE, sendo ilustrado o seu comportamento na figura 2.6. Deve ser realçado que este parâmetro define a influência de x_{center,c_i} em x_{new,c_i} . Valores reduzidos de β implicam novos pontos longe de x_{center,c_i} , induzindo um alto nível de exploração. Portanto, espera obter-se uma melhor precisão do algoritmo para valores mais baixos

de β , situação observada na figura 2.6. Os resultados apresentados são gerados variando β entre 0.1 e 0.8, nas mesmas condições expostas anteriormente.

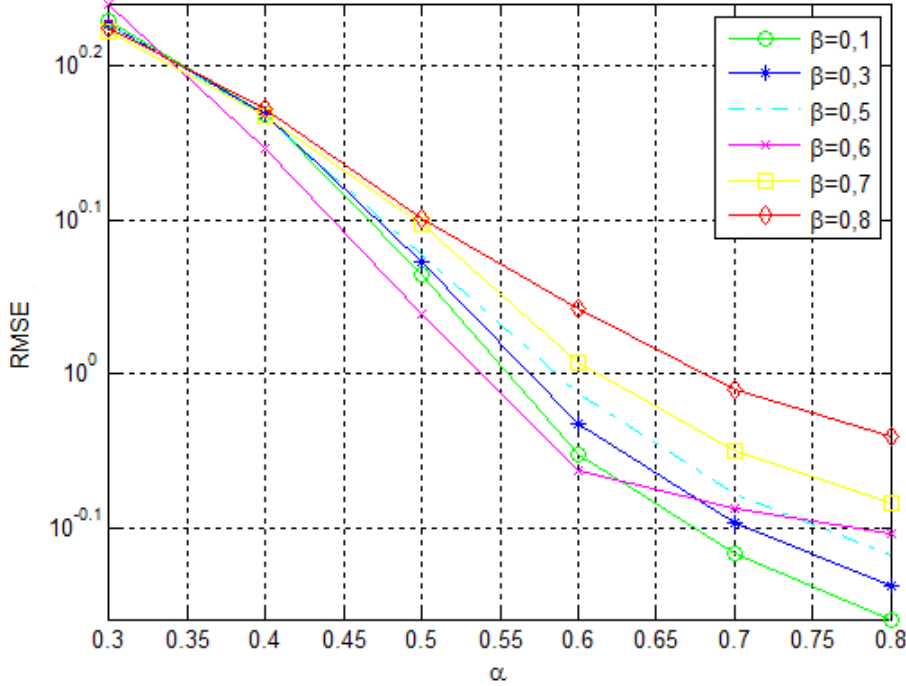


Figura 2.6: RMSE função de α para diferentes valores de β .

Pela análise da figura 2.6, observa-se que, embora não haja uma variação significativa do erro quando α é reduzido (por exemplo, $\alpha \leq 0.4$), no caso de valores mais elevados de α , devem ser usados valores mais baixos de β . No entanto, o efeito de β na precisão da localização é marginal.

O segundo conjunto de parâmetros analisados dizem respeito à população de elefantes e sua organização em clãs. No algoritmo EHO, a população de elefantes é organizada em c_i clãs com n_{c_i} elefantes por clã. Portanto, o tamanho da população (T_{Pop}) é dado por:

$$T_{Pop} = n_{c_i} c_i. \quad (2.10)$$

Assim, considera-se o número de clãs entre 3 e 10 com um incremento de 5 elefantes, preservando as condições de simulação anteriores, em relação ao modelo de energia, layout dos sensores, e número de simulações de *Monte Carlo*. Note-se

que não se está a considerar um tamanho da população constante, pois o mesmo varia entre 15 (3 clãs com 5 elefantes) e 300 (10 clãs com 30 elefantes). A influência desses parâmetros no RMSE é ilustrada na figura 2.7.

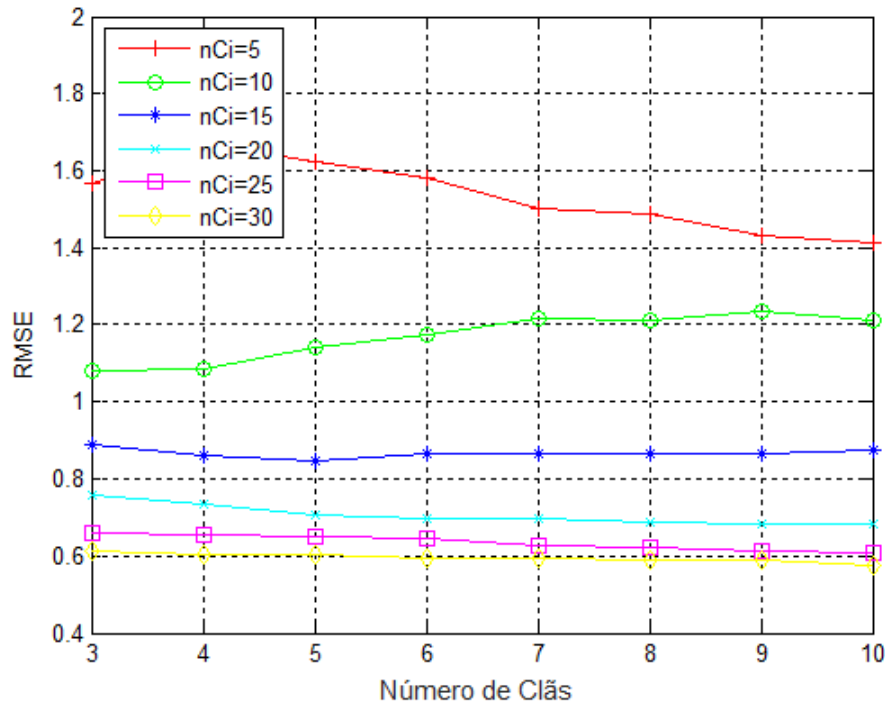


Figura 2.7: Dependência do RMSE com o número de clãs e n_{c_i} .

Como se pode observar na figura 2.7, o aumento do número de elefantes em cada clã produzirá um erro do posicionamento menor. No entanto, para valores mais elevados de n_{c_i} , a diminuição é muito mais lenta em relação a valores mais baixos. Isso ocorre porque, com o aumento de n_{c_i} , está a realçar-se o efeito da exploração das regiões de interesse. Com base na figura 2.7, observa-se que 15 (ou mais) elefantes por clã é praticamente suficiente para que o algoritmo converja, ou seja, adicionar mais clãs não tem um efeito significativo. O modo mais eficaz de reduzir o erro nesse caso consiste em adicionar mais elefantes em cada clã. Deve ainda realçar-se que, para valores reduzidos do número de elefantes, $n_{c_i} = 5$ e $n_{c_i} = 10$, o comportamento do método é contraditório. Tal situação justifica-se devido ao baixo nível de exploração, predominando o comportamento estatístico dos operadores.

Tabela 2.4: Condições de simulação para validação de resultados com população constante.

Pop.	Clã	n_{ci}	Pop.	Clã	n_{ci}	Pop.	Clã	n_{ci}
100	2	50	50	2	25	75	3	25
	4	25		5	10		5	15
	5	20		10	5		15	5
	10	10	25	3				
	20	5						
25	4							

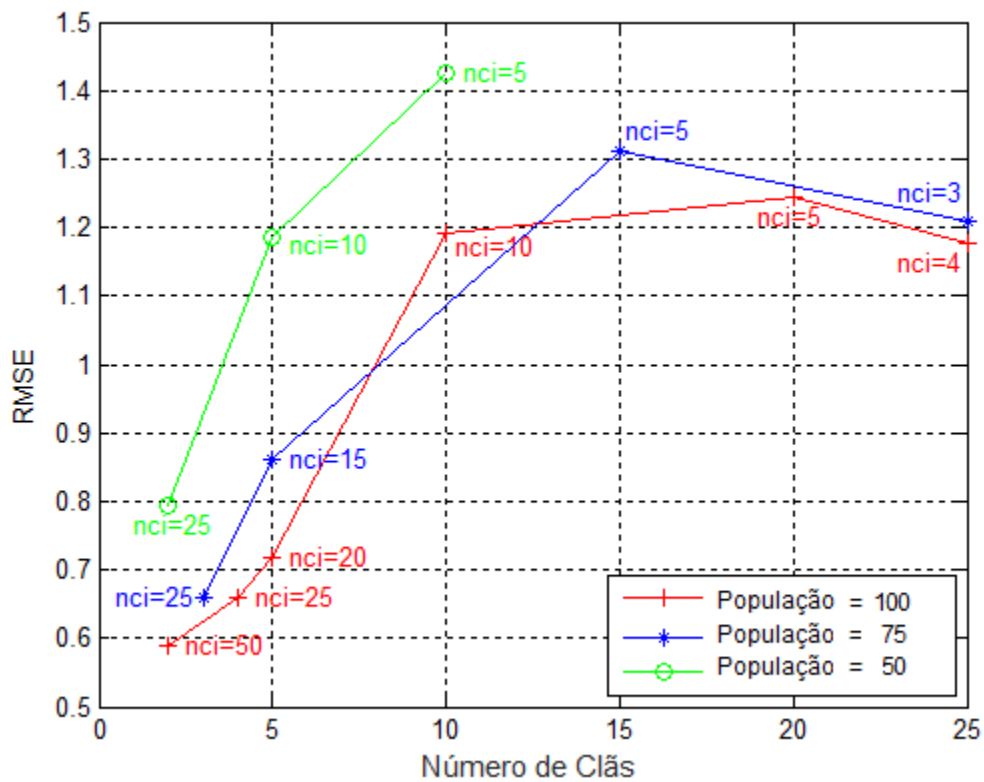


Figura 2.8: Dependência do RMSE com o número de clãs e n_{ci} , mantendo a população de tamanho constante.

Num segundo estágio, considera-se uma população constante de 100, 75 e 50 elefantes, e repetem-se as simulações sob as mesmas condições, mas combinando o número de clãs e n_{c_i} , de modo a manter a população constante. Um resumo da configuração é apresentado na tabela 2.4, e os resultados ilustrados na figura 2.8.

Perante a figura 2.8, pode observar-se que tamanhos de população mais elevados implicam um RMSE mais baixo. As três curvas para diferentes dimensões de população indicam que se deve dar mais importância à exploração das regiões de interesse, através do número de elefantes por clã n_{c_i} , em relação à exploração do espaço de procura (número de clãs). Isso pode ser explicado, em certa medida, pela intuição de que a região de procura seja coberta por um número menor de grupos, compreendendo mais elefantes, portanto com um maior grau de liberdade.

2.5 Análise de Resultados

A presente secção apresenta um conjunto de resultados, de modo a aferir o desempenho do algoritmo de localização considerado. Tanto a complexidade computacional, quanto a precisão das estimativas do algoritmo, são objeto de análise e comparação com algoritmos determinísticos.

2.5.1 Análise de Complexidade

No sentido de comparar os vários métodos abordados, procede-se a uma análise de complexidade. Pretende comparar-se o esforço de computação do método de EHO, em relação aos atuais métodos determinísticos. Para um número máximo predefinido de gerações, o algoritmo EHO realiza quatro operações: classificação de elefantes, atualização, separação, e avaliação da população.

No presente trabalho, a classificação de elefantes foi implementada usando a função de classificação do software MATLAB®, que implementa o algoritmo Quick Sort, de ordem $O(n \log_{10}(n))$. O operador de atualização consiste em dois níveis de operações cíclicas, dependendo do número de clãs e do número de elefantes em cada clã, resultando em $O(n_{Cla}n_{c_i})$. O operador de separação executa operações em cada clã, obtendo-se $O(n_{Cla})$. A avaliação da população diz respeito ao número

total de elefantes. Portanto, obtêm-se para esta operação $O(n_{Cla}n_{c_i})$. Assim, a complexidade de computação do algoritmo de EHO é a junção das quatro operações, para um total do número máximo de gerações (MaxGen):

$$O\left(\text{MaxGen}\left(n_{Cla}n_{c_i}\log(n_{Cla}n_{c_i}) + n_{Cla}n_{c_i} + n_{Cla} + n_{Cla}n_{c_i}\right)\right) \quad (2.11)$$

onde os quatro termos da soma correspondem, respetivamente, às quatro operações do algoritmo.

Consequentemente, a complexidade computacional do algoritmo EHO é de ordem $\text{MaxGen } O(N)$, onde MaxGen corresponde ao número máximo de gerações da população. Tendo em conta que K corresponde ao número máximo de etapas no procedimento de bissecção em [146], a tabela 2.5 fornece uma visão geral dos algoritmos considerados, juntamente com as suas complexidades computacionais. A partir desta, pode observar-se que o algoritmo EHO possui complexidade linear com N , inferior à maioria dos algoritmos considerados. Isso significa que será espectável que o tempo de execução do algoritmo EHO seja menor que o tempo de execução dos algoritmos SDP, SOCP e WDC, o que é uma vantagem na maioria das aplicações práticas.

Tabela 2.5: Resumo dos algoritmos analisados

Algoritmo	Descrição	Complexidade	\overline{T}_{CPU} (s)
SDP	Algoritmo SDP proposto em [81]	$O(N^{4.5})$	3.5
SOCP	Algoritmo SOCP proposto em [76]	$O(N^{3.5})$	2.6
WDC	Algoritmo WDC proposto em [78]	$O(N^2)$	1.74
EHO	Algoritmo Analisado no Cap. 2.4	$\text{MaxGen} \cdot O(N)$	0.23
EXACT	Algoritmo proposto em [146]	$K \cdot O(N)$	0.03

Para verificar empiricamente a análise de complexidade, a tabela 2.5 compara o tempo médio de execução de 10000 execuções em *Monte Carlo* de cada algoritmo implementado. A configuração da rede consiste em 9 sensores, e o modelo acústico foi considerado com $P = 500$ e uma potência de ruído $\sigma_v^2 = -30$ dB, somado às observações y_i . Como esperado pela análise de complexidade apresentada, a tabela 2.5 demonstra que o algoritmo proposto é o segundo mais rápido, com

um tempo médio de execução abaixo de 1 s. A complexidade e o consequente tempo de execução deverá ser analisado em conjunto com os resultados ao nível do erro. Não considerando estratégias de melhoria, objeto de estudo no presente trabalho, empiricamente um algoritmo de maior complexidade resultaria em erros menores e vice-versa.

2.5.2 Resultados de Simulação

Todos os algoritmos considerados no presente capítulo foram implementados recorrendo à aplicação computacional MATLAB®R2009b. As simulações foram realizadas numa plataforma composta por um computador em *cluster* com sete nós, cada um com dois processadores Intel®Xeon®E5520, 24 GB de RAM, executando o Windows®2008 Server HPC. O algoritmo proposto é considerado com os parâmetros

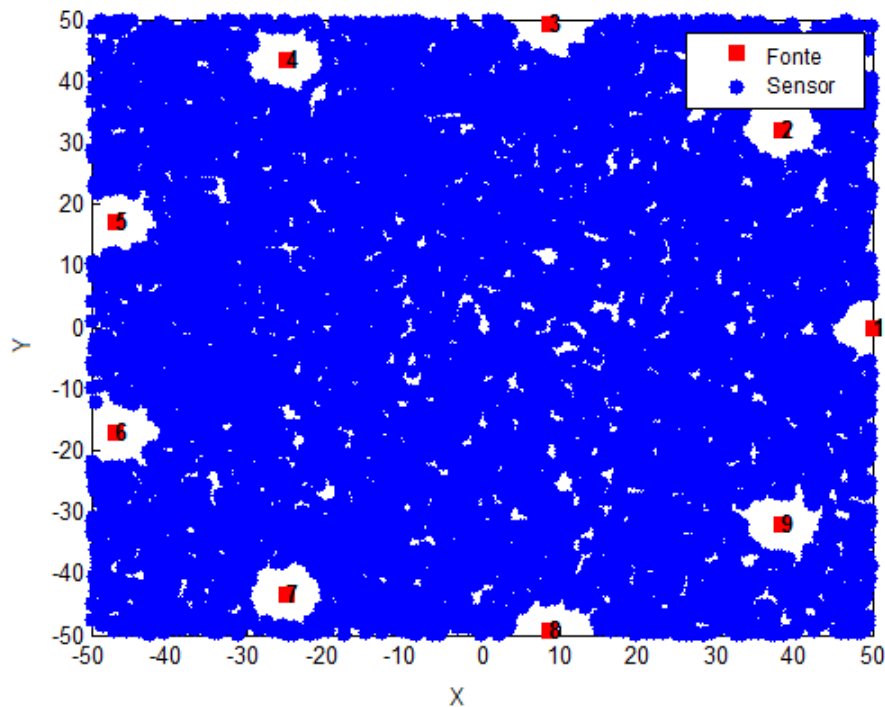


Figura 2.9: Distribuição aleatória de 10000 fontes ($N = 9$)

definidos de acordo com as conclusões estabelecidas no capítulo 2.4.3, ou seja, $\alpha = 0.7$ e $\beta = 0.1$. O número de clãs foi definido como 5 e $n_{c_i} = 20$ elefantes em cada clã. Foi utilizado um critério de paragem de 5000 avaliações da função de custo,

implicando um máximo de 50 gerações (valor de $MaxGen$ considerado na análise de complexidade) para um tamanho da população de 100 elefantes. O desempenho do algoritmo proposto foi comparado com o algoritmo SDP [81], indicado por “SDP”, o algoritmo de bissecção [146], aqui identificado por “EXACT”, o algoritmo “WDC” [78] e, finalmente, o algoritmo “SOCP” [76]. Estes mesmos algoritmos são considerados como o estado da arte em relação aos métodos determinísticos.

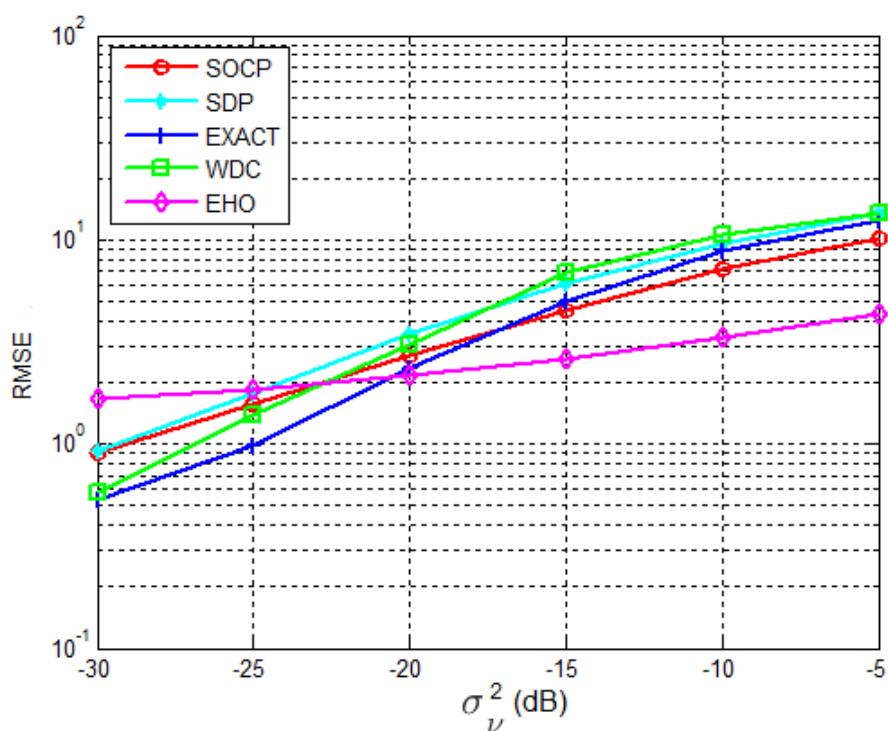


Figura 2.10: RMSE em função de σ_v^2 (dB) para $N = 9$.

A configuração da rede para as simulações consiste em N sensores distribuídos de modo uniforme sobre uma circunferência, cujo raio foi definido em 50 m (onde um mínimo de 3 sensores é necessário, pois considera-se um espaço bidimensional), enquanto a fonte foi posicionada aleatoriamente num espaço com dimensões $100 \times 100\text{ m}^2$ (figura 2.9). Observe-se que em todas as simulações apresentadas, $MC = 10000$ e o RMSE é usado como métrica de desempenho, no sentido de dissipar qualquer efeito da distribuição das fontes no espaço de procura, nomeadamente, fontes localizadas fora do espaço convexo formado pelos sensores. Inicialmente, considera-se $N = 9$ e analisa-se a influência da potência do ruído (σ_v^2). Os resultados

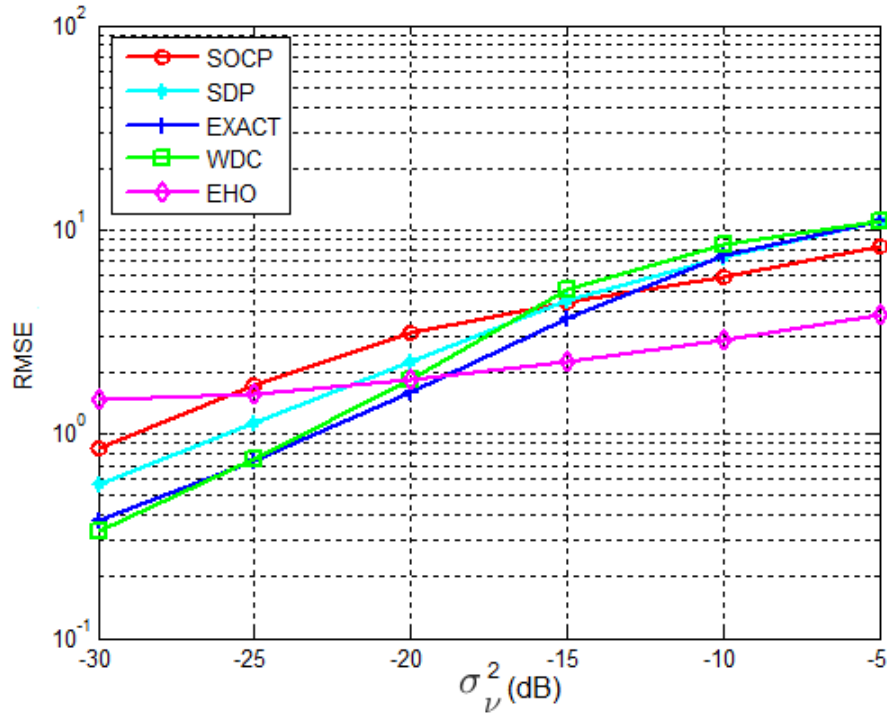


Figura 2.11: RMSE em função de σ_v^2 (dB) para $N = 12$.

são ilustrados na figura 2.10. Posteriormente, considera-se $N = 12$, mantendo as restantes condições, sendo os resultados apresentados na figura 2.11.

2.6 Validação Experimental

No sentido de avaliar o método analisado em ambiente real, com observações acústicas obtidas a partir de medições de campo, procede-se à implementação de uma plataforma de recolha de dados. Tendo em conta as limitações ao nível dos sensores acústicos, sobretudo ao nível do seu ganho, assim como a potência de transmissão da fonte utilizada, foi definido um novo cenário, perante uma alteração em relação às simulações realizadas no presente capítulo, sendo as novas condições novamente simuladas, de modo a obterem-se resultados comparáveis para validação.

2.6.1 Simulação em Ambiente Real

Para simular o ambiente de campo onde o teste é realizado, considera-se um espaço de procura de dimensões $10\text{ m} \times 20\text{ m}$, com 12 sensores distanciados de 5 m , e

uma fonte acústica posicionada fora do espaço convexo formado pelos sensores (figura 2.12a), nas coordenadas $(0\text{ m}, 5\text{ m})$. Para atenuar o efeito do posicionamento da fonte no resultado final, e validar o algoritmo em qualquer ponto do espaço de pesquisa, consideram-se 10000 fontes distribuídas aleatoriamente no espaço (figura 2.12b). A potência de transmissão foi ajustada para $P = 100\text{ dB}$ (isto é, 10 mW , onde 10^{-12} W é usado como potência de referência), e os sensores com um ganho $g_i = 1$ ($i = 1, \dots, 12$).

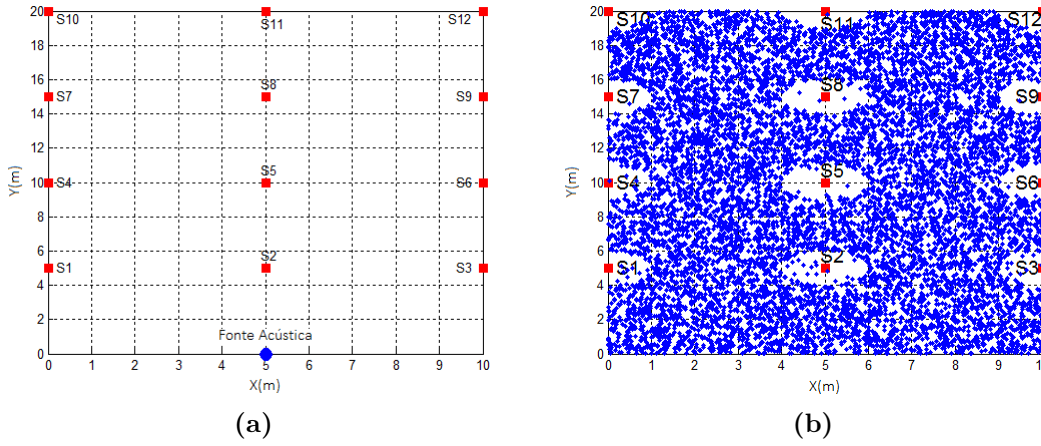


Figura 2.12: Condições de implementação experimental. (a) Condições de simulação para $N = 12$ sensores e 1 fonte . (b) Distribuição das 10000 fontes posicionadas aleatoriamente.

O algoritmo EHO foi executado em relação a cada uma das fontes individualmente, considerando os parâmetros $\alpha = 0.7$, $\beta = 0.1$, 100 elefantes organizados em 5 clãs, obtendo-se uma população de 20 elefantes por clã. À semelhança das simulações anteriores, o objetivo de usar 10000 fontes e, deste modo executar $M_c = 10000$ simulações de *Monte Carlo*, é reduzir o efeito do comportamento estocástico do algoritmo, e a dependência da posição da fonte no espaço.

As observações consideradas no ambiente de simulação, foram perturbadas com ruído gaussiano branco, no intervalo -80 dB a -45 dB em passos 5 dB . Como se pode verificar pela análise da figura 2.13, para valores de ruído reduzidos, o erro diminui com o aumento do número de sensores. No entanto, para valores de ruído elevado, o mesmo efeito não se verifica. Tal comportamento deve-se ao facto de

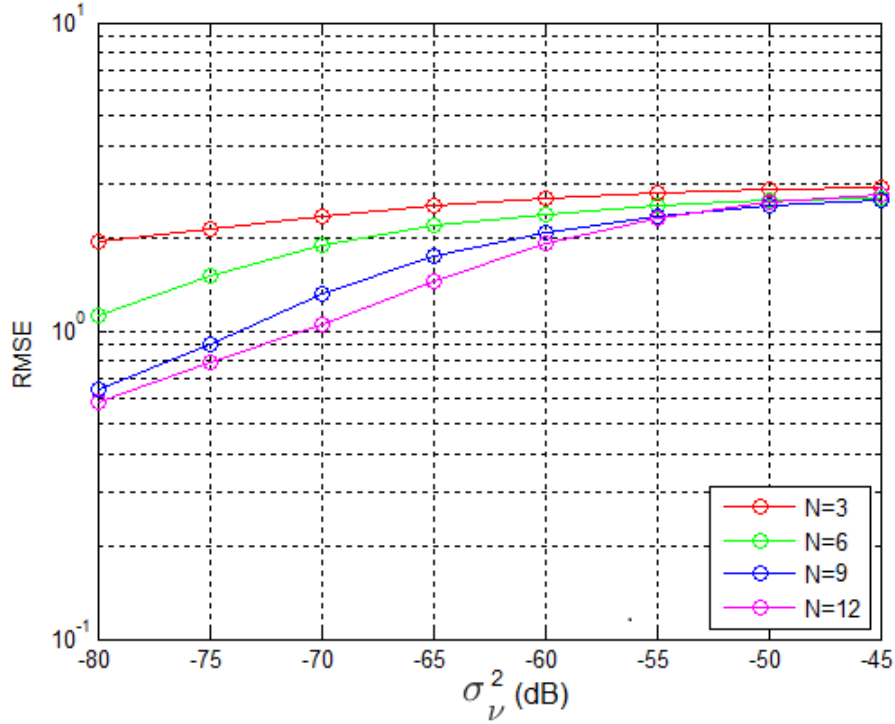


Figura 2.13: Resultados de simulação para $N = 3, 6, 9$ e 12 sensores e σ_v^2 (dB) = $-80 : 5 : -45$

os sensores serem adicionados ao modelo por ordem de distância, ou seja, os três últimos sensores incluídos são posicionados a uma distância de 20 m em relação à fonte. Em tal situação, a relação sinal/ruído ou SNR (Signal-to-Noise Ratio) diminui, significando que existe mais ruído do que sinal com informação, o que implica que estes sensores não tenham grande preponderância no algoritmo.

Tabela 2.6: Resultados de simulação para $N = 3, 6, 9$ e 12 sensores e σ_v^2 (dB) = $-80 : 5 : -45$

		σ_v^2 (dB)							
		-80	-75	-70	-65	-60	-55	-50	-45
N	3	1.94	2.13	2.35	2.55	2.70	2.79	2.86	2.92
	6	1.12	1.51	1.91	2.20	2.39	2.54	2.65	2.70
	9	0.65	0.90	1.32	1.75	2.10	2.36	2.55	2.66
	12	0.59	0.79	1.05	1.45	1.92	2.33	2.60	2.76
$ \Delta\text{RMSE}_{S3-S12} $		1.35	1.34	1.30	1.10	0.77	0.46	0.26	0.16

A tabela 2.6 quantifica a diferença incremental entre o RMSE obtido com $N = 3$ sensores e $N = 12$ sensores na rede (última linha), a partir dos valores de erro

obtidos para cada condição individual (número de sensores e variância). Como se pode observar, no caso em que $\sigma_v^2 = -80 \text{ dB}$, obtém-se um decréscimo de 1.35 m . O erro de 1.94 m com $N = 3$ sensores diminui para 0.59 m quando a rede inclui $N = 12$ sensores. No entanto, a mesma análise revela um decréscimo de apenas 0.16 m para uma variância do nível de ruído de $\sigma_v^2 = -45 \text{ dB}$.

2.6.2 Resultados Experimentais

Os testes experimentais foram realizados em duas fases distintas. Em primeiro lugar procura validar-se o modelo acústico, seguidamente, o algoritmo de localização, onde a fonte acústica é composta por um gerador de sinal, amplificado de modo a obter um valor específico de potência acústica.

Validação do Modelo Acústico

No sentido de validar o modelo de decaimento acústico, torna-se necessário determinar a potência acústica emitida pela fonte. Esta corresponde à quantidade de energia emitida pela fonte sonora, por unidade de tempo, ou seja, Joule por segundo (J/s). A potência acústica é um valor teórico, portanto não mensurável, sendo calculado e expresso em Watt (W), ou como um nível de potência acústica (L_W) em decibéis, onde a referência é 10^{-12} W [147]. A metodologia adoptada para ajustar a fonte acústica com uma determinada potência baseia-se na medição da pressão acústica (L_P) com um sonógrafo, sendo a relação entre as duas grandezas expressa por [148, 149]:

$$L_W = L_P - 10 \log_{10} \left(\frac{Q}{4\pi r^2} \right) \text{ (dB)}. \quad (2.12)$$

Na expressão 2.12, L_P corresponde ao nível de pressão acústica (em dB, neste caso com referência $20 \mu\text{Pa}$), Q um factor direccional considerado unitário e representando uma propagação omnidireccional esférica em torno da fonte, e r corresponde à distância de medição. Note-se que, ao considerar-se uma distância de medição de aproximadamente 30 cm ($r = \sqrt{Q/4\pi} = 0.2821 \text{ m}$), obtém-se $L_W \approx L_P$. Assim, o gerador de sinal é ajustado para gerar uma sinusóide com uma frequência

2. Métodos de Otimização Meta-Heurísticos para Solução do Problema de Localização de uma Fonte Acústica

de 2.3 kHz, e o amplificador para obter um nível de pressão acústica (L_P) de 100 dB, medidos com um sonógrafo CESVA modelo SC310, a uma distância de aproximadamente 30 cm, implicando uma potência acústica de $L_W = 100$ dB.

Para ajustar a potência, considera-se a equação

$$L_W(dB) = 10 \log_{10} \left(\frac{P}{P_0} \right), \quad (2.13)$$

onde P é a potência acústica e P_0 é a referência de potência, geralmente considerada como 1 pW, obtendo-se $P = 0.01$ W = 10 mW [147], em concordância com a simulação realizada.

A fonte é posicionada nas coordenadas (0 m, 5 m), correspondendo ao ponto azul da figura 2.15. As medidas foram obtidas entre 1 m e 20 m, com um incremento de 1 m, amostradas a uma frequência de 14 kHz (f_s). Tendo em conta a frequência do sinal acústico, a frequência de amostragem está consideravelmente acima do limite de Nyquist, assim como é enquadrada dentro dos valores aceitáveis para a medição de sinais audíveis em contextos de localização [150, 151]. O equipamento escolhido para realizar a tarefa de adquirir os dados consiste num Arduino UNO, que integra um microcontrolador ATMEGA328p (16 MHz), dada a sua flexibilidade do ponto de vista de operação. O microcontrolador integra um conversor analógico para digital (ADC) de 10 bits, o que implica uma resolução de aproximadamente 5 mV, tendo em conta a gama de entrada do ADC (5 V). Este valor terá implicações em termos da distância máxima a que se poderá observar um sinal, devido à atenuação do mesmo com a distância.



Figura 2.14: Diagrama de blocos da plataforma de aquisição de dados

Na figura 2.14 estão representados os vários blocos constituintes da plataforma de aquisição. Desta, realça-se os seguintes pontos:

- O firmware do microcontrolador realiza a aquisição dos dados com $f_s = 14 \text{ kHz}$, o que implica um intervalo de tempo de aproximadamente $70 \mu\text{s}$ entre aquisições.
- Após a aquisição, os dados são transmitidos numa comunicação série, assíncrona, para um computador onde corre a aplicação terminal *Putty*¹.
- A aplicação terminal realiza o registo dos dados em ficheiro, ficando disponíveis para um processamento posterior.
- Os dados são processados com o algoritmo de otimização EHO previamente implementado.

Tendo em conta o exposto, o firmware assume as tarefas de gerir o tempo de amostragem, através de interrupção, realizar a aquisição de dados, e proceder à sua transmissão. Note-se que os dados são transmitidos após a conversão, dado o microcontrolador não possuir memória volátil suficiente para armazenar os mesmos (2k Bytes de SRAM no ATMEGA328p). O firmware desenvolvido não assume grande complexidade, sendo apenas necessário garantir que a soma do tempo de execução das várias tarefas seja inferior ao tempo de amostragem. Em relação ao tempo de conversão do ADC, o mesmo é configurado com uma frequência de amostragem de 1 MHz , parametrizando o divisor de frequência intercalado entre o relógio do processador e o ADC, com um factor de 16. Embora se possa configurar o ADC com uma frequência maior, permitindo menor tempo de conversão, o mesmo é desaconselhado numa conversão de 16 bits². Assim, tendo em conta que o conversor necessita de 13,5 ciclos de relógio, uma conversão consome um tempo de aproximadamente $14 \mu\text{s}$. No que diz respeito à comunicação, e tendo em conta a necessidade de envio de vários bytes para cada leitura, esta é configurada para uma taxa de transmissão de 1 Mbps . Tal configuração implica um tempo de transmissão de $1 \mu\text{s/bit}$, ou seja, $10 \mu\text{s/byte}$ (8 bits de dados, um bit de início e um bit de paragem).

¹<https://www.putty.org/>

²http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

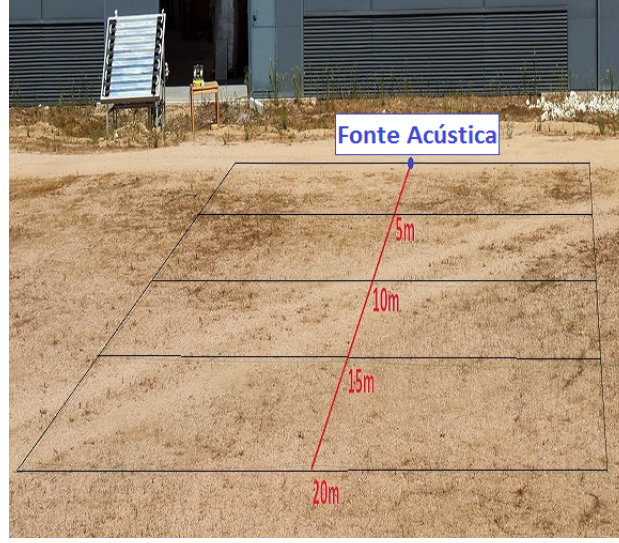


Figura 2.15: Campo de medições para validação do modelo

Os restantes componentes são externos, e são compostos por um microfone de condensador (ou microfone de Eletreto) com uma sensibilidade (S) de -55 dB, gerando um potencial de 1.77 mV/ P_a , segundo:

$$S(\text{dB}) = 20 \log_{10} \left(\frac{S(\text{mV}/P_a)}{A_{\text{Ref}}} \right). \quad (2.14)$$

Na expressão 2.14, $A_{\text{Ref}} = 1$ V/ P_a . Tendo em conta que as observações aplicadas ao modelo apresentado em 2.3 representam energia acústica, o último passo consiste em obter a intensidade acústica (I) em W/m^2 , pela relação com a pressão acústica:

$$I = \frac{p^2}{Z_0} \quad (2.15)$$

onde Z_0 representa a impedância acústica, neste caso a impedância do ar $Z_0 = 400$ Ns/m^3 [148, 149]. Tendo em conta o sinal discreto adquirido, p corresponde à pressão acústica média, obtida como $p = \frac{1}{N_a} \sum_{i=1}^{N_a} x^2(i)$, onde N_a corresponde ao número de amostras $x(i)$ [152].

O microfone é alimentado pelo regulador interno do módulo Arduino e polarizado com uma resistência de $6,8$ k Ω . Um condensador de 1 uF é usado para implementar um filtro passa-alto, eliminando a componente contínua correspondente à polarização

do microfone, assim como flutuações de baixa frequência. A saída do filtro é ligada diretamente à entrada analógica do microcontrolador (figura 2.16).

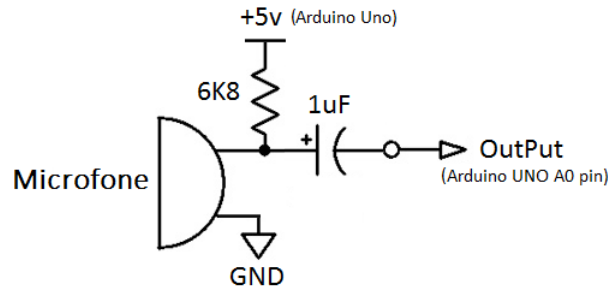


Figura 2.16: Circuito de polarização do microfone

Para visualizar o sinal durante a execução das medições em campo, foi também utilizado um osciloscópio Hantek, modelo 6022BE, para garantir que todas as ligações estivessem operacionais, e que os sinais estavam efetivamente a serem adquiridos. Como o microfone é omnidirecional, o mesmo foi orientado na posição vertical. A configuração combinada da plataforma de aquisição pode ser observada na figura 2.17.

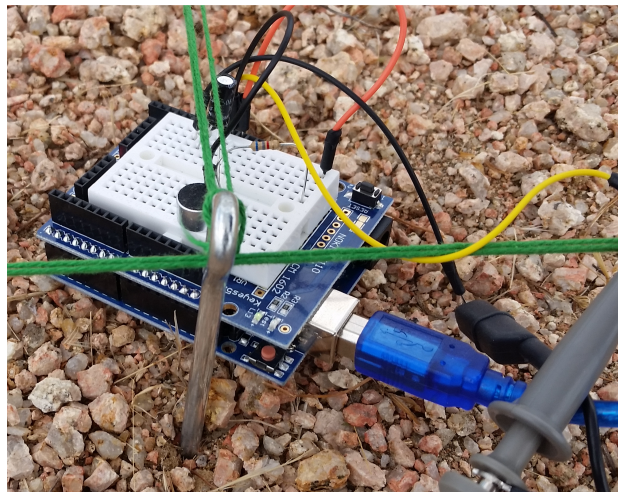


Figura 2.17: Plataforma de aquisição (circuito do microfone e placa Arduino UNO)

A figura 2.18 mostra três sinais diferentes medidos nas distâncias de 1 m, 10 m e 20 m. Como pode ser observado na figura, os sinais adquiridos mostram algumas características esperadas, nomeadamente, o facto de a sua amplitude diminuir com a distância, numa relação de proporcionalidade inversa. Também é notória

a componente constante da tensão (componente DC), devido à polarização do microfone que não foi completamente eliminada pelo condensador de desacoplamento. Isso ocorre devido à impedância de fugas do próprio condensador, acoplada à impedância do microfone e da entrada analógica. No entanto, realça-se que este valor é bastante reduzido, tendo sobretudo influência quando são medidos sinais distantes da fonte. Também pode ser observado um ruído de alta frequência, que deve ser esperado devido à frequência do relógio do processador. O sinal do relógio do processador é gerado a partir de um cristal de quartzo que possui a propriedade mecânica de vibrar a uma frequência fixa. Como a vibração é mecânica, propaga-se naturalmente na estrutura até ao microfone.

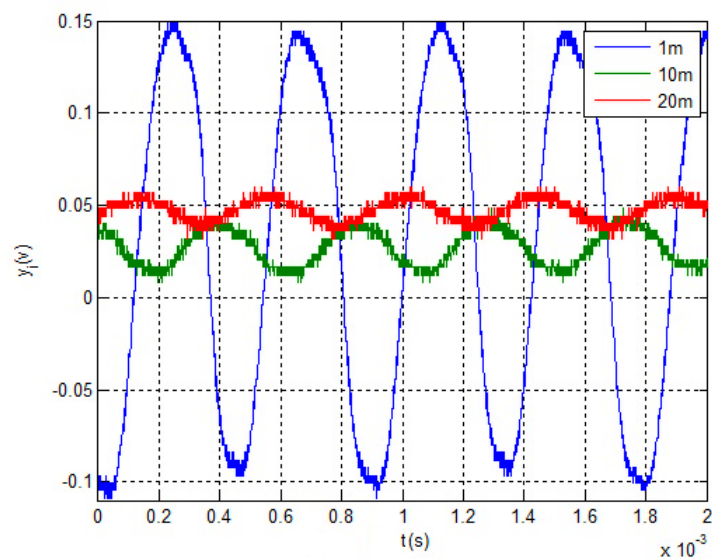


Figura 2.18: Sinais medidos em três distâncias diferentes

Embora o ruído de alta frequência não se apresente como um problema em termos de medição, por outro lado, o ruído de baixa frequência, como por exemplo o efeito do vento, é mais difícil de observar pela simples análise do gráfico. No sentido de caracterizar o sinal e permitir a implementação de filtros digitais, procede-se a uma análise espectral. Assim, implementou-se a Transformada Fourier (FFT) do sinal medido a 5 m, conforme figura 2.19, para avaliar todas as harmónicas presentes no sinal medido. Como pode ser observado pela FFT, a componente

DC é notória, bem como a componente fundamental (2.3 kHz), vários harmônicos de alta frequência, e ruído de baixa frequência.

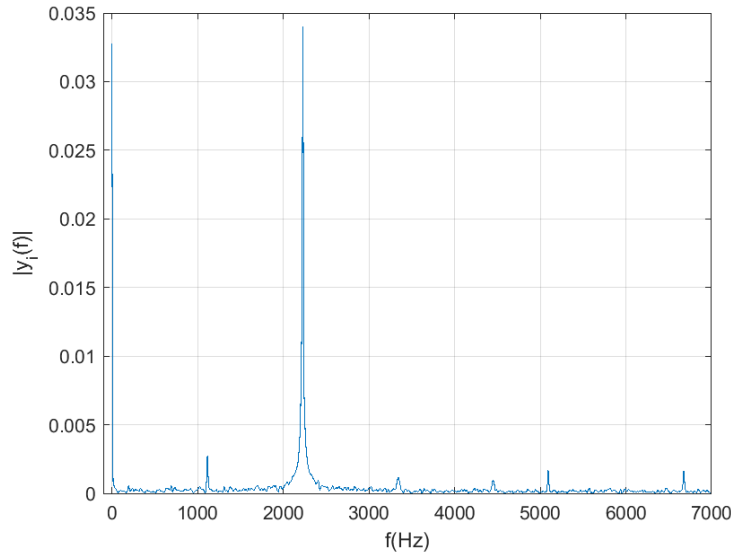


Figura 2.19: FFT do sinal medido a $5m$

Com o objetivo de eliminar fontes de ruído, considerou-se uma etapa de pré-processamento, constituída pelo projeto de um filtro digital de resposta finita (FIR) do tipo passa-banda. A região de transição da banda de paragem para a banda de passagem foi ajustada com frequências de 1.8 kHz e 2 kHz , e a região de transição da banda de passagem para a banda de paragem foi ajustada nas frequências de 2.6 kHz e 2.8 kHz . As bandas de paragem foram consideradas com atenuações de 60 dB e 80 dB , respetivamente. Um parâmetro importante do filtro projetado é o ganho da banda passante, que neste caso é de 0 dB . Caso contrário, esse ganho teria que ser considerado no modelo, no parâmetro g_i . O módulo da resposta em frequência do filtro pode ser observado na figura 2.20, sendo o filtro projetado na ferramenta *Filter Designer*, integrada no MATLAB®.

Como o esquemático do microfone não possui amplificador e o conversor ADC do microcontrolador não possui amplificador de ganho programável, o ganho considerado em cada sensor é unitário ($g_i = 1$). A partir das medições efetuadas, através do esquema da figura 2.17, e do modelo teórico apresentado anteriormente, obtemos os gráficos da figura 2.21. As medições foram realizadas entre 1 m e 20 m ,

2. Métodos de Otimização Meta-Heurísticos para Solução do Problema de Localização de uma Fonte Acústica

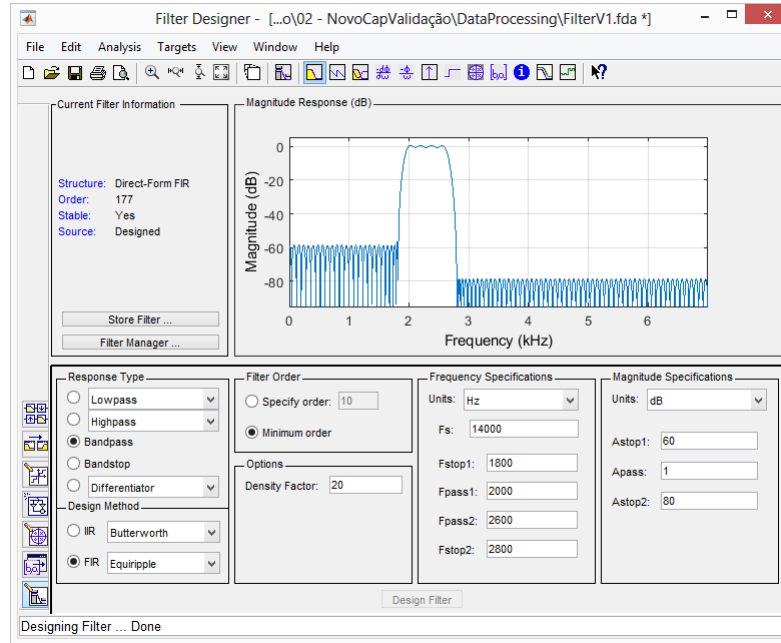


Figura 2.20: Projeto do filtro digital (*Filter Designer*-MATLAB®)

em intervalos de 1 m, numa linha reta com origem na fonte acústica, onde d (m) corresponde à distância de medição, obtendo-se um vetor com $N_a = 20$ pontos.

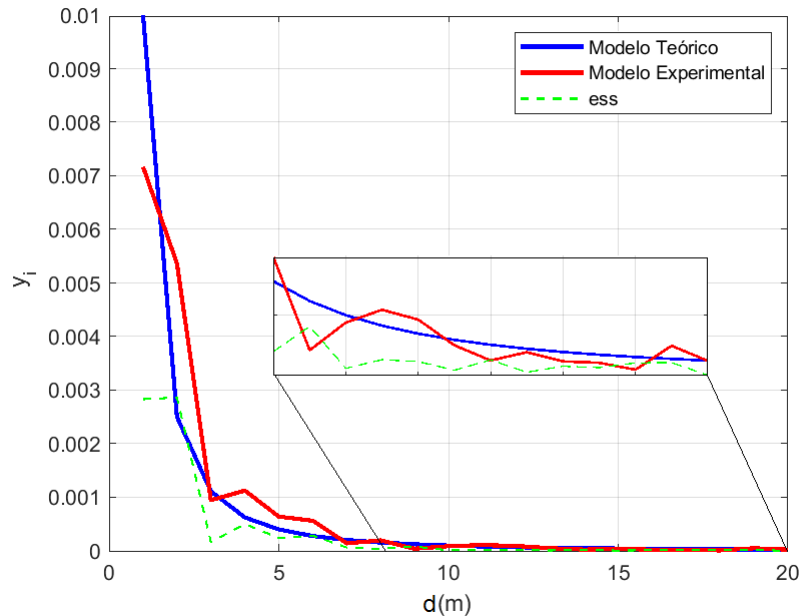


Figura 2.21: Modelo teórico e experimental do modelo de decaimento acústico

Para avaliar a aproximação obtida entre o modelo teórico e o modelo experimental, calcula-se o erro médio relativo (Ess) e o seu desvio padrão, em relação ao sinal

de erro, de acordo com a expressão:

$$Ess = \frac{1}{N_a} \sum_{i=1}^{N_a} \left| \frac{y_{T_i} - y_{E_i}}{y_{T_i}} \right| \times 100\%, \quad (2.16)$$

onde y_{T_i} corresponde à entrada i do vetor dos valores obtidos através do modelo e y_{E_i} à entrada i dos valores obtidos por medição.

O erro obtido é de $Ess = 47.29\%$, com um desvio padrão relativo de $\sigma_{ss} = 15.27\%$. Note-se que, junto à fonte onde a amplitude dos sinais são superiores, também os desvios são superiores, contribuindo para um erro e desvio padrão mais elevados. Do ponto de vista absoluto, para o erro médio obtêm-se um valor de 0.0048 W/m^2 com um desvio padrão de $7.33 \times 10^{-4} \text{ W/m}^2$.

Validação Algoritmo de Localização

Para validar o algoritmo de localização EHO, as coordenadas dos sensores (pontos de medição no campo) colocam-se de acordo com a figura 2.12a. Em 2.22, representam-se as posições dos sensores, projetadas no campo de medição, onde o ponto azul representa a fonte localizada em $(0 \text{ m}, 5 \text{ m})$.

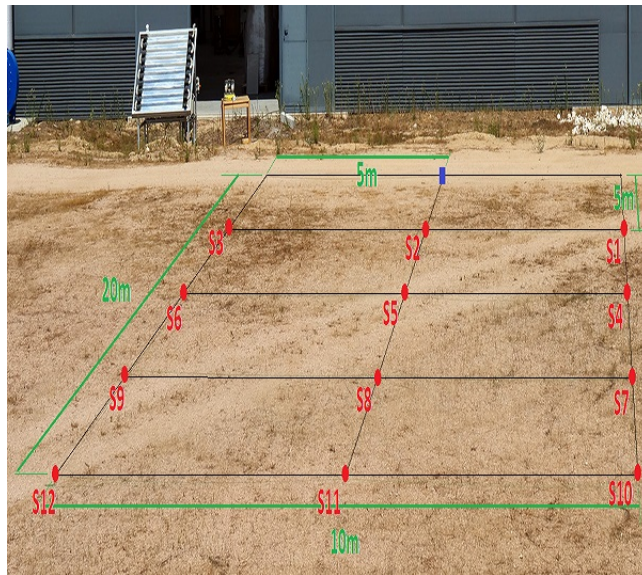


Figura 2.22: Medidas de campo para o algoritmo de localização EHO

Os parâmetros do algoritmo mantêm-se os mesmos em relação às simulações realizadas, no entanto, devido ao comportamento estocástico do mesmo, foram

2. Métodos de Otimização Meta-Heurísticos para Solução do Problema de Localização de uma Fonte Acústica

realizadas 1000 execuções para cada conjunto de combinações de sensores ($N = 3$, $N = 6$, $N = 9$, $N = 12$).

Os gráficos da figura 2.23 representam os resultados obtidos para cada configuração do número de sensores, em que as cruzes verdes representam as várias soluções para cada execução do método. Para comparar os resultados com a adição de sensores, foi calculado para cada conjunto de sensores: o RMSE, o desvio padrão, e o tempo de execução ($Tempo_{CPU}(s)$) em relação às 1000 execuções do algoritmo. Os resultados são apresentados na tabela 2.7.

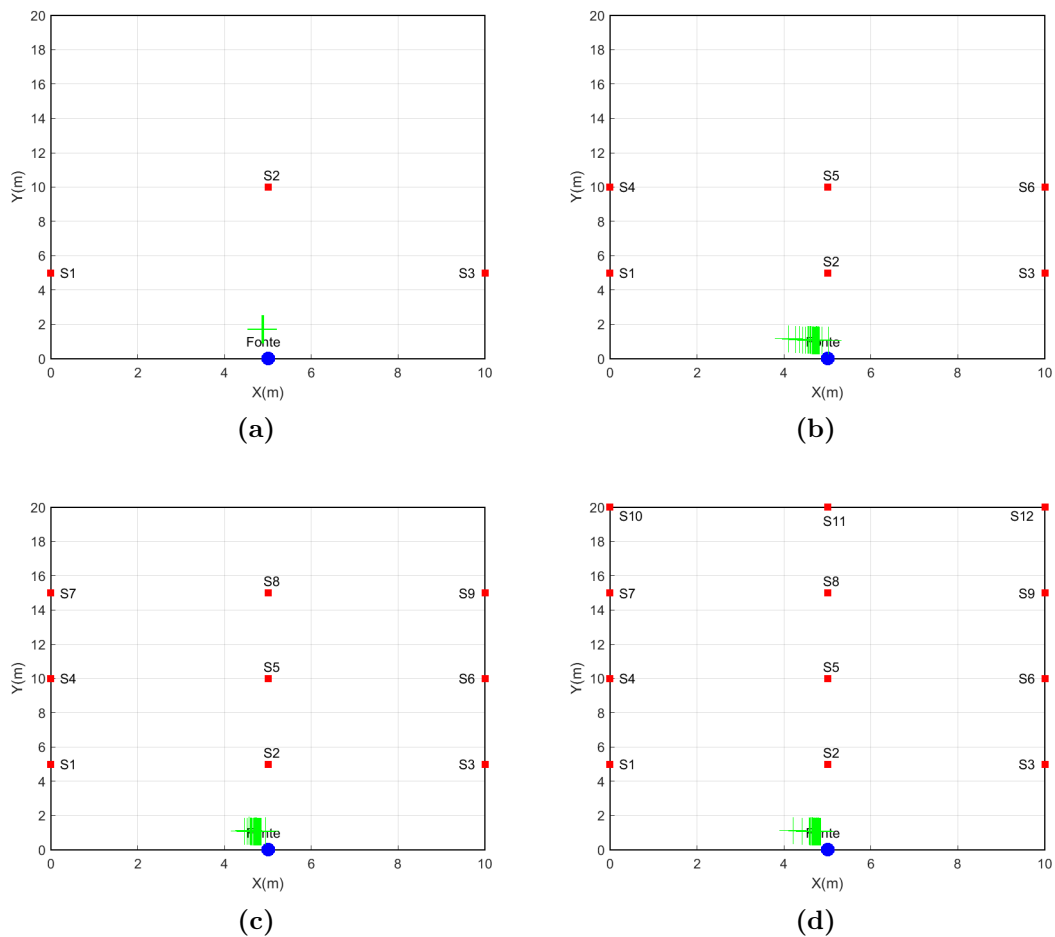


Figura 2.23: Resultado do algoritmo EHO com dados reais. (a) $N = 3$ sensores e 1 fonte (b) $N = 6$ sensores e 1 fonte (c) $N = 9$ sensores e 1 fonte (d) $N = 12$ sensores e 1 fonte

Na tabela 2.7 pode ser observado que o erro diminui com o aumento do número de sensores, embora sem muito significado quando N atinge valores mais

Tabela 2.7: Resultado da aplicação do método EHO

N	$RMSE_{Exp}(m)$	$RMSE_{Simul}(m)$	$\sigma_{Exp}(m)$	$\overline{Tempo}_{CPU}(s)$	S_i
3	1.7178	2.3502	0.0025	0.2346	1,3,5
6	1.0984	1.9094	0.0193	0.2402	1..6
9	1.0923	1.3185	0.0105	0.2373	1..9
12	1.0923	1.0517	0.0156	0.2350	1..12

elevados. O tempo de execução do método (\overline{Tempo}_{CPU}) é praticamente constante, independentemente do número de sensores. Tal situação deve-se ao facto do tamanho da rede apenas ter influência no cálculo da função de custo e não diretamente no algoritmo. Da análise das figuras, nomeadamente o resultado em relação a $N = 3$, pode observar-se a influência na geometria da distribuição dos sensores (figura 2.23a). Tal situação é também observável na tabela 2.7, onde o desvio padrão correspondente a $N = 3$ é inferior aos restantes cenários.

2.7 Conclusões

No presente capítulo, foi apresentada uma nova abordagem para o problema de localização baseado na medição da energia acústica, com base no algoritmo EHO, inspirado no comportamento dos elefantes na natureza. Em nítido contraste com os algoritmos existentes que aplicam aproximações para alcançar a sua solução final, o método EHO aborda diretamente o problema de ML não convexo. Com a realização de simulações computacionais, os principais parâmetros do algoritmo EHO foram otimizados, de modo a que pudesse ser aplicado ao problema de interesse com maior eficácia. O desempenho do algoritmo proposto foi comparado com trabalhos existentes na literatura. Os resultados da simulação demonstraram que o algoritmo EHO reduz o erro de estimativa em todas as configurações consideradas, sobretudo em ambientes com elevada potência de ruído. Além disso, o método EHO representa uma excelente relação entre a complexidade computacional e a precisão das estimativas, uma vez que é significativamente menos complexo que os métodos determinísticos.

Foram ainda apresentados resultados para localização acústica baseada em energia com base na implementação de EHO, a partir de observações obtidas com medições reais, em campo aberto. Tais resultados, demonstram a correlação entre os dados de simulação e os dados obtidos a partir das medições de campo. Do ponto de vista do modelo acústico, foi inicialmente obtido um erro absoluto considerado aceitável, com um baixo desvio padrão entre os valores teóricos e experimentais. De seguida, a aplicação do método de EHO permitiu obter resultados que, quando comparados com as simulações para o mesmo nível de ruído, realçaram as características esperadas, sobretudo em termos da sua relação com o número de sensores.

Questões sob Investigação: Tendo em conta o problema de aplicação concreto, de que forma os algoritmos baseados em enxames de partículas podem ser otimizados? Tendo em conta que as partículas de enxames são, normalmente geradas aleatoriamente, por estarmos perante algoritmos genéricos a qualquer problema de otimização, de que forma se poderá inicializar a população de modo a se ajustar ao modelo de energia acústica? Que efeito na convergência dos métodos poderá ter uma procura local entre gerações dos enxames de partículas? As metodologias desenvolvidas podem ser integradas noutros algoritmos?

3

Desenvolvimento de Metodologias para o Aperfeiçoamento de Algoritmos Meta-Heurísticos

Conteúdos

3.1	Motivação e Objetivos	49
3.2	Inicialização Inteligente da População	50
3.3	Métodos Baseados no Gradiente	57
3.4	Algoritmo Híbrido	59
3.5	Simulações e Análise de Resultados	62
3.5.1	Aplicação ao Método Original	63
3.5.2	Aplicação à mais Recente Versão do Método EHO	72
3.5.3	Aplicação ao Método " <i>Coyote Optimization Algorithm</i> "	77
3.6	Conclusões	82

3.1 Motivação e Objetivos

No presente capítulo, é proposta uma nova abordagem ao problema de localização acústica, baseada em otimização meta-heurística, e no seguimento do trabalho desenvolvido no capítulo 2, no qual são propostas duas inovações. Essas serão baseadas nos dois pontos seguintes:

- Geração da população inicial baseada em informação do modelo. A inovação apresentada consiste, primeiramente, no desenvolvimento de um novo método baseado numa abordagem geométrica, formulado para criar a primeira iteração da população e proposto a partir de um desenvolvimento analítico.
- Em segundo lugar, a convergência do método é acelerada com base numa busca local, implementada com um método baseado num gradiente iterativo, sem a necessidade do cálculo de derivadas, em cada iteração do método EHO original. Os resultados de simulação irão demonstrar que o desempenho do algoritmo proposto se aproxima do desempenho do algoritmo original, em termos de precisão da localização, mas pode reduzir o número de iterações em valores superiores a 50%.

Por último, a nova metodologia proposta é alargada a outros algoritmos meta-heurísticos, no sentido de validar e demonstrar a extensão e generalização das propostas.

3.2 Inicialização Inteligente da População

Os métodos para geração de populações aleatórias e a sua aplicação como estimativas iniciais têm sido alvo de desenvolvimento e propostas na literatura científica. O principal objetivo tem sido a geração de populações uniformes, garantindo que a distribuição dos pontos gerados cobre de uma forma homogénea o espaço de procura. A inicialização da população é um passo crucial de um algoritmo, uma vez que pontos iniciais afastados do ótimo global impedirão o mesmo de encontrar soluções globais [153]. Além de métodos genéricos, como o gerador de números pseudoaleatórios (PRNG) [154] ou o gerador de números caóticos (CNG) [155], métodos de inicialização específicos em relação a determinadas aplicações, também foram considerados como um conjunto particular de problemas, nomeadamente para o design de antenas [153] ou segmentação de imagem [156]. Verifica-se que, mesmo considerando funções genéricas de teste e métodos de inicialização estocásticos,

diferentes metodologias ao nível da inicialização da população implicam diferenças significativas nos resultados obtidos [157].

Conforme analisado no capítulo 2, o algoritmo EHO original inicializa e organiza os elefantes em clãs, distribuídos aleatoriamente no espaço de procura, sem considerar nenhum conhecimento prévio do problema em si. Isso deve-se principalmente ao facto de o algoritmo ter sido proposto para uso geral e ter como propósito ser aplicado a qualquer tipo de problema, sem o conhecimento da função a ser otimizada (situação transversal nos métodos meta-heurísticos). No entanto, ao aplicar o algoritmo a um problema específico, podemos aproveitar o conhecimento do modelo, da função de custo que é aplicada, e a sua relação com as observações utilizadas. Se considerarmos o modelo de energia acústica apresentado na equação 2.3, é possível obter uma estimativa da distância entre o sensor \mathbf{S}_i e a fonte \mathbf{X} , a partir das observações y_i :

$$\hat{d}_i = \left(\frac{g_i P}{y_i} \right)^{\frac{1}{\beta_E}}, \quad \text{for } i = 1, \dots, N. \quad (3.1)$$

A equação 3.1 fornece uma estimativa da distância de cada sensor à fonte acústica (de coordenadas desconhecidas), o que significa que a fonte está confinada a um raio conhecido, em relação a cada sensor. Se as medições fossem livres de ruído, as verdadeiras coordenadas da fonte seriam o ponto de intersecção de todos os raios amostrados.

Veja-se, no entanto, que cada observação, e portanto, cada estimativa de distância, está corrompida com ruído, $\nu_i \sim \mathcal{N}(0, \sigma_\nu^2)$, implicando que não exista um ponto de convergência único e comum a todas as distâncias. Para estudar a região mais provável relativa aos pontos de intersecção, consideramos um grupo de 3 sensores. A extrapolação para outro conjunto de números de sensores é direta e não será considerada no presente capítulo.

Como afirmado anteriormente, devido ao ruído de observação, várias configurações podem ocorrer entre as intersecções dos raios. Com o objetivo de demonstrar algumas situações possíveis, consideramos uma configuração de sensores $N = 9$, com observações y_i geradas entre \mathbf{S}_1 , \mathbf{S}_4 e \mathbf{S}_7 (identificados numericamente) e uma fonte representada no espaço pelo quadrado azul (figura 3.1). No caso da figura

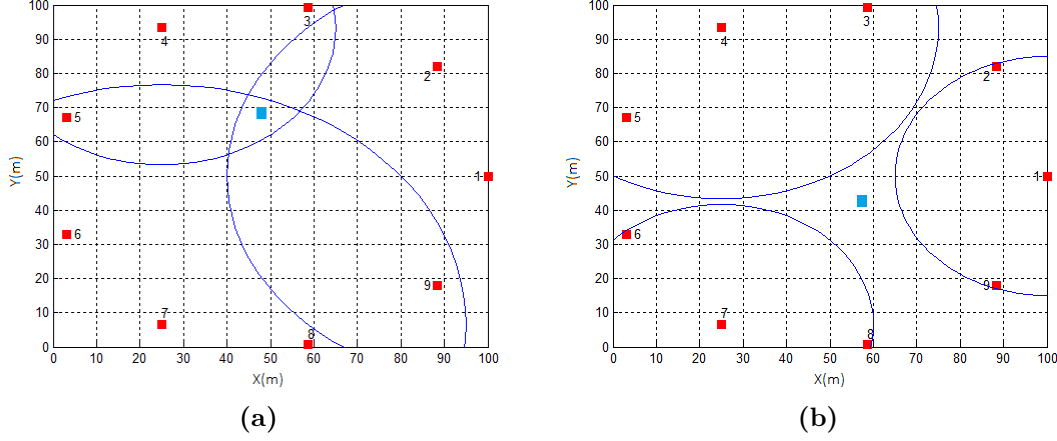


Figura 3.1: Possíveis configurações segundo a distância. (a) Circunferências secantes (b) Circunferências externas

3.1a, o ruído considerado nas observações teve um efeito aditivo na estimativa da distância, formando um espaço convexo. No caso da figura 3.1b, o ruído das observações apresenta um efeito contrário, não estando a solução limitada por um espaço fechado. Esta situação assume elevada importância, tendo em conta que a maioria dos métodos determinísticos referidos no capítulo 2 exigem que se verifiquem condições de delimitação, ou seja, que as interseções formem um espaço convexo. No presente trabalho será apresentada uma solução para cada um dos casos, sendo ainda extrapolado para as situações em que uma combinação das configurações referidas possa ocorrer. Isto torna o método apresentado imune ao tipo de ruído com que as observações são perturbadas, e portanto de maior robustez e viabilidade de implementação.

Proposição 3.1 (Circunferências Secantes). *Considere-se três circunferências com centro nas coordenadas $\mathbf{S}_i \in \mathbb{R}^2$ e raios $R_i \in \mathbb{R}, i = 1, \dots, 3$, onde as circunferências se interceitam em pelo menos dois pontos, quando a expressão 3.2 é logicamente verdadeira (Figura 3.1a),*

$$(d_{12} < R_1 + R_2) \wedge (d_{13} < R_1 + R_3) \wedge (d_{23} < R_2 + R_3) \quad (3.2)$$

3. Desenvolvimento de Metodologias para o Aperfeiçoamento de Algoritmos Meta-Heurísticos

onde d_{ij} se refere à distância Euclidiana entre \mathbf{S}_i e \mathbf{S}_j .

Os pontos que delimitam o espaço convexo definido pela interseção das três circunferências, \mathbf{P}_{C1} , \mathbf{P}_{C2} e \mathbf{P}_{C3} , são determinados pelas seguintes expressões:

$$P_{ck} = \begin{cases} x_{ck} = P_x(x) - h \frac{S_j(y) - S_i(y)}{\|S_i - S_j\|} \\ y_{ck} = P_x(y) + h \frac{S_j(x) - S_i(x)}{\|S_i - S_j\|} \end{cases}, \quad \|P_k - P_{ck}\| < R_k$$

$$P_{ck} = \begin{cases} x_{ck} = P_x(x) + h \frac{S_j(y) - S_i(y)}{\|S_i - S_j\|} \\ y_{ck} = P_x(y) - h \frac{S_j(x) - S_i(x)}{\|S_i - S_j\|} \end{cases}, \quad \|P_k - P_{ck}\| > R_k$$

onde x_{ck} e y_{ck} são as coordenadas do ponto \mathbf{P}_{ck} , e

$$\mathbf{P}_x = \mathbf{P}_i + a \frac{\mathbf{P}_j - \mathbf{P}_i}{\|\mathbf{S}_i - \mathbf{S}_j\|}$$

$$a = R_i^2 - R_j^2 + \|\mathbf{S}_i - \mathbf{S}_j\|^2$$

$$h^2 = R_i^2 - a^2$$

Demonstração. Ver anexo A, secção A.1. ■

Proposição 3.2 (Circunferências Externas). *Considere-se três circunferências com centro nas coordenadas $\mathbf{S}_i \in \mathbb{R}^2$ e raios $R_i \in \mathbb{R}, i = 1, \dots, 3$, onde as circunferências não se intercetam em qualquer ponto (figura 3.1b), o que corresponde a que a expressão 3.3 seja logicamente falsa:*

$$(d_{12} < R_1 + R_2) \vee (d_{13} < R_1 + R_3) \vee (d_{23} < R_2 + R_3). \quad (3.3)$$

Conforme mencionado anteriormente, d_{ij} refere-se à distância Euclidiana entre \mathbf{S}_i e \mathbf{S}_j . Para determinação do ponto de interesse considerarmos o segmento reta entre \mathbf{S}_i e \mathbf{S}_j , $\overline{\mathbf{S}_i\mathbf{S}_j}$, e a sua interseção com os raios das circunferências, aqui designados por \mathbf{P}_A e \mathbf{P}_B . O ponto que se pretende determinar, \mathbf{P}_{ij} , corresponde ao ponto médio do segmento de reta $\overline{\mathbf{P}_A\mathbf{P}_B}$, obtido através das seguintes expressões:

$$\mathbf{P}_{ij} = \frac{\mathbf{P}_A + \mathbf{P}_B}{2} \quad (3.4)$$

$$\mathbf{P}_A(x, y) = \begin{cases} P_A(x) = S_i(x) + \cos(\alpha_p)R_i \\ P_A(y) = S_i(y) + \sin(\alpha_p)R_i \end{cases} \quad (3.5)$$

$$\mathbf{P}_B(x, y) = \begin{cases} P_B(x) = S_j(x) - \cos(\alpha_p)R_j \\ P_B(y) = S_j(y) - \sin(\alpha_p)R_j \end{cases} \quad (3.6)$$

Demonstração. Ver anexo A, seção A.2. ■

Realça-se que, em aplicações em ambiente real, a figura 3.1a corresponderia a ruído aditivo em todos os sensores, enquanto a figura 3.1b corresponderia a ruído subtrativo em todas as leituras relativas aos sensores.

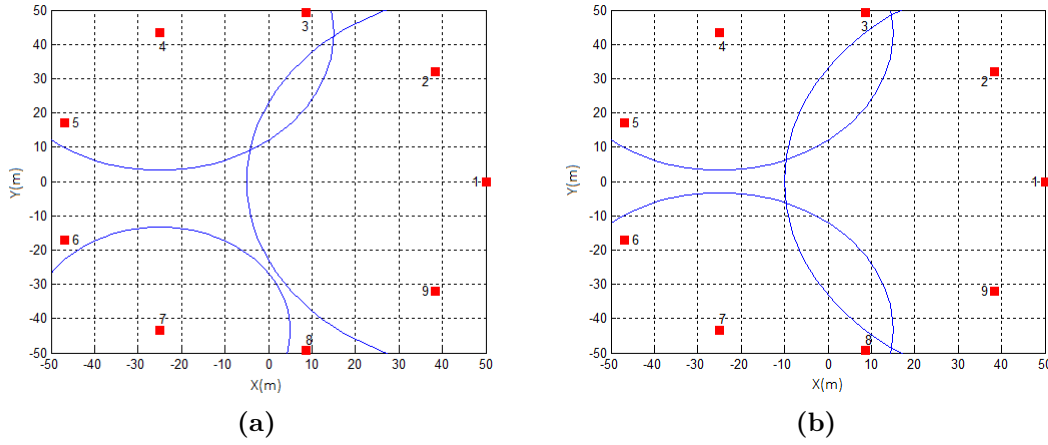


Figura 3.2: Diferentes combinações, dependentes do ruído das observações. (a) Uma interseção (b) Duas interseções

No entanto, como estamos a considerar um conjunto de três sensores para determinar o ponto de interesse, diferentes combinações podem ocorrer com o ruído das observações, ou seja, ruído aditivo num sensor e ruído subtrativo nos restantes, entre outras.

As figuras 3.2a e 3.2b representam duas situações distintas, onde, no primeiro caso, obtemos apenas uma interseção e na segunda duas interseções. Situações com circunferências internas também podem ocorrer, onde em relação às mesmas, para encontrar pontos médios, aplicaríamos as Proposições 3.1 e 3.2 para cada par de sensores respetivamente. O objetivo de determinar os pontos centrais do espaço convexo, limitado pelas interseções ou pelos pontos médios, no caso de circunferências externas, é o facto da solução do problema, que será objeto de

determinação pelo algoritmo EHO, considerar-se localizada nas regiões de interceção. Assim, o objetivo da metodologia desenvolvida consiste em:

1. Inicializar os clãs no centro dos pontos de interceção, no caso de circunferências externas;
2. Inicializar os clãs no centro de massa do espaço convexo limitado pelas interceções, no caso de circunferências internas.

Considerando o procedimento proposto, a posição da matriarca será inicializada no centro e os elefantes pertencentes ao mesmo clã serão inicializados no interior de uma circunferência delimitada no espaço, pelo maior raio que cobre todos os pontos de interceção. Observe-se que como estamos a considerar três sensores para cada conjunto de interceções, o número total de sensores deve ser múltiplo de três, e o número de clãs que serão gerados está diretamente correlacionado com o número de sensores segundo a equação:

$$N_{Clans} = N/3 \quad (3.7)$$

onde N é o número total de sensores.

Tendo em conta tratar-se de um espaço bidimensional, este é o valor mínimo para formar um espaço fechado. No entanto, um número superior de sensores pode ser considerado para gerar o centro do clã.

A figura 3.3a mostra um exemplo de geração de clãs para a configuração inicial apresentada no capítulo 2, em que $\mathbf{X}_{\text{True}} = (40, 70)$ m e um ruído de $\sigma_v^2 = -40$ dB adicionado às observações. Os restantes parâmetros do método e do modelo mantêm-se em relação ao capítulo 2. Nesta, as linhas azuis representam os raios correspondentes às estimativas de distância e as circunferências verdes os raios determinados para a geração dos clãs, com centro em "*" (a vermelho). A população representada por "•" (a verde), é gerada aleatoriamente e circunscrita aos respetivos clãs. A figura 3.3b mostra uma aproximação à área espacial em que a fonte real está localizada (símbolo "+" a vermelho). Como se pode observar, os

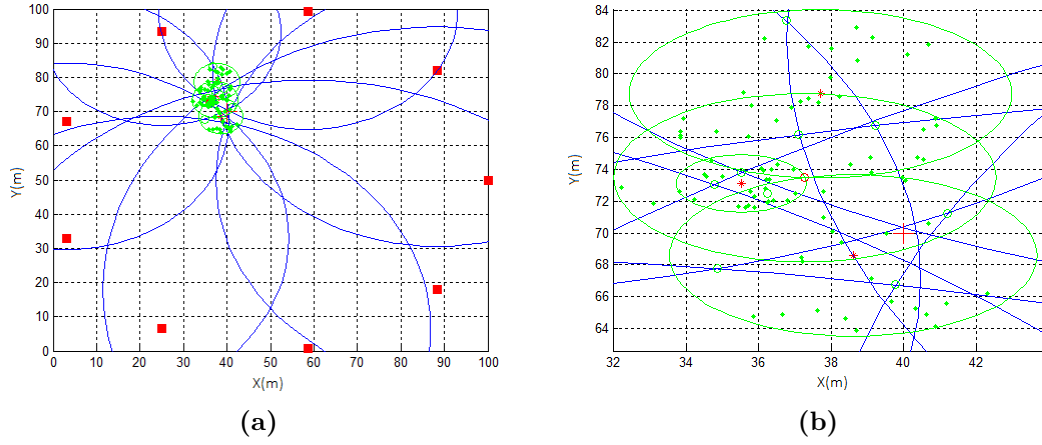


Figura 3.3: Exemplo de uma geração de clãs, $N = 9$, $X = (30, 70)$ m e $\sigma_v^2 = -40$ dB. (a) Clãs e população gerada (b) Visualização no espaço alargado

elefantes estão localizados em torno da solução real dentro de uma área restrita, em vez de distribuídos pela totalidade do espaço de procura.

Algoritmo 3.1 Procedimento para Inicialização dos Clãs

```

1: function CLANINIT( $S, \hat{d}$ )
2:    $w = 1$ 
3:    $L = \text{length}(S)$  ▷ Número de sensores
4:    $q = 0 : L/3 : L - L/3$  ▷ Selecciona o sensor mais distante para cada grupo
5:   for  $k = 1 : L/3$  do ▷ Grupos de 3 sensores para criação dos clãs
6:      $p = \|S_{kq_1} - S_{kq_2}\|$ 
7:     if  $D_k < D_{kq_2}$  then ▷ Verifica posição dos pontos
8:        $Rm = D_{kq_1}; RM = D_{kq_2}$ 
9:     else
10:       $Rm = D_{kq_2}; RM = D_{kq_1}$ 
11:    end if
12:    if  $(D_k + D_k + q_2 > p) \ \&\& \ (p + Rm > RM)$  then
13:       $P_x = \text{"Aplicar Proposição 3.1"}$ 
14:    else
15:       $P_x = \text{"Aplicar Proposição 3.2"}$ 
16:    end if
17:     $P(w) = P_x$ 
18:     $w ++$ 
19:    ... Repetir o procedimento para os restantes grupos ( $q$ )
20:  end for
21: end function

```

A nova abordagem proposta e sistematizada na forma de pseudocódigo no

algoritmo 3.1 pretende apresentar-se como sendo uma melhoria, tendo em conta que as várias gerações de população no processo iterativo do método poderão ser reduzidas. Consequentemente, reduzindo o número de avaliações da função de custo ($N_{Clans} \cdot N_{C_i} \cdot N_{Geracoes}$), como clarifica a expressão,

$$N_{Eval} = P_{Tam} \cdot N_{Geracoes}. \quad (3.8)$$

onde P_{Tam} corresponde ao tamanho da população ($P_{Tam} = N_{Clans} \cdot N_{C_i}$).

3.3 Métodos Baseados no Gradiente

Os métodos baseados no gradiente para otimização consistem numa procura local. Originalmente propostos por Horn e Schunck [158], assumem a forma de um processo iterativo de pesquisa, ou seja, o algoritmo começa com uma estimativa inicial $\mathbf{x}^0 \in \mathbb{R}^n$ e evolui a cada passo que define um novo ponto \mathbf{x}^{k+1} a partir de um ponto \mathbf{x}^k conforme a expressão (regra de Armijo [159]):

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \mathbf{d}_s^k. \quad (3.9)$$

Na expressão, $\alpha^k \in \mathbb{R}$ corresponde à etapa de pesquisa e $\mathbf{d}_s^k \in \mathbb{R}^n$ é um vector que indica a direção da pesquisa, com a finalidade de encontrar o mínimo de uma função dentro de um número finito de iterações [160, 161].

O problema da otimização baseada no gradiente consiste em tornar a proposição 3.3 numa proposição verdadeira.

Proposição 3.3. *Dado um vector \mathbf{x}^k , considerando $\nabla f(\mathbf{x}^k) \neq 0$, encontre $\alpha^k \in \mathbb{R}$ e $\mathbf{d}_s^k \in \mathbb{R}^n$ garantindo $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$.*

Os métodos baseados no gradiente usam o declive da função de custo, considerando um determinado ponto para determinar uma direção, com o propósito de encontrar um mínimo.

O presente trabalho não pretende ser um estudo exaustivo sobre os métodos baseados no gradiente, para os quais neste capítulo serão considerados apenas alguns dos métodos encontrados e mais citados na literatura.

O método mais simples, embora não seja o mais eficiente para determinar a direção da pesquisa, é assumir \mathbf{d}_s^k como a direção oposta ao gradiente da função. Note-se que o gradiente de uma função multi-variável num ponto corresponde à direção do máximo crescimento máximo da mesma. Portanto, um vector na direção definida implicará a direção da deterioração máxima [162]. Assim, a equação 3.9 assume a forma:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \nabla f(\mathbf{x}^k). \quad (3.10)$$

A determinação de α^k , que corresponde à etapa de pesquisa, será considerada um problema de pesquisa unidimensional, de tal modo que:

$$\alpha^k = \underset{\alpha > 0}{\operatorname{argmin}} f(\mathbf{x}^k - \alpha^k \nabla f(\mathbf{x}^k)). \quad (3.11)$$

O método de pesquisa unidimensional, conforme considerado em 3.10, tem um grande impacto na robustez do novo ponto para o processo iterativo. Portanto, a escolha de um valor de α^k merece alguma atenção. A solução considerada no presente trabalho é baseada na aplicação do método *Backtracking Line Search* (BLS). O método BLS consiste num esquema baseado na condição *Armijo-Goldstein* (AGC) [163]. Assim, inicia-se o processo de procura, considerando um passo elevado (α^k) para se mover ao longo da direção da pesquisa, e devolver iterativamente o passo de procura até que uma diminuição da função de custo seja observada. O processo de procura local é sistematizado em forma de pseudocódigo no algoritmo 3.2.

Algoritmo 3.2 Processo de Procura Local

```

1: function PROCURALOCAL
2:   k=0
3:   while not (Critério de Paragem) do
4:      $\mathbf{d}_k = -\mathbf{g}_k$  ▷ Onde  $g_k$  corresponde ao gradiente
5:      $\alpha_k = \min(\mathbf{x}_k - \alpha_k \cdot \mathbf{d}_k)$  ▷ Condição AGC
6:      $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \cdot \mathbf{d}_k$ 
7:      $k = k + 1$ 
8:   end while
9: end function

```

Considerar o gradiente da função para a direção de procura, o método *Steepest Descent*, embora implique alguma simplicidade em termos de processamento,

geralmente implica uma taxa de convergência reduzida, ou seja, o número de iterações necessárias para atingir o mínimo da função é elevado. Isso deve-se ao facto da direção do gradiente ser ortogonal em relação à direção do mínimo da função. Com a intenção de acelerar o procedimento de procura unidimensional, outros métodos foram propostos, nomeadamente a *Condição de Wolf* e a *Condição de Goldstein* [164] que consideram um conjunto de desigualdades para realizar uma procura exata do problema unidimensional. No presente contexto, considera-se que o aumento na complexidade não compensa a melhoria obtida, sobretudo devido ao facto de se estar próximo do ponto de procura [165].

O método de *Steepest Descent* foi um dos primeiros métodos propostos para otimização, seguindo-se vários algoritmos com propostas de melhorias, nomeadamente, o método de *Newton*, que considera a expansão da função em Série Taylor de segunda ordem, sendo, portanto, dependente da segunda derivada da função ou da matriz Hessiana, considerando um espaço multidimensional. Embora se obtenha um algoritmo com convergência mais rápida, é necessário obter informações sobre a segunda derivada da função, garantindo que a matriz Hessiana seja semi-definida positiva, de modo a que o método seja convergente [166]. Métodos para estimar a matriz Hessiana também foram propostos por vários autores, dado o seu cálculo se tornar complexo [167–170]. No contexto do presente trabalho, o uso de métodos baseados no gradiente serão um acessório para melhorar a convergência e a precisão da ferramenta principal. Assim, são integrados no método EHO, considerando-se o método de *Steepest Descent* e a condição de *Armijo-Goldstein*, com a vantagem de se tornar num processamento menos complexo.

3.4 Algoritmo Híbrido

Embora os métodos baseados no gradiente tenham sido propostos há várias décadas, estes ainda são muito usados, seja como métodos iniciais quando uma estimativa inicial pode ser facilmente fornecida, ou como um complemento para obter soluções com maior precisão, refinando uma solução pré-determinada. Vários exemplos podem ser encontrados na literatura, nomeadamente para a reconstrução de imagens

de tomografia por emissão de positrões (PET) usando gradiente conjugado [163], como na aplicação ao de tratamento de ruído de imagem [171], fusão de imagens de múltipla resolução [172], bem como vários outros exemplos que poderiam ser citados.

Em termos de fusão com outros métodos, podemos encontrar exemplos aplicados ao planeamento de trajetórias em robótica móvel [173], controlo de tensão e potência reativa em redes elétricas [174] ou sistemas de suporte à decisão para controlar veículos aéreos não tripulados [175].

Como aplicação direta em determinados campos da ciência e tecnologia, também podemos encontrar novos métodos propostos para melhorar a convergência e a precisão de determinados problemas [176–180].

Como se pode observar na figura 3.4, o presente trabalho propõe um novo algoritmo, baseado na versão original do método de EHO, com duas alterações significativas. Inicialmente, os desenvolvimentos apresentados no presente capítulo são aplicados para inicializar a população e a sua distribuição em clãs, ao invés de considerar uma geração aleatória distribuída no espaço de procura. Seguidamente, um método baseado no gradiente local é usado para acelerar a convergência, antes da aplicação do operador de atualização do clã original. No entanto, para não aumentar drasticamente o número de avaliações da função de custo, o método é aplicado apenas ao elefante matriarca de cada clã. Como o procedimento é realizado antes do operador de atualização, o benefício obtido será propagado para todos os outros elefantes através da equação 2.4. Realça-se também o facto de que, além do processo iterativo do método da procura apresentado para implementação da procura local, também o gradiente da função é calculado com o mesmo processo, segundo:

$$\begin{aligned}\frac{\partial f}{\partial x} &= \frac{f(x + h_p, y) - f(x, y)}{h_p} \\ \frac{\partial f}{\partial y} &= \frac{f(x, y + h_p) - f(x, y)}{h_p}\end{aligned}\tag{3.12}$$

No presente caso, f corresponde à função de custo e h_p deverá ser considerado um valor reduzido. Note-se que, 3.12 é obtida pela formula de Taylor, pelo que, h_p não deverá ser reduzido ao ponto de provocar instabilidade na aproximação. Tal significa

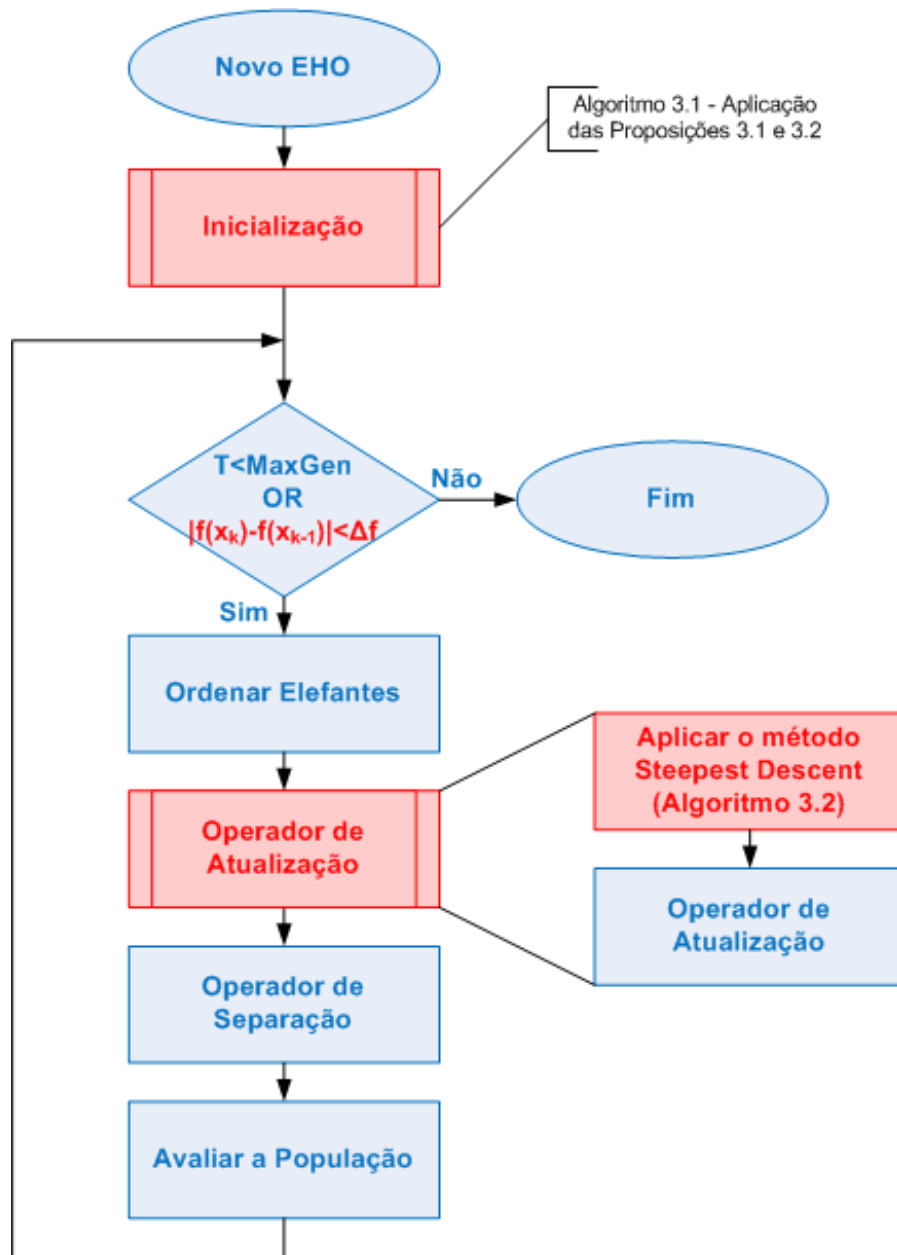


Figura 3.4: Fluxograma do algoritmo proposto perante o algoritmo original

que não existe a necessidade de realizar o cálculo das derivadas da função de custo. Para tal, utiliza-se o método das diferenças finitas [181, 182], conforme expresso em 3.12. Com base nas melhorias apresentadas, espera-se obter uma convergência mais rápida no método, ou seja, um menor número de gerações, para um erro equiparado. Essa hipótese será testada na secção seguinte, alterando os critérios de paragem para uma condição que monitoriza a evolução do algoritmo, conforme se expressa em:

$$(nAvaliações < MaxFAval) \wedge (|f(x_{k-1}) - f(x_k)| > \Delta f), \quad (3.13)$$

onde Δf é uma constante pequena e arbitrária.

3.5 Simulações e Análise de Resultados

Com o objetivo de validar o método e o seu desempenho, foram realizadas simulações computacionais, comparando:

1. O método EHO original sintonizado com parâmetros obtidos em [183], ou seja: $g_i = 1$, $P = 500$ para $i = 1, \dots, N$, $\beta_E = 2$, $\xi = 0.7$, $\alpha = 0.1$, com uma população de 120 elefantes dividida em 5 clãs, e um número máximo de avaliações da função de custo de 3000.
2. A metodologia apresentada no capítulo 3.2, referente à inicialização da população de elefantes e a sua distribuição em clãs, e a procura local ($h = 0.1$), novo método intitulado abreviadamente como iEHO (improved Elephant Herding Optimization).
3. Considerando o critério de paragem materializado pela equação 3.13, onde $\Delta f = 10^{-5}$.
4. Sem considerar o critério de paragem materializado pela equação 3.13.

Numa segunda fase, a metodologia foi aplicada, nas mesmas condições, à mais recente versão do método de EHO [184].

Para finalizar os testes de simulação e a validação das metodologias propostas, estendem-se as inovações abordadas a outro método evolutivo. O método escolhido é o algoritmo "*Coyote Optimization Algorithm*" (COA) [185] que representa um método mais recente na literatura. Este pretende ser demonstrador de que as propostas de melhoria podem ser aplicadas a qualquer método de otimização baseado em enxames de partículas, apresentando o mesmo comportamento, quer em termos de erro (RMSE), quer em termos de tempo de computação (número de iterações ou avaliações).

3.5.1 Aplicação ao Método Original

Em relação a todas as simulações realizadas, são consideradas $M_C = 10000$ simulações de *Monte Carlo*, com uma potência de ruído de $\sigma_\nu^2 = -30 \text{ dB}$ a $\sigma_\nu^2 = -5 \text{ dB}$ com incremento de 5 dB . O algoritmo SOCP [76] foi simulado, aplicando as mesmas condições apresentadas no artigo citado, consideradas aqui como o estado da arte dos métodos determinísticos. O RMSE definido em 2.6 é usado como métrica de desempenho para dissipar qualquer efeito da distribuição da fonte no espaço de pesquisa, nomeadamente, fontes localizadas fora do casco convexo dos sensores. As figuras 3.5 e 3.6 representam os resultados de simulação considerando $N = 9$, e $N = 12$ sensores, respetivamente.

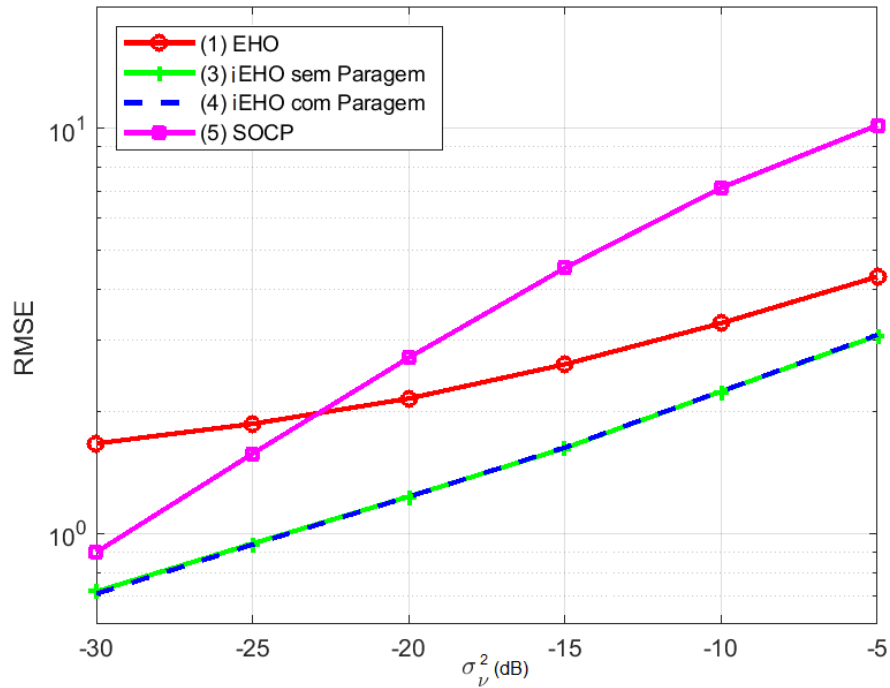


Figura 3.5: Resultados de simulação considerando $N = 9$ sensores

Como se pode observar a partir dos resultados das figuras 3.5 e 3.6, o procedimento de inicialização implica uma redução do RMSE. Embora a diminuição do erro seja mais evidente para baixos valores de ruído, onde é observada uma redução de cerca de 1 m , o método iEHO proposto oferece melhorias também para altos valores da potência de ruído. Evidencia-se que o EHO padrão implementado no capítulo 2,

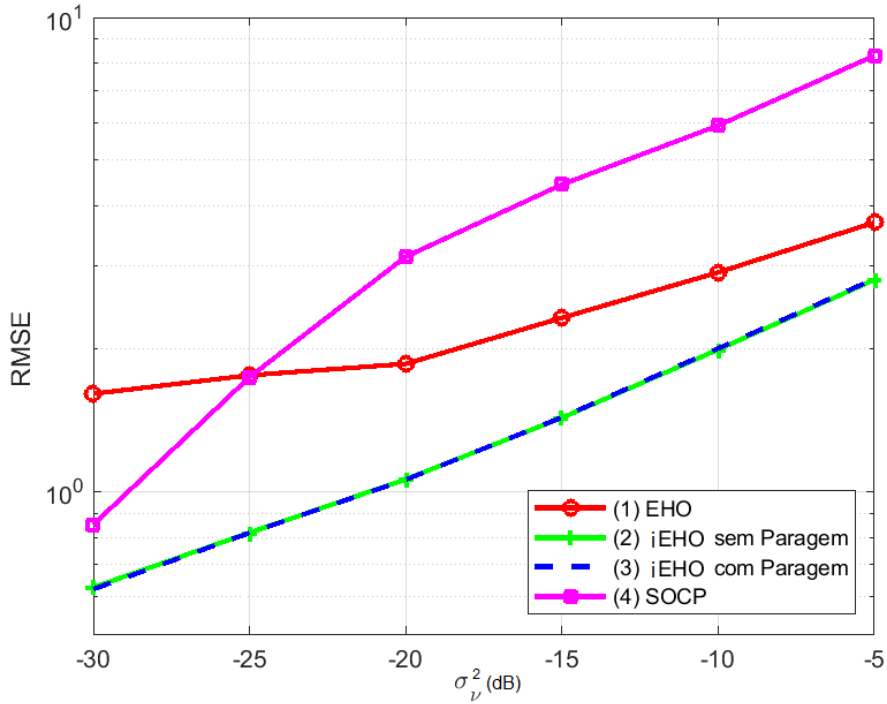


Figura 3.6: Resultados de simulação considerando $N = 12$ sensores

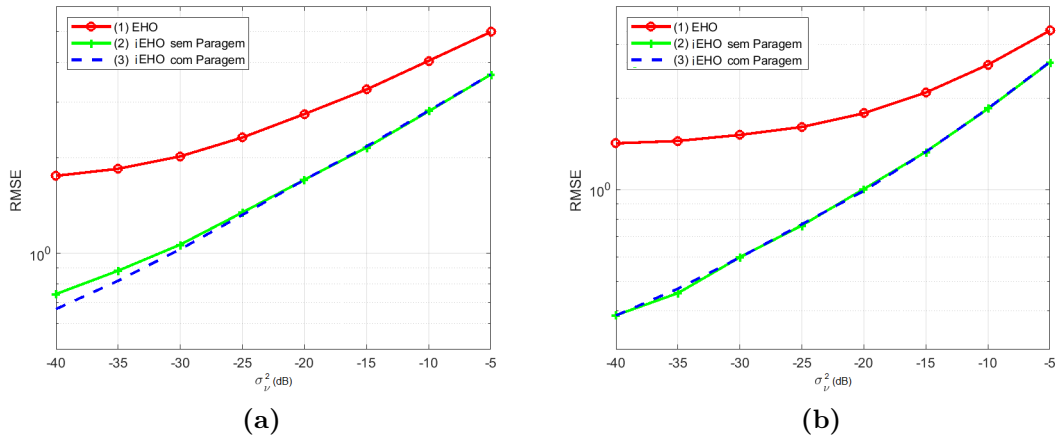


Figura 3.7: Resultados de simulação considerando: (a) $N = 6$ sensores. (b) $N = 15$ sensores

apenas superou outros métodos para valores elevados da potência de ruído e teve alguma degradação para valores mais baixos do ruído, situação que não se encontra ao executar a inicialização dos clãs. Também é possível observar que o iEHO possui apenas uma redução marginal do erro, em comparação com a sua contraparte, usando apenas a inicialização do clã, uma vez que sua principal vantagem é a melhoria da

taxa de convergência, como mencionado anteriormente. Curiosamente, embora o desempenho do EHO seja bastante eficiente em ambientes ruidosos, este apresenta deterioração em ambientes de baixa potência de ruído, conforme foi verificado no capítulo 2, onde não superou os métodos determinísticos, nomeadamente o algoritmo SOCP. Este resultado leva-nos a estudar abordagens alternativas que complementariam o seu desempenho, tanto em termos de precisão como de taxa de convergência, com os esquemas de inicialização *inteligente* e pesquisa local propostos. Nas figuras 3.5 e 3.6, pode observar-se que esses esquemas permitiram uma redução do erro para baixo nível de potência do ruído, que é mantida (com uma margem diminuta) por toda a extensão considerada dessa potência.

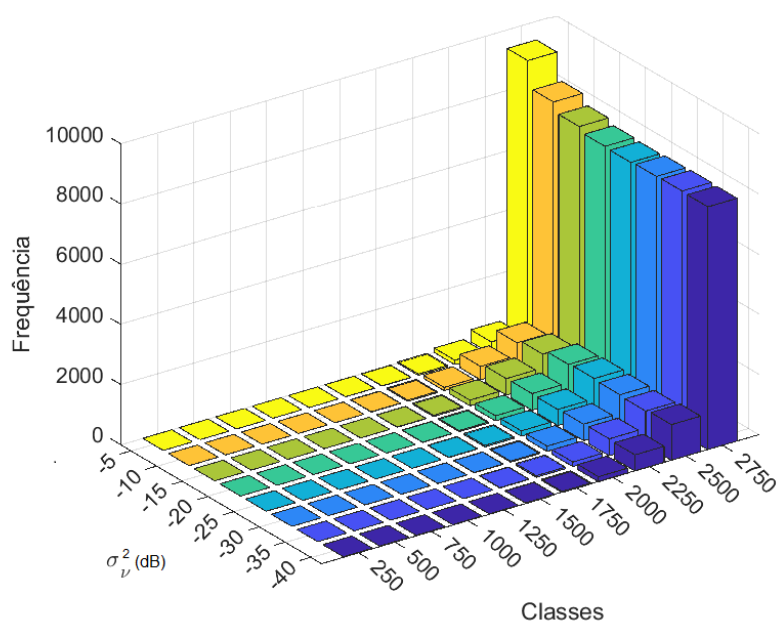


Figura 3.8: Histograma do número de avaliações para o algoritmo EHO ($N = 9$) com critério de paragem

Outra característica importante a destacar é o facto de o iEHO obter o mesmo desempenho em termos de erro (RMSE) com e sem a implementação do critério de paragem (equação 3.13). Este facto demonstra que o algoritmo iEHO converge antes que o número máximo de avaliações de funções seja alcançado. A figura 3.7 estende as simulações, considerando diferentes configurações em termos do número de sensores. A figura 3.7a demonstra o mesmo comportamento das curvas

em relação ao método original, ao novo iEHO com, e sem considerar o critério de paragem, quando apenas integrados $N = 6$ sensores na rede. A figura 3.7b apresenta resultados quando incluídos $N = 15$ sensores na rede.

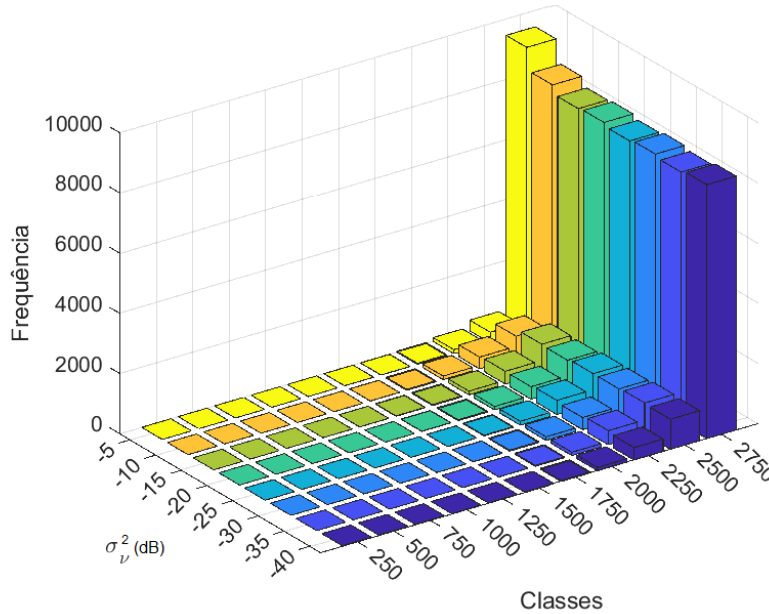


Figura 3.9: Histograma do número de avaliações para o algoritmo EHO ($N = 12$) com critério de paragem

Em relação ao comportamento dos métodos em termos de convergência, foram realizadas mais simulações, aplicando os mesmos critérios de paragem ao EHO padrão. Os resultados são apresentados nas figuras 3.8 a 3.11 na forma de histogramas, com o número de avaliações da função de custo para diferentes potências de ruído. Os histogramas mostram a comparação do método padrão com o método melhorado, em termos do número de avaliações de funções. Como pode ser visto pelos resultados, independentemente do facto de um critério de paragem ter sido adicionado ao algoritmo EHO padrão, ele exigiu o número máximo de gerações disponíveis na maioria das vezes e parou somente quando esse limite foi atingido, independentemente do critério de paragem. Em contraste, pode observar-se que o algoritmo melhorado requer um número muito menor de iterações, e o número máximo de avaliações nunca foi atingido. Portanto, os resultados das simulações corroboram a eficácia dos dois esquemas propostos (inicialização e

3. Desenvolvimento de Metodologias para o Aperfeiçoamento de Algoritmos Meta-Heurísticos

melhoria), indicando que o novo algoritmo iEHO beneficia de maior precisão e de uma taxa de convergência mais rápida, em comparação com o equivalente EHO original.

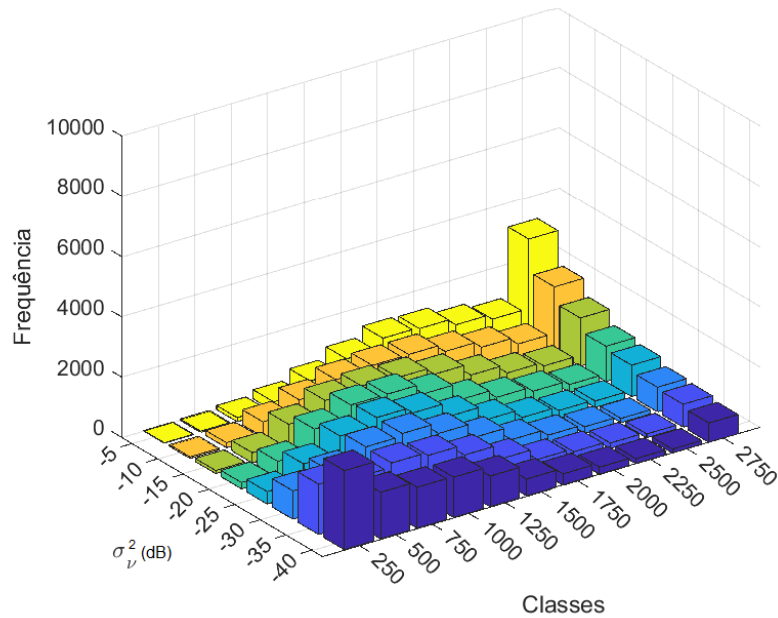


Figura 3.10: Histograma do número de avaliações para o algoritmo iEHO ($N = 9$)

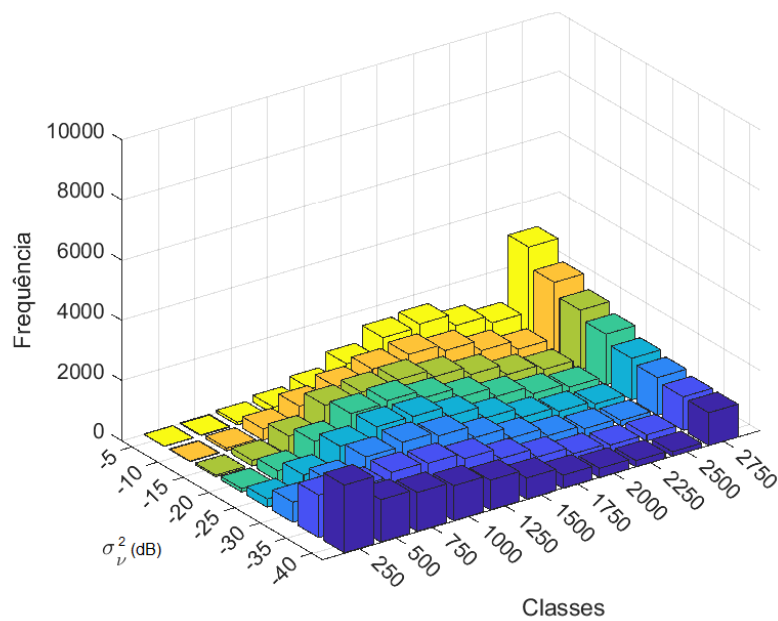


Figura 3.11: Histograma do número de avaliações para o algoritmo iEHO ($N = 12$)

As figuras 3.12 e 3.13 estendem a metodologia para $N = 6$ e $N = 15$ respetivamente.

Como se pode verificar, tanto a diminuição de sensores como o seu acréscimo, mantiveram o comportamento já observado. A metodologia proposta, quando aplicada a uma rede de sensores com outras dimensões, permite obter uma redução do erro ou um valor próximo, reduzindo significativamente o número de avaliações necessárias para obtenção desse resultado.

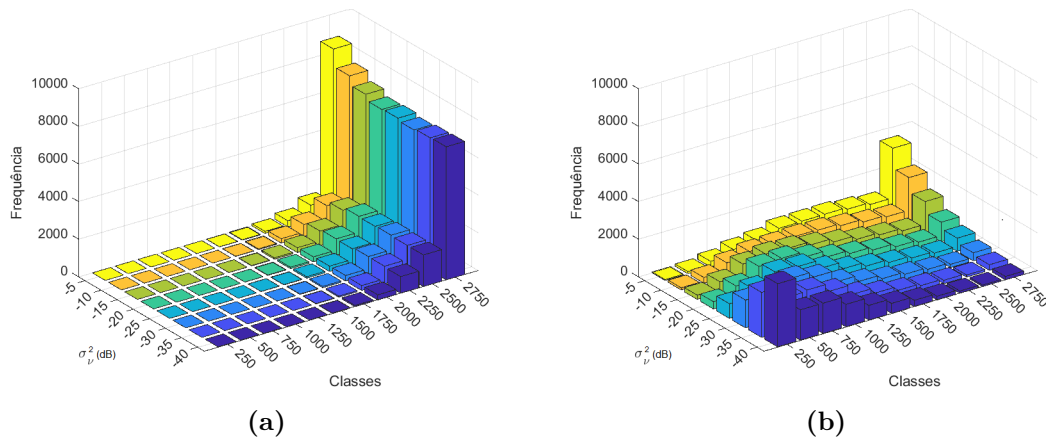


Figura 3.12: Histograma do número de avaliações para $N = 6$. (a) Algoritmo EHO (b) Novo Algoritmo (iEHO)

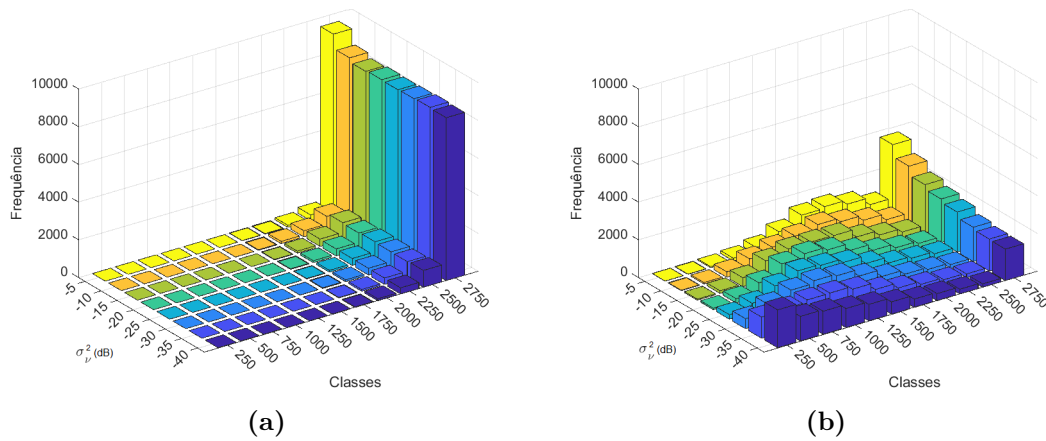


Figura 3.13: Histograma do número de avaliações para $N = 15$. (a) Algoritmo EHO (b) Novo Algoritmo (iEHO)

No Anexo B apresentam-se histogramas detalhados, juntamente com os dados estatísticos considerados mais relevantes. No sentido de demonstrar a versatilidade do método e a sua validação com outros conjuntos do número de sensores, o Anexo

B estende as simulações para $N = 6$ e $N = 15$. As tabelas 3.1 a 3.8 resumem os indicadores calculados em relação ao número de avaliações da função de custo para os métodos EHO e iEHO nas Mc simulações. Consideram-se como indicadores o valor mínimo (N_{min}) e máximo (N_{max}), o primeiro (Q_1) e terceiro (Q_3) quartil, a mediana (ou segundo quartil) (\tilde{N}), a média (\bar{N}) e o desvio padrão da mesma (σ_N).

Através da análise estatística dos resultados obtidos em termos do número de avaliações da função de custo, retiram-se as seguintes conclusões:

1. O método iEHO consegue gerar soluções, executando apenas uma geração da população, portanto, apenas baseadas no método de inicialização.
2. Considerando o número máximo de avaliações, 25% das simulações apresentam valores reduzidos no método iEHO (tipicamente 1/3 do valor máximo), enquanto no caso EHO este valor corresponde maioritariamente ao máximo de avaliações.
3. Quando consideradas 75% das simulações (terceiro quartil), o método iEHO não atinge o número máximo de avaliações, situação oposta em relação ao método original.
4. No caso iEHO, a mediana apresenta-se sempre bastante próxima da média, o que indica uma distribuição uniforme em torno da mesma. Semelhante avaliação pode ser efetuada no caso EHO, no entanto, a média e a mediana estão próximas do valor máximo. Tal situação já tinha sido observada nos histogramas.
5. Em relação à média, no caso EHO, esta assume valores elevados com um desvio padrão reduzido. Tal significa, que a distribuição está concentrada no valor máximo. Em relação ao novo iEHO, a média é sempre reduzida, embora com maior dispersão. Tendo em conta que a mesma é bastante reduzida, a dispersão mostra que as diversas simulações assumem maioritariamente valores mais elevados.

Tabela 3.1: Indicadores estatísticos ($\sigma_v^2 = -40$ dB)

	N=6		N=9		N=12		N=15	
	iEHO	EHO	iEHO	EHO	iEHO	EHO	iEHO	EHO
N_{\min}	120	1080	120	1320	120	1200	120	1080
N_{\max}	3000	3000	3000	3000	3000	3000	3000	3000
Q3	1080	3000	1320	3000	1560	3000	1800	3000
Q1	240	2520	360	2640	480	2760	480	2880
\tilde{N}	720	2880	840	3000	960	3000	960	3000
$\sigma_{\tilde{N}}$	605	305	751	269	851	249	943	238
\bar{N}	774	2736	965	2826	1117	2854	1268	2872

Tabela 3.2: Indicadores estatísticos ($\sigma_v^2 = -35$ dB)

	N=6		N=9		N=12		N=15	
	iEHO	EHO	iEHO	EHO	iEHO	EHO	iEHO	EHO
N_{\min}	120	1200	120	720	120	1200	120	1320
N_{\max}	3000	3000	3000	3000	3000	3000	3000	3000
Q3	1200	3000	1320	3000	1560	3000	1920	3000
Q1	480	2520	600	2640	600	2760	720	2880
\tilde{N}	720	2880	960	3000	1080	3000	1200	3000
$\sigma_{\tilde{N}}$	638	305	739	274	813	254	890	234
\bar{N}	897	2739	1076	2825	1219	2852	1371	2875

Tabela 3.3: Indicadores estatísticos ($\sigma_v^2 = -30$ dB)

	N=6		N=9		N=12		N=15	
	iEHO	EHO	iEHO	EHO	iEHO	EHO	iEHO	EHO
N_{\min}	120	1320	120	1200	120	1440	120	1320
N_{\max}	3000	3000	3000	3000	3000	3000	3000	3000
Q3	1320	3000	1440	3000	1680	3000	1920	3000
Q1	600	2520	720	2640	840	2760	840	2760
\tilde{N}	980	2880	1080	3000	1200	3000	1320	3000
$\sigma_{\tilde{N}}$	665	310	713	271	780	249	847	239
\bar{N}	1046	2732	1195	2823	1352	2856	1485	2870

Tabela 3.4: Indicadores estatísticos ($\sigma_v^2 = -25$ dB)

	N=6		N=9		N=12		N=15	
	iEHO	EHO	iEHO	EHO	iEHO	EHO	iEHO	EHO
N_{\min}	120	1200	120	600	120	1440	120	1320
N_{\max}	3000	3000	3000	3000	3000	3000	3000	3000
Q3	1560	3000	1680	3000	1800	3000	2238	3000
Q1	720	2520	840	2640	960	2760	960	2760
\tilde{N}	1080	2880	1200	3000	1320	3000	1440	3000
$\sigma_{\tilde{N}}$	109	309	721	275	753	253	822	237
\bar{N}	1236	2742	1343	2821	1477	2852	1647	2870

3. Desenvolvimento de Metodologias para o Aperfeiçoamento de Algoritmos Meta-Heurísticos

Tabela 3.5: Indicadores estatísticos ($\sigma_v^2 = -20$ dB)

	N=6		N=9		N=12		N=15	
	iEHO	EHO	iEHO	EHO	iEHO	EHO	iEHO	EHO
N_{\min}	120	1080	120	1200	120	1080	120	1320
N_{\max}	3000	3000	3000	3000	3000	3000	3000	3000
Q3	1800	3000	1800	3000	2040	3000	2400	3000
Q1	840	2520	960	2640	1080	2760	1200	2760
\bar{N}	1200	2880	1320	3000	1440	3000	1560	3000
$\sigma_{\bar{N}}$	735	310	709	274	744	250	777	240
\bar{N}	1419	2741	1474	2822	1629	2858	1769	2871

Tabela 3.6: Indicadores estatísticos ($\sigma_v^2 = -15$ dB)

	N=6		N=9		N=12		N=15	
	iEHO	EHO	iEHO	EHO	iEHO	EHO	iEHO	EHO
N_{\min}	120	1080	120	1200	120	960	120	1080
N_{\max}	3000	3000	3000	3000	3000	3000	3000	3000
Q3	2160	3000	2040	3000	2280	3000	2640	3000
Q1	960	2520	1080	2760	1200	2760	1320	2880
\bar{N}	1440	2880	1560	3000	1560	3000	1800	3000
$\sigma_{\bar{N}}$	774	304	725	272	727	254	751	242
\bar{N}	1608	2762	1654	2832	1764	2859	1895	2869

Tabela 3.7: Indicadores estatísticos ($\sigma_v^2 = -10$ dB)

	N=6		N=9		N=12		N=15	
	iEHO	EHO	iEHO	EHO	iEHO	EHO	iEHO	EHO
N_{\min}	120	1080	120	1440	120	1320	120	840
N_{\max}	3000	3000	3000	3000	3000	3000	3000	3000
Q3	2520	3000	2400	3000	2520	3000	2880	3000
Q1	1200	2640	1320	2760	1440	2880	1560	2880
\bar{N}	1800	3000	1800	3000	1800	3000	2040	3000
$\sigma_{\bar{N}}$	773	285	709	250	695	239	706	235
\bar{N}	1866	2799	1875	2861	1948	2879	2066	1885

Tabela 3.8: Indicadores estatísticos ($\sigma_v^2 = -5$ dB)

	N=6		N=9		N=12		N=15	
	iEHO	EHO	iEHO	EHO	iEHO	EHO	iEHO	EHO
N_{\min}	120	1080	120	1440	120	1440	120	1440
N_{\max}	3000	3000	3000	3000	3000	3000	3000	3000
Q3	2880	3000	2880	3000	2760	3000	3000	3000
Q1	1560	2880	1680	3000	1680	3000	1800	3000
\bar{N}	2160	3000	2160	3000	2160	3000	2280	3000
$\sigma_{\bar{N}}$	718	234	637	172	603	161	595	156
\bar{N}	2133	2874	2173	2935	2189	2945	2282	2948

3.5.2 Aplicação à mais Recente Versão do Método EHO

Sendo o processo de disseminação científica bastante dinâmico e globalizado, uma preocupação constante deverá ser a monitorização dos avanços publicados na área de investigação. No caso da presente dissertação, para além dos métodos baseados em enxames de partículas que foram sendo publicados e atualizados no estado da arte, foi prestada especial atenção ao método EHO.

Uma metodologia para a inicialização da população, tendo em conta sequências caóticas [186], foi apresentada em [187]. A metodologia que atua na diversificação da população inicial gerada (com dois mapas propostos) apresenta resultados de simulação em diversas funções de teste comparados com o algoritmo EHO original [84] e o método PSO [54]. Pela análise dos resultados, verifica-se a superioridade em relação ao algoritmo de PSO, mas em relação ao algoritmo EHO original, as melhorias não são globais, sendo divididas pelos dois mapas propostos.

O desempenho do método de EHO aplicado a redes WSN foi comparado com uma versão adaptada do algoritmo "Tree Growth Algorithm" (TGA) [188, 189] em [88]. O método de EHO é atualizado, considerando-se que o nível de exploração não é suficiente e pode levar a soluções sub-ótimas. A proposta de melhoria consiste em considerar dinâmico o número de elefantes que abandonam os clãs, sendo atualizado com o número de iterações. O algoritmo apresenta melhorias na maioria das funções de teste consideradas, embora não na sua totalidade. O mesmo acontece quando aplicado ao problema concreto das WSNs onde, do ponto de vista do tempo de computação, não se apresenta como a melhor solução.

No trabalho [190], são propostas várias estratégias de atualização da população, nas quais um, dois, ou três indivíduos são selecionados a partir das iterações anteriores, e a sua informação é incorporada através de uma média ponderada. Os pesos são determinados aleatoriamente, tendo em conta a avaliação da função de custo dos elefantes na iteração anterior. O método é apresentado com seis variantes diferentes, dependentes da quantidade de informação prévia das iterações que é incorporada e aplicada a várias funções de teste. Embora os resultados apresentados configurem melhorias, as mesmas são distribuídas pelas várias variantes, não sendo

possível unificar a metodologia. Além disso, determinadas funções não apresentam resultados melhorados em relação ao método original, sobretudo com um custo de maior complexidade de computação.

Os parâmetros de sintonização do método são abordados em [191]. Neste caso, é considerado que o parâmetro α (ver método original) deve ser ajustado durante o processo iterativo de forma linear. Também ao nível da inicialização da população, é considerado que o método não deve iniciar até que atinja ou exceda um limiar predeterminado [192]. Por último, é apresentada uma proposta de melhoria ao nível da regeneração dos elefantes com pior desempenho. À semelhança de resultados obtidos com outros métodos, as simulação mostram melhorias divididas entre as várias abordagens, tendo em conta que são analisadas separadamente.

Proposta recentemente, uma versão melhorada do algoritmo, "Enhanced Elephant Herding Optimization" (EEHO), apresenta os dois novos operadores para a atualização da população [184]. As reivindicações do método têm por base uma análise vetorial em relação ao movimento que os novos elefantes devem assumir no espaço de procura (Figura 3.14).

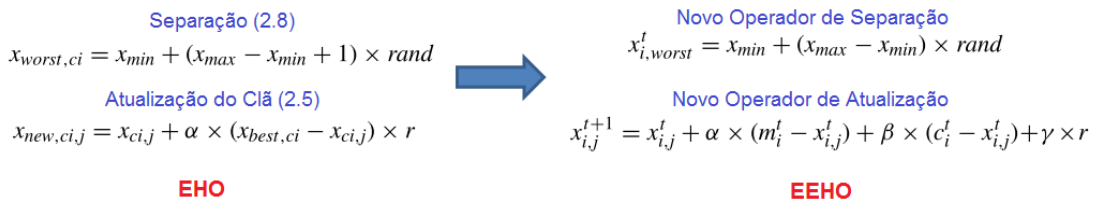


Figura 3.14: Novos operadores do algoritmo EEHO [184]

Sendo esta a mais recente versão, será aqui considerada para integração das metodologias propostas no presente trabalho. Os parâmetros considerados para simulação serão os mesmos utilizados para validação do método de EHO ($\alpha = 0.7$ e $\beta = 0.1$). Em relação ao novo parâmetro γ , que atua como fator de escala controlando o grau de aleatoriedade na geração dos clãs, será considerado o valor proposto em [88] ($\gamma = 0.015$), e que apresenta o melhor desempenho. As figuras 3.15 e 3.16 mostram os resultados relativos ao erro (RMSE), comparando o método SOCP (1), a versão EEHO original (2), a aplicação do critério de paragem ao

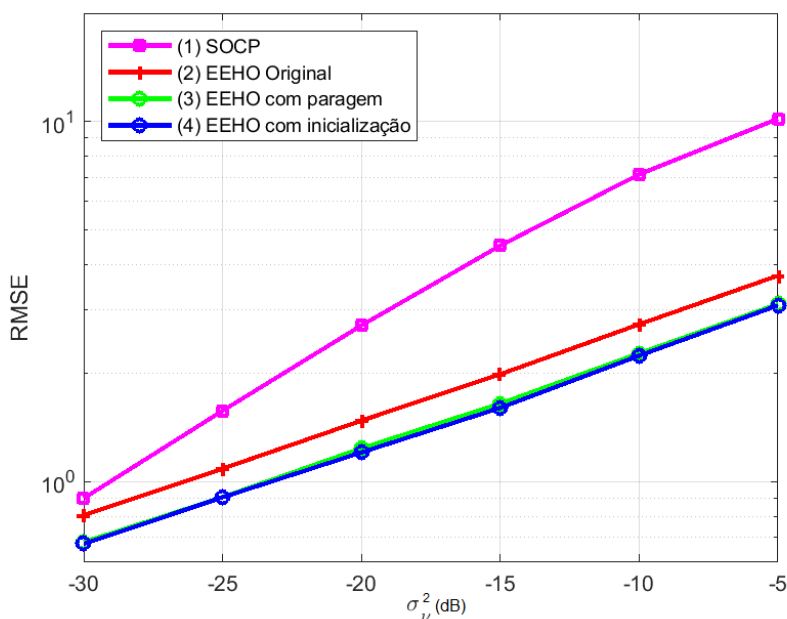


Figura 3.15: Resultados de simulação para o algoritmo EEHO ($N = 9$ sensores)

método EEHO (3) e a aplicação da inicialização inteligente da população (capítulo 3) no algoritmo de EEHO (4), considerando $N = 9$ e $N = 12$, respetivamente. Como se pode observar, o comportamento do método mantém a tendência já verificada anteriormente em relação ao método original EHO. Mais precisamente, verificam-se as duas contribuições já reivindicadas. Por um lado, a redução do erro, por outro, o facto de que a utilização do critério de paragem não influenciar o erro de estimação.

Para demonstrar a aceleração da convergência espectável, procede-se à análise dos histogramas do número de avaliações da função de custo.

As figuras 3.17 a 3.20 demonstram a extensão da análise efetuada anteriormente, agora aplicada ao método EEHO. Verifica-se que a inclusão do critério de paragem mantém o método com um nível de convergência semelhante à versão original, ou seja, para a maioria das simulações de *Monte Carlo*, é executado o número máximo de avaliações parametrizado. No entanto, quando aplicada a metodologia de inicialização da população, o número de avaliações diminui significativamente. Realça-se que o erro de estimação é mantido nos mesmos níveis (figuras 3.15 e 3.16).

3. Desenvolvimento de Metodologias para o Aperfeiçoamento de Algoritmos Meta-Heurísticos

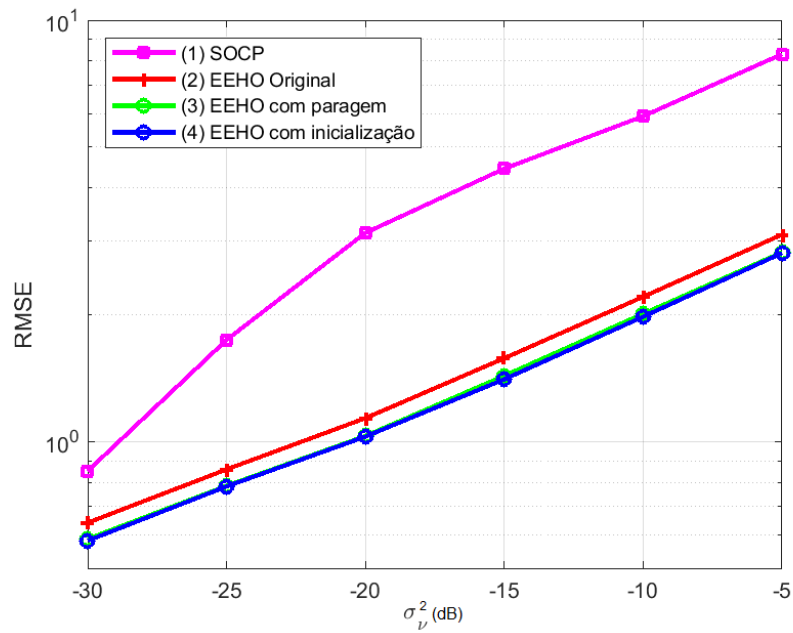


Figura 3.16: Resultados de simulação para o algoritmo EEHO ($N = 12$ sensores)

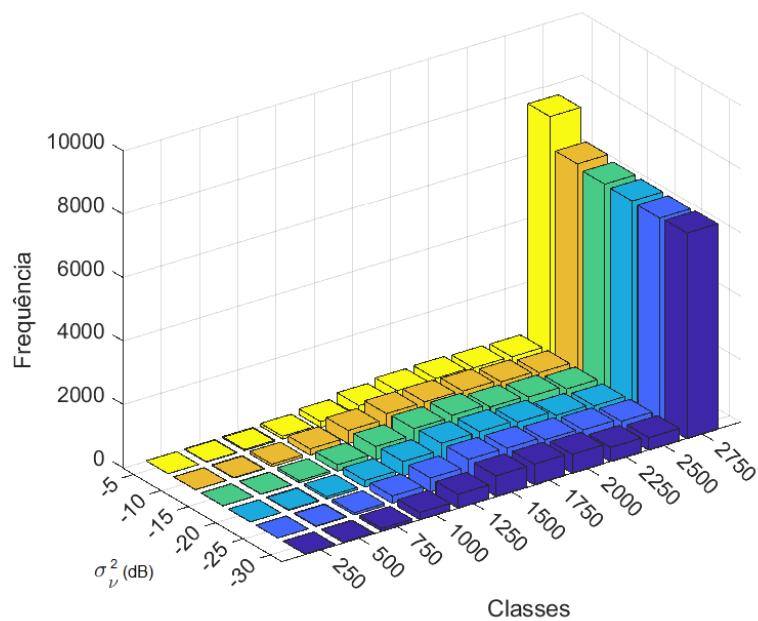


Figura 3.17: Histograma do número de avaliações para o algoritmo EEHO ($N = 9$) com critério de paragem

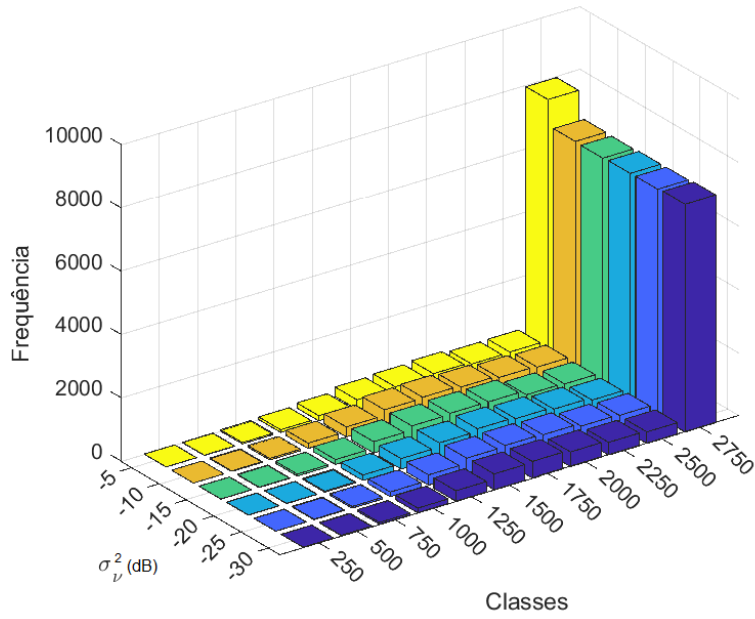


Figura 3.18: Histograma do número de avaliações para o algoritmo EEHO ($N = 12$) com critério de paragem

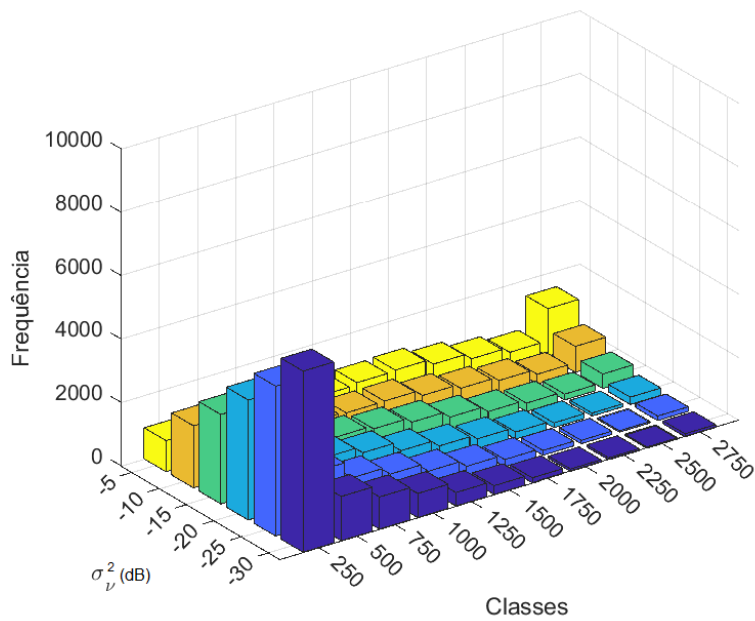


Figura 3.19: Histograma do número de avaliações para o algoritmo EEHO ($N = 9$) com inicialização de população

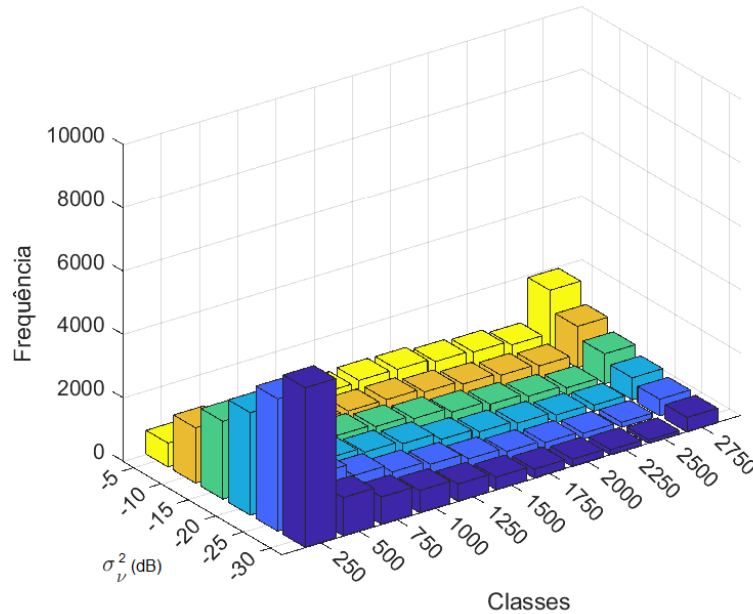


Figura 3.20: Histograma do número de avaliações para o algoritmo EEHO ($N = 12$) com inicialização de população

3.5.3 Aplicação ao Método "*Coyote Optimization Algorithm*"

O algoritmo *Coyote Optimization Algorithm* (COA), baseado no comportamento da espécie "*Canis latrans*", consiste numa formulação matemática para representar o modo como o coioete procura alimento e sobrevive na natureza. Embora recentemente proposto na literatura, foi introduzido para a resolução de vários problemas no contexto energético [193–195]. O algoritmo parte de uma população inicializada de modo aleatório, organizada em N_p grupos de N_c coioetes, cada um sob a liderança de um coioete dominante. A geração e atualização da população tem por base a interação social nos grupos e entre grupos dos animais, definindo métricas para esta avaliação, assim como a idade, conceito utilizado para eliminar coioetes com menor desempenho com base na avaliação da função de custo [185]. Note-se que do ponto de vista conceptual, a abordagem à gestão da população, a sua manutenção e atualização, é semelhante ao método de EHO. Do ponto de vista de simulação, foram considerados os parâmetros propostos em [185] em relação aos testes de

desempenho, nomeadamente, $N_p = 20$ grupos, com $N_c = 5$ coiotes por grupo e um número máximo de avaliações de 20000.

As condições de simulação são mantidas, sendo consideradas $M_C = 10000$ simulações de *Monte Carlo*, potência do ruído de $\sigma_v^2 = -30$ dB a $\sigma_v^2 = -5$ dB, com incremento de 5 dB. Em relação ao modelo acústico, considera-se $P = 500$, $g_i = 1$, um espaço de procura com dimensões $100\text{ m} \times 100\text{ m}$ e dois cenários com $N = 9$ e $N = 12$ sensores.

As figuras 3.21 e 3.22 demonstram o mesmo comportamento do método em relação aos restantes no que diz respeito ao critério de paragem e à inicialização inteligente da população. Deve, no entanto, referir-se que a implementação original do método apresenta resultados ao nível do erro superior face ao algoritmo de EHO, mesmo tendo sido considerado um valor bastante elevado do número máximo de avaliações.

Ao nível dos histogramas (figuras 3.23 a 3.26), observa-se o comportamento esperado, onde a inicialização da população implica uma forte redução do número de avaliações da função de custo. Realça-se o facto de as classes apresentarem uma escala ajustada ao máximo das 20000 avaliações. Mais uma vez, a implementação das metodologias propostas validam a melhoria significativa do método original, em termos de convergência para todos os níveis da potência de ruído. Note-se que o algoritmo termina nas primeiras gerações de população, mantendo o erro de estimação ao nível da localização. Neste caso específico, o erro é superior aos métodos de otimização já testados, no entanto, do ponto de vista de convergência, caso se estivesse perante uma imposição de aplicação deste método, integrar a inicialização inteligente da população implicaria manter o seu desempenho inicial em termos de erro, mas com um nível de computação significativamente mais eficiente.

3. Desenvolvimento de Metodologias para o Aperfeiçoamento de Algoritmos Meta-Heurísticos

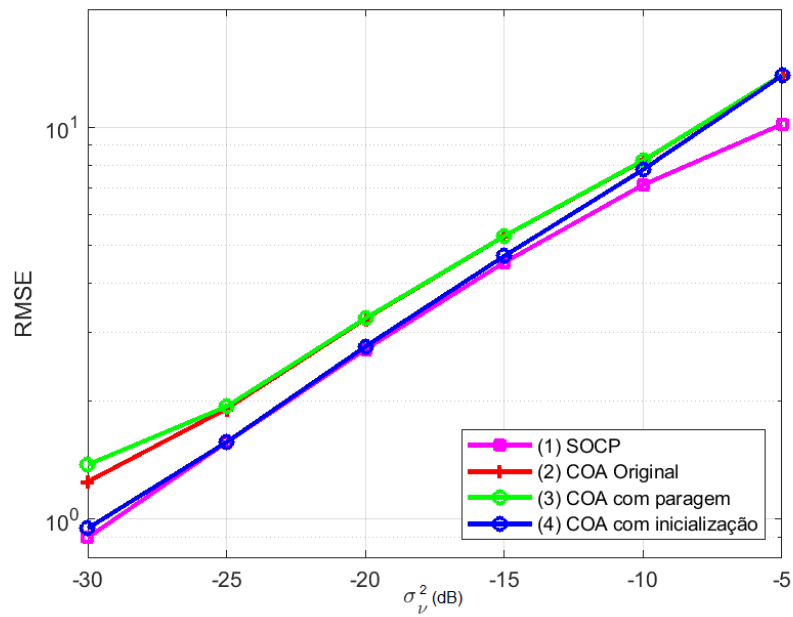


Figura 3.21: Resultados de simulação para o algoritmo COA ($N = 9$ sensores)

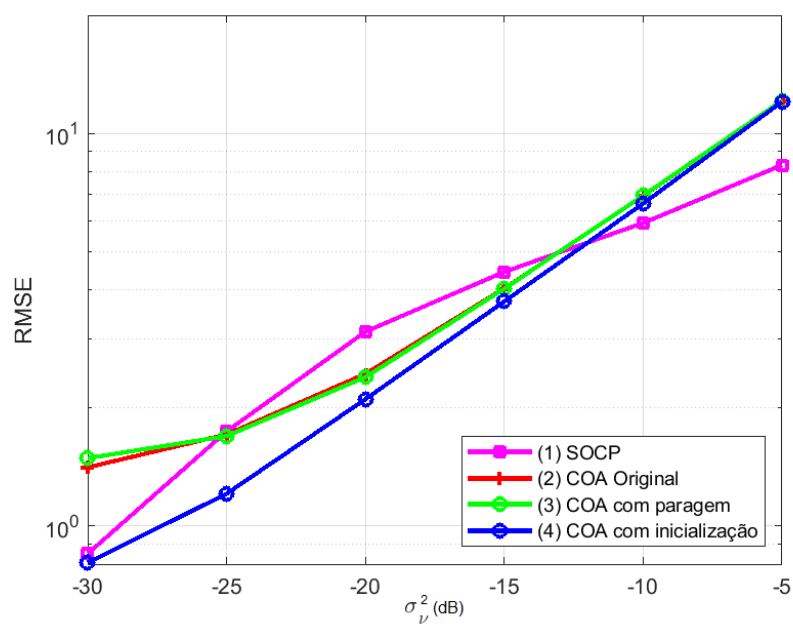


Figura 3.22: Resultados de simulação para o algoritmo COA ($N = 12$ sensores)

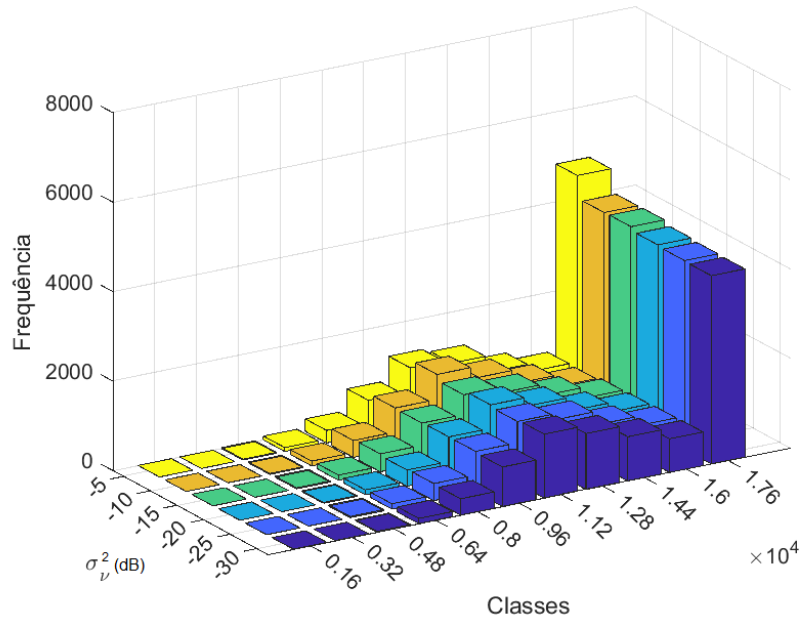


Figura 3.23: Histograma do número de avaliações para o algoritmo COA ($N = 9$) com critério de paragem

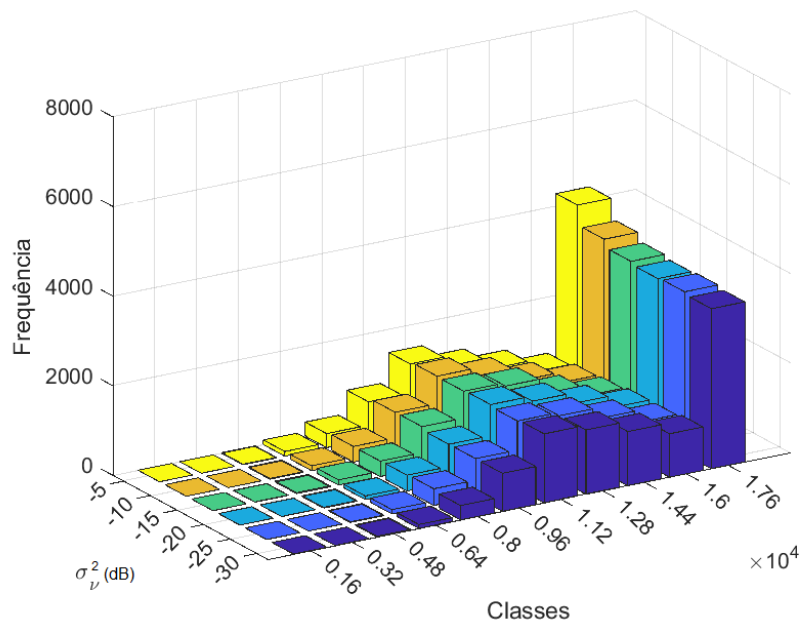


Figura 3.24: Histograma do número de avaliações para o algoritmo COA ($N = 12$) com critério de paragem

3. Desenvolvimento de Metodologias para o Aperfeiçoamento de Algoritmos Meta-Heurísticos

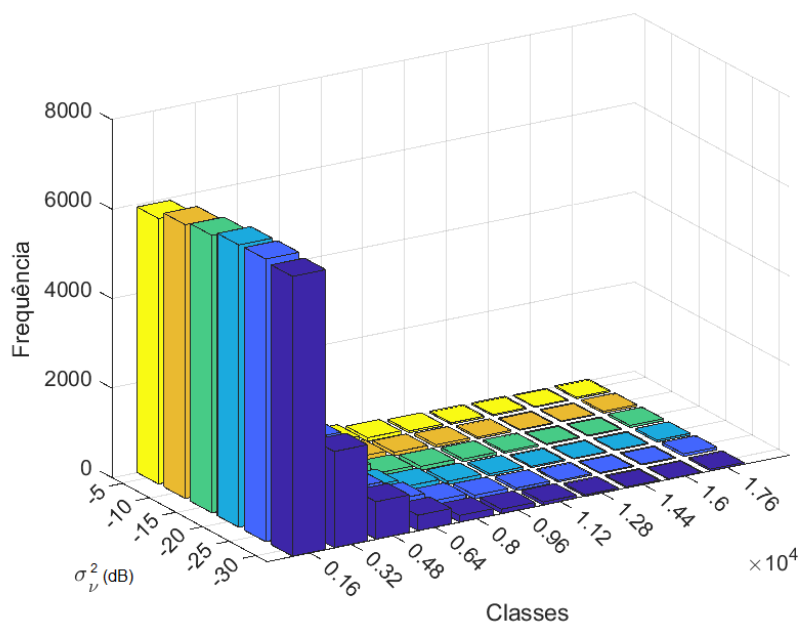


Figura 3.25: Histograma do número de avaliações para o algoritmo COA ($N = 9$) com inicialização de população

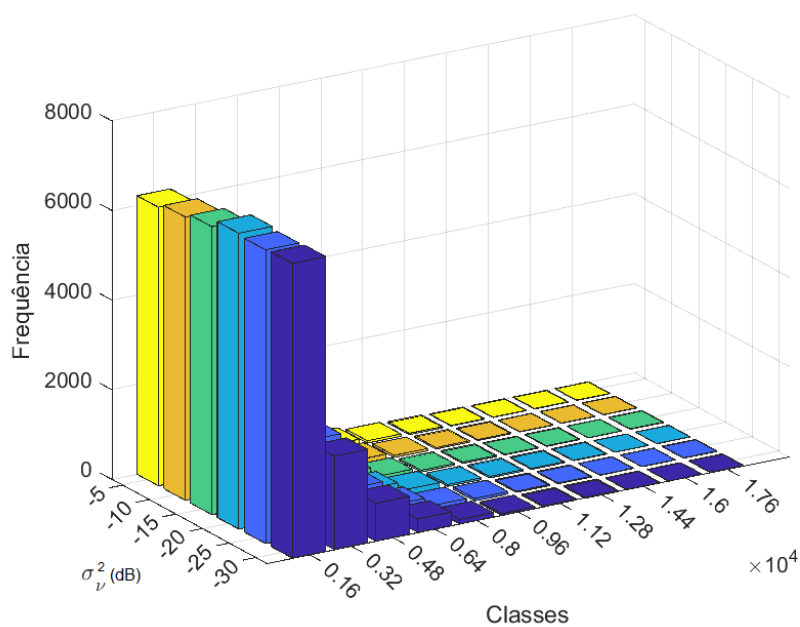


Figura 3.26: Histograma do número de avaliações para o algoritmo COA ($N = 12$) com inicialização de população

3.6 Conclusões

No presente capítulo, duas contribuições foram desenvolvidas para melhorar o desempenho do algoritmo original de EHO, quando aplicado ao problema de localização baseado na medição de energia acústica. A metodologia proposta assume particularidades do problema em análise, nomeadamente a estimativa da distância obtida através das observações do modelo.

Inicialmente, foi introduzida uma metodologia para a inicialização da população e sua organização em clãs, considerando a estimativa da distância entre a fonte acústica e os sensores. Através de simulações computacionais, foi demonstrado que a metodologia proposta resulta numa maior precisão para níveis de potência de ruído elevado, onde outros métodos tendem a falhar.

Em segundo lugar, uma versão discreta do método *Steepest Descent* baseado em diferenças finitas foi incorporada no operador de atualização dos clãs, o que contribuiu para obter uma taxa de convergência mais elevada. Os resultados de simulação validaram o desempenho dos métodos propostos, permitindo à nova metodologia iEHO diminuir o erro de localização em aproximadamente 1 *m* para níveis de ruído reduzidos, enquanto se manteve uma correspondência em relação ao desempenho do EHO original em ambientes ruidosos. No entanto, o último resultado foi alcançado com um número significativamente menor de gerações de clãs, o que torna o iEHO mais adequado para aplicações em ambientes de tempo real, com recursos limitados de energia.

Com o propósito de estender as conclusões e demonstrar a viabilidade da metodologia de um ponto de vista mais genérico, estas foram integradas e implementadas em dois novos métodos. Os resultados de simulação permitem afirmar que a metodologia teve o mesmo efeito nas novas implementações, nomeadamente a manutenção ou redução do erro de localização e, um decréscimo significativo em termos do número de avaliações necessárias da função de custo. O facto de o número de avaliações da função de custo ser substancialmente reduzido, permite afirmar que a taxa de convergência é acelerada e, portanto, o nível de computação necessário à obtenção dos resultados será reduzido em relação às versões originais dos métodos.

3. Desenvolvimento de Metodologias para o Aperfeiçoamento de Algoritmos Meta-Heurísticos

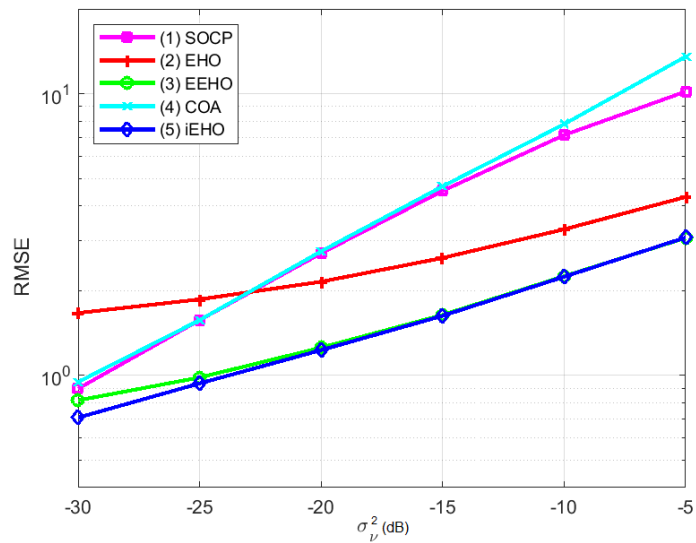


Figura 3.27: Resumo dos algoritmos ao nível do erro ($N = 9$ sensores)

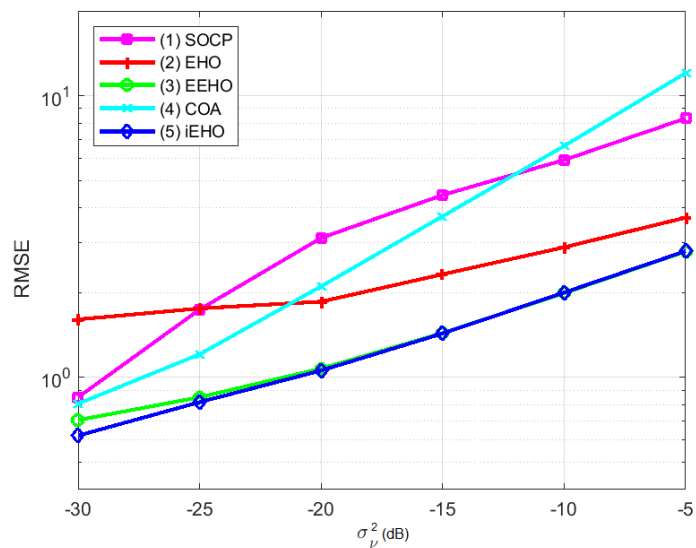


Figura 3.28: Resumo dos algoritmos ao nível do erro ($N = 12$ sensores)

As figuras 3.27 ($N = 9$) e 3.28 ($N = 12$) resumem os resultados obtidos das implementações dos vários métodos, permitindo validar o comportamento do erro e o efeito da integração da metodologia proposta entre os algoritmos.

Questões sob Investigação: Perante os resultados obtidos no capítulo 3, tendo sido proposto uma sub-região no espaço para inicialização da população, esta é gerada de uma modo aleatório. De que forma se podem entender os resultados e obter um população globalmente gerada a partir da informação obtida do modelo? Qual o efeito produzido em termos de erro? Qual o efeito produzido em termos de convergência?

4

Inicialização Inteligente da População pelo Método de *Monte Carlo*

Conteúdos

4.1	Enquadramento e Objetivos	85
4.2	Geração do Estado Inicial da Cadeia de Markov	86
4.3	Cadeias de Markov a partir do Algoritmo de Metropolis-Hastings	88
4.4	Resultados e Discussão	95
4.5	Conclusão	97

4.1 Enquadramento e Objetivos

No capítulo 3 foi apresentada uma metodologia para a inicialização da população, dividida em *clãs*, do algoritmo EHO baseada numa abordagem geométrica. Depois de determinados pontos de interceção relativos à estimação de três medidas de distância, os *elefantes* foram gerados aleatoriamente, numa sub-região do espaço delimitada por esses mesmos pontos. No presente capítulo, será realizada uma extensão da referida metodologia, segundo a qual, a população relativa a cada clã será gerada a partir de uma sequência de *Monte Carlo* baseada em cadeias de *Markov*. O início das sequências serão constituídas pelos centros dos clãs, obtidos

pelo método de inicialização da população proposto no capítulo 3. A função de densidade de probabilidade (PDF) do modelo será considerada para a geração da cadeia. Para tal, será aplicado o algoritmo de *Metropolis-Hastings* [196, 197].

Com a presente metodologia, pretende obter-se uma distribuição dos elefantes em torno da perspectivada localização, objeto de estimação. Perspetiva-se, tendo em conta a estreita relação com o modelo, a aceleração da convergência do método, e um menor erro em relação à estimativa determinada.

4.2 Geração do Estado Inicial da Cadeia de Markov

Numa primeira fase da metodologia desenvolvida, à semelhança do exposto no capítulo 3, a equação 3.1 será considerada para obtenção de uma estimativa de distância.

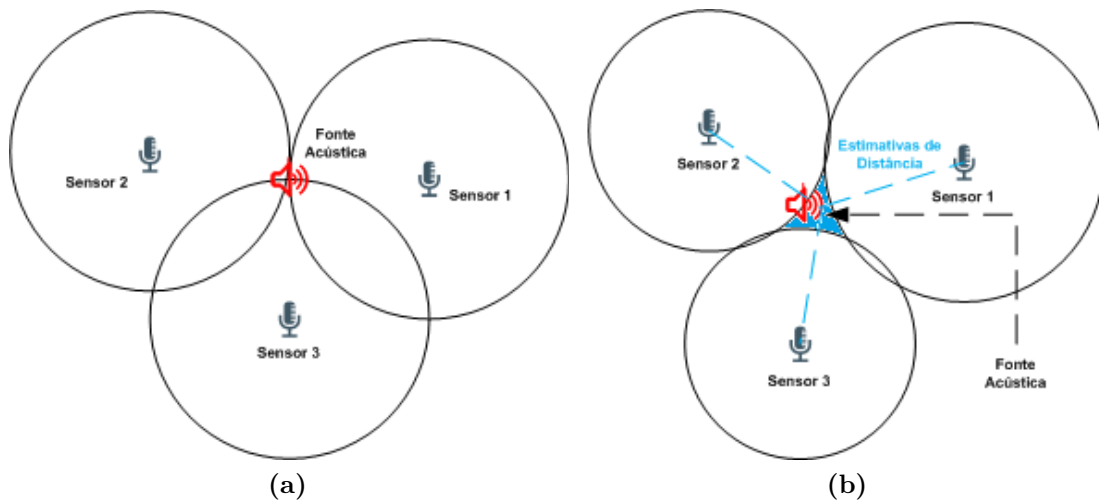


Figura 4.1: Solução do problema de localização. (a) Situação ideal sem ruído nas medições (b) Situação real

A solução do problema de localização consiste na determinação do ponto de interceção dos raios obtidos a partir da estimativa de distância. Numa situação ideal (Figura 4.1a), tal implicaria um ponto único no espaço. No entanto, devido ao ruído presente nas observações do modelo representado na equação 2.3, o ponto referido não é único. Dependendo do nível de ruído que perturba as observações, o

conjunto dos pontos de interceção poderá não formar um espaço convexo, limitando as possíveis metodologias aplicáveis ao problema.

Tendo em conta o exposto, considera-se o problema de localização com N sensores, dos quais são obtidas \mathbf{y}_N observações, sendo estimadas $\hat{\mathbf{d}}_N$ distâncias. Conforme analisado no capítulo 3, para determinação de uma estimativa inicial, que neste caso corresponde ao estado inicial da cadeia de *Markov*, considera-se a interceção entre pares de sensores, sendo obtidas C_2^N combinações.

O número de pontos a utilizar para a geração da população irá depender do número de *clãs* considerado para o algoritmo de *iEHO*, obtidos através da expressão:

$$L = \frac{\binom{N}{2}}{NClas}. \quad (4.1)$$

Relativamente a cada combinação de sensores, são calculados pontos dependentes das suas interceções. No caso de circunferências secantes, consideram-se os pontos de interceção da circunferência com o segmento de reta que une as posições dos sensores. No caso de circunferências externas ou internas, considera-se o ponto intermédio do segmento de reta que une os sensores, descontando os raios das mesmas:

$$\begin{aligned} \mathbf{P}_{Al} \in \mathbb{R}^2 : \|\mathbf{P}_{Al} - \mathbf{S}_i\| = d_i \wedge \mathbf{P}_{Al} = \mathbf{S}_i + k_i \overrightarrow{\mathbf{S}_i \mathbf{S}_j}, k_i \in \mathbb{R}^+ \\ \mathbf{P}_{Bl} \in \mathbb{R}^2 : \|\mathbf{P}_{Bl} - \mathbf{S}_j\| = d_j \wedge \mathbf{P}_{Bl} = \mathbf{S}_j + k_j \overrightarrow{\mathbf{S}_j \mathbf{S}_i}, k_j \in \mathbb{R}^+ \end{aligned} \quad (4.2)$$

As expressões 4.2 generalizam as proposições 3.1 e 3.2 desenvolvidas no capítulo 3, onde a determinação das constantes k_i e k_j foram abordadas detalhadamente. Neste caso, k_i e k_j representam um fator de escala nos vetores que unem as coordenadas dos sensores, obtendo-se o ponto de interceção, \mathbf{P}_{Al} e \mathbf{P}_{Bl} . Realça-se também o facto de que os vetores diretores assumem direções opostas conforme $\overrightarrow{\mathbf{S}_i \mathbf{S}_j}$ e $\overrightarrow{\mathbf{S}_j \mathbf{S}_i}$ em relação a \mathbf{P}_{Al} e \mathbf{P}_{Bl} . Caso contrário, o ponto obtido corresponderia à segunda interceção do segmento de recta com a circunferência, correspondente a um ângulo πrad , e que estaria fora da zona de interesse. O estado inicial do método é determinado pelos centros de massa dos pontos obtidos, conforme a expressão:

$$\mathbf{P}_l = \frac{\mathbf{P}_{Al} + \mathbf{P}_{Bl}}{2}, \quad l = 1, \dots, L. \quad (4.3)$$

Uma representação gráfica do exposto, identificando o ponto de interesse procurado e o ponto que deve ser desprezado, é representada na figura 4.2, onde $S_i \hat{S}_j$ representa o vetor unitário (versor), definido por: $S_i \hat{S}_j = \frac{\vec{S}_i \vec{S}_j}{|\vec{S}_i \vec{S}_j|}$.

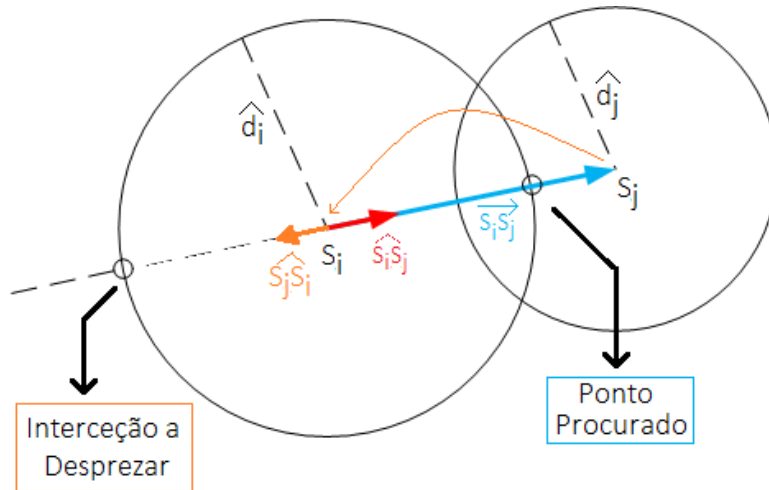


Figura 4.2: Ponto de interesse perante as duas interseções entre a reta e a circunferência.

4.3 Cadeias de Markov a partir do Algoritmo de Metropolis-Hastings

O algoritmo de *Metropolis-Hastings* corresponde a uma implementação do método de *Monte Carlo* baseado em Cadeias de *Markov* (MCMC).

O algoritmo destina-se a fornecer um método para amostragem a partir de uma distribuição de probabilidade genérica, de modo a gerar a população inicial.

A motivação do método consiste no facto de, embora possa ser descrita matematicamente uma expressão para a PDF (expressão 4.4), a mesma seja demasiado complexa para viabilizar o seu cálculo. Note-se que no presente caso, trata-se da obtenção de amostras a partir da distribuição Normal considerada como a PDF do ruído das observações do modelo:

$$f(y_1, y_2, \dots, y_n | \sigma, \mu_i) = \prod_{i=1}^n \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(y_i - \mu_i)^2}{2\sigma^2}}. \quad (4.4)$$

O conceito de base consiste em definir uma Cadeia de *Markov* sobre os possíveis valores de \mathbf{X} , de modo a que a distribuição estacionária da Cadeia de *Markov*

corresponda de facto à distribuição 4.4. Assim, pretende gerar-se uma sequência de valores \mathbf{X}_n , denotados (x_0, x_1, \dots, x_n) , de modo a que, para $n \rightarrow \infty$, possamos garantir que $\mathbf{X}_n \sim f(y_1, y_2, \dots, y_n | \sigma, \mu_i)$ [196].

No algoritmo Metropolis-Hastings, pretendendo gerar-se um ponto x_{n+1} a partir de um estado x_n , procede-se a uma formulação em dois passos.

O primeiro consiste em gerar um ponto candidato, representado por x^* , a partir de uma distribuição considerada, aqui denotada por $Q(x^* | x_n)$, que depende naturalmente do estado atual da cadeia de *Markov*, x_n .

Considera-se para efeitos do presente estudo a distribuição Normal, ou seja,

$$x^* | x_n \sim \mathcal{N}(x_n, \sigma_M^2). \quad (4.5)$$

Note-se que a geração do ponto será dependente de um σ_M^2 especificado (expressão 4.5). A determinação desse ponto será dependente do segundo passo do método proposto, sendo fornecido um procedimento para a sua determinação no decorrer desta secção.

Numa segunda etapa, denominada de "Aceitação/Rejeição", procede-se à decisão se o ponto gerado deve, ou não, ser integrado na sequência. Para tal, define-se uma probabilidade de aceitação $A(x_n \rightarrow x^*)$ expressa por:

$$A(x_n \rightarrow x^*) = \min\left(1, \frac{P(x^*)}{P(x_n)} \times \frac{Q(x_n | x^*)}{Q(x^* | x_n)}\right). \quad (4.6)$$

Tendo em conta a propriedade de simetria da distribuição Normal considerada no modelo $Q(x_n | x^*) = Q(x^* | x_n)$, o seu quociente será unitário. A etapa termina com uma condição de aceitação perante o valor calculado de 4.6, dependente da geração de um valor aleatório, com $\lambda \in [0, 1]$.

Aplicando a metodologia, um ponto candidato gerado é aceite ($x_{n+1} = x^*$) ou é rejeitado ($x_{n+1} = x_n$) consoante o resultado de 4.7. A relação entre o número de pontos gerados e o número de pontos aceites será definido como taxa de aceitação (T_{Accept}), definida por:

$$x_{n+1} = \begin{cases} x^*, & \lambda \leq A(x_n \rightarrow x^*) \\ x_n, & \lambda > A(x_n \rightarrow x^*) \end{cases}. \quad (4.7)$$

Existem algumas restrições do ponto de vista das distribuições que implicam a convergência do método e que devem ser salvaguardadas. No entanto, tendo em conta que foi considerada a distribuição Normal, não se considera a necessidade de uma análise teórica de convergência [196, 198–200].

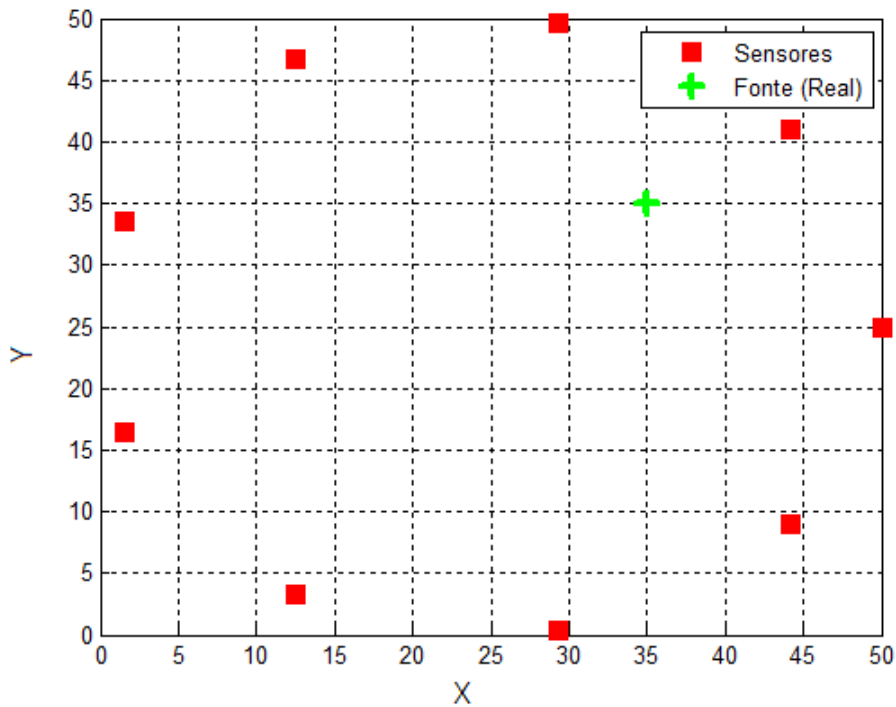


Figura 4.3: Condições de simulação ($N = 9$ sensores)

Em relação a σ_M^2 e para dar resposta à sua determinação, procede-se a uma análise do comportamento do método em função deste mesmo parâmetro. Para tal, foram consideradas condições de simulação semelhantes às já expostas, nomeadamente um espaço de procura de 50×50 , $N = 9$ sensores distribuídos de um modo uniforme e circular, em torno do centro do espaço de procura (figura 4.3). Também uma potência de transmissão $P = 500$, $g_i = 1$, uma potência de ruído de -20 dB, e grupos de 3 sensores para obtenção das estimativas iniciais. O ponto real é desconhecido, sobre o qual se consideram as observações com ruído, tem coordenadas (35 m, 35 m).

Como referido anteriormente, o processo inicia-se, aplicando a metodologia desenvolvida no Capítulo 3.

4. Inicialização Inteligente da População pelo Método de Monte Carlo

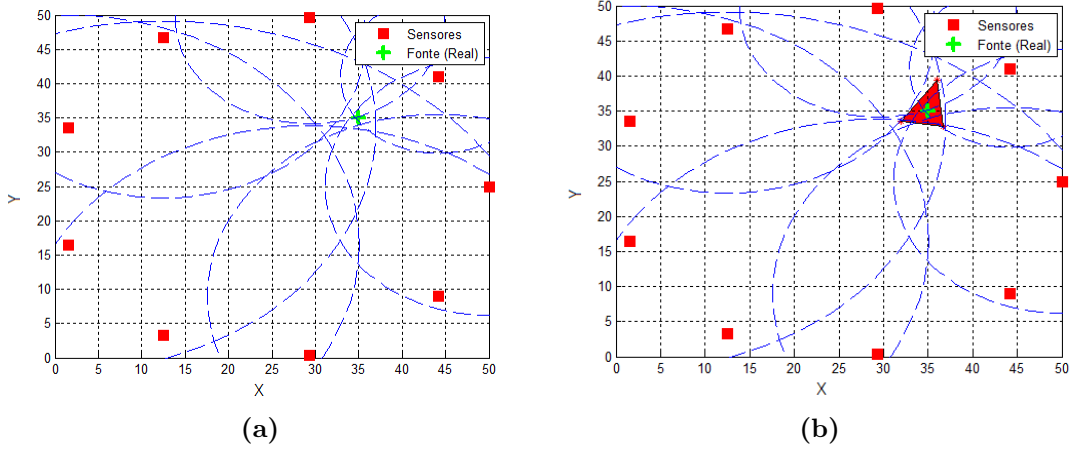


Figura 4.4: Ponto de partida para a geração da Cadeia de *Markov*. (a) Raios obtidos pela estimação da distância (b) Espaço convexo obtido por aplicação do Algoritmo (3.1)

Na figura 4.4a pode observar-se o efeito das estimativas de distância em torno do ponto de interesse. Após aplicação do método de determinação dos pontos referentes às expressões 4.2, obtém-se o espaço convexo representado na figura 4.4b. Os pontos que delimitam a figura serão os pontos de partida das sequências das Cadeias de *Markov* a desenvolver.

Os resultados apresentados na figura 4.5, foram obtidos considerando $\sigma_M^2 \in [0.01 \ 0.1 \ 0.2 \ 0.4 \ 1 \ 10]$. Para cada valor de σ_M^2 , foi considerada a geração de 1000 pontos da sequência de *Monte Carlo*. Em relação a valores de σ_M^2 demasiado elevados, é notória a dispersão dos pontos. No caso de $\sigma_M^2 = 10$ é obtida uma taxa de aceitação de 0.7% (figura 4.5f). Pelo contrário, no caso de valores de σ_M^2 reduzidos ($\sigma_M^2 = 0.01$), a sequência tem uma dinâmica muito baixa, sendo os pontos gerados em torno do início da sequência (figura 4.5a). Nesta situação, temos uma taxa de aceitação de 85.4%.

Tendo em conta o comportamento estocástico do método, sobretudo devido ao comportamento de λ , procede-se a uma análise mais exaustiva da taxa de aceitação. Para tal, são realizadas 1000 gerações para cada um dos valores de σ_M^2 , registando-se o valor médio da taxa de aceitação.

Note-se que os primeiros pontos da sequência gerados ficarão próximos do estado inicial, e só a partir de um determinado momento do processo de geração estes se irão

4.3. Cadeias de Markov a partir do Algoritmo de Metropolis-Hastings

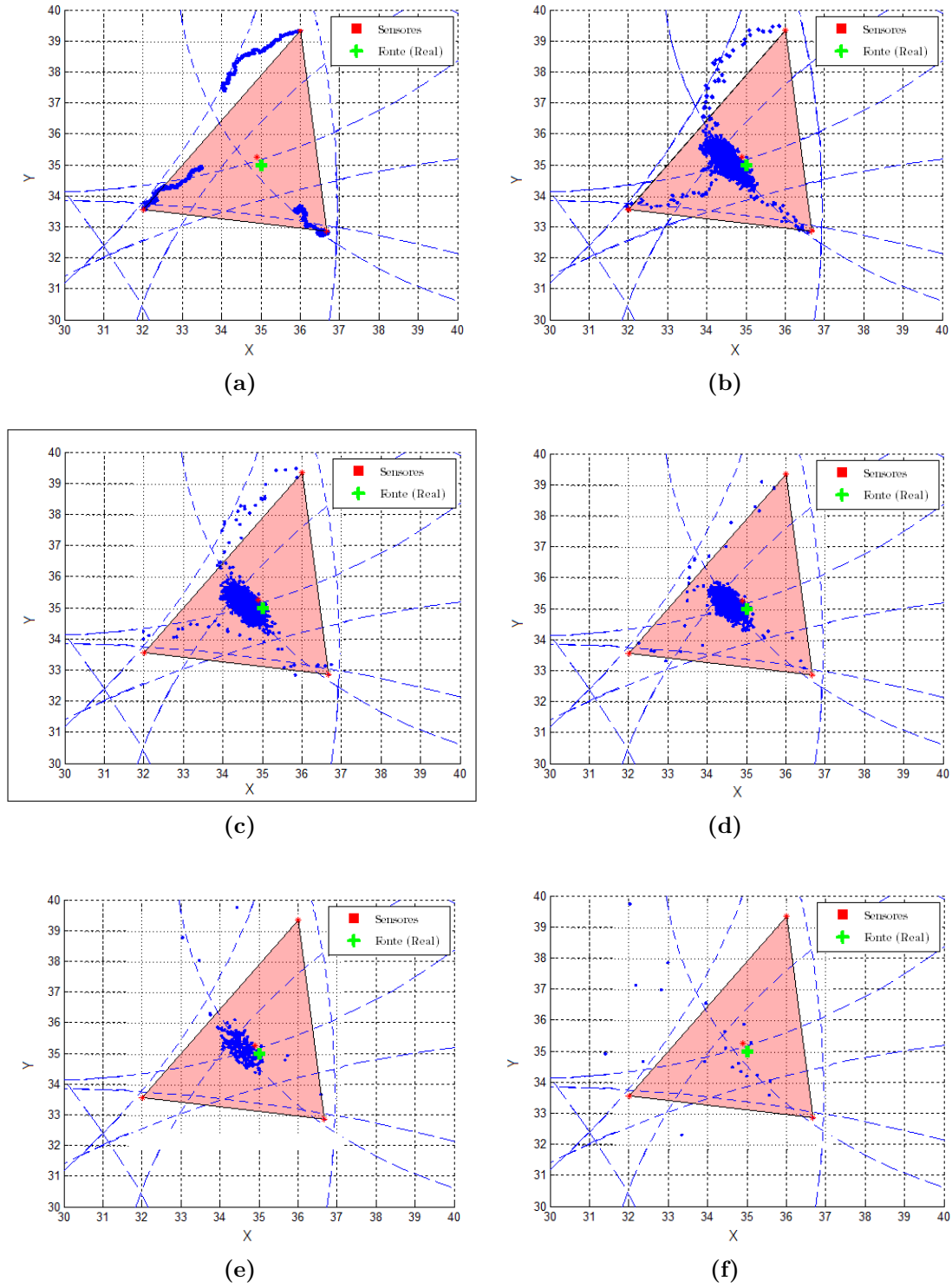


Figura 4.5: Pontos gerados da sequência de Metropolis-Hastings.

- (a) $\sigma_M^2 = 0.01$, $T_{Accept} = 85.40\%$ (b) $\sigma_M^2 = 0.1$, $T_{Accept} = 73.10\%$
(c) $\sigma_M^2 = 0.20$, $T_{Accept} = 56.90\%$ (d) $\sigma_M^2 = 0.4$, $T_{Accept} = 38.60\%$
(e) $\sigma_M^2 = 1.00$, $T_{Accept} = 29.80\%$ (f) $\sigma_M^2 = 10$, $T_{Accept} = 0.70\%$

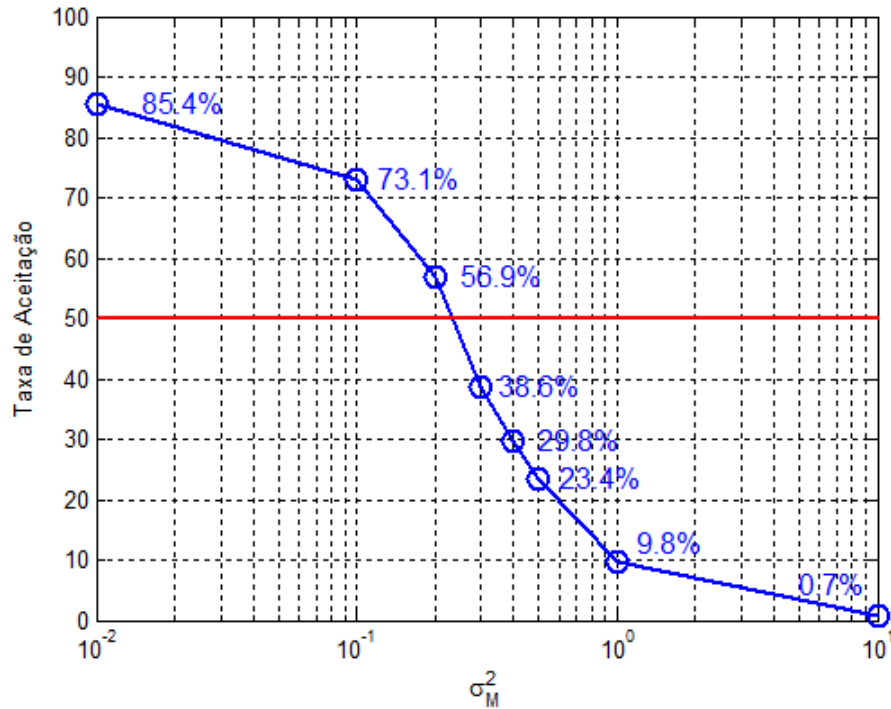


Figura 4.6: Variação da taxa de aceitação com σ_M^2

aproximar da área de interesse, o centro de massa dos vários estados iniciais. Para que seja considerado um número de pontos da sequência mais concentrados junto ao centro de massa, considera-se a necessidade de gerar o dobro dos pontos necessários (o dobro da população pretendida) e procura obter-se uma taxa de aceitação de metade dos valores gerados. Como se observa através dos resultados obtidos, no sentido de obter uma T_{Accept} na ordem dos 50%, valor que mais se ajusta à concentração de pontos pretendida (ver figura 4.5 e tabela 4.1), deve considerar-se $\sigma_M^2 = 0.2$.

Tabela 4.1: Variação da taxa de aceitação com σ_M^2

σ_M^2	0.01	0.1	0.2	0.3	0.4	0.5	1	10
$\overline{T_{Accept}}$	85.40%	73.10%	56.90%	38.60%	29.80%	23.40%	9.80%	0.70%
$\sigma_{T_{Accept}}$	1.35	1.16	1.04	0.95	0.91	0.87	0.86	0.05

Na tabela 4.1, $\overline{T_{Accept}}$ representa a média da taxa de aceitação das 1000 sequências geradas, e $\sigma_{T_{Accept}}$ o respetivo desvio padrão.

No entanto, dependendo do comportamento do modelo no presente caso, das observações e do respetivo ruído, pode ser necessário sintonizar outros valores de

σ_M^2 . Os resultados de simulação apresentados apenas são válidos nas presentes condições de simulação ($\mathbf{X} \in [0, 50]m$, $N = 9$, $P = 500$, $g_i = 1$ e $\sigma_v^2 = -20$ dB). No entanto, é possível considerar a metodologia inversa para obter um valor de σ_M^2 para um determinado conjunto de condições. A estratégia consiste em utilizar um valor de σ_M^2 arbitrário, gerar uma sequência de pontos e calcular a taxa de aceitação. O processo de ajuste, de modo a obter um valor de cerca de 50%, pode ser facilmente realizado por pesquisa binária sem implicar processos de cálculo complexos (complexidade de ordem $O(\log(n))$) [201, 202].

Algoritmo 4.1 Método de Localização com Inicialização Metropolis-Hastings

```

1: function METROEHO( $\mathbf{S}, \mathbf{y}$ )
2:    $L \leftarrow \text{length}(S)$  ▷ Número de sensores
3:    $q = 0 : L/3 : L - L/3$  ▷ Seleciona o sensor mais distante para cada grupo
4:    $\sigma_M^2 \leftarrow \sigma_{v_i}^2$  ▷ Estimar a Variância das observações [203, 204]
5:    $n \leftarrow 1$ 
6:   for  $k = 1 : L/3$  do ▷ Grupos de 3 sensores para criação dos clãs
7:      $P(n) \leftarrow \text{Algoritmo (3.1)}$  ▷ Expressões sumárias em (4.2) e (4.3)
8:     while ( $n < \text{MaxIter}$ ) do ▷ Ciclo opcional
9:        $[\mathbf{X}_{\text{MH}}, T_{\text{Accept}}] = \text{Metropolis}(N_s, \sigma_M^2, P(n))$ 
10:      if ( $T_{\text{Accept}} \subset 50\% \pm Tol$ ) then ▷ Manter  $\sigma_M^2$  constante
11:        Break ▷ Sequência validada
12:      else if ( $T_{\text{Accept}} > 50\% + Tol$ ) then
13:         $\sigma_M^2 \uparrow\uparrow$ 
14:      else ▷ Aplicar método de pesquisa
15:         $\sigma_M^2 \downarrow\downarrow$ 
16:      end if
17:    end while
18:     $\mathbf{X}_i \leftarrow \text{Sort}(\mathbf{X}_{\text{MH}})$  ▷ Ordenar o vetor com a métrica da Função de Custo
19:     $n++$  ▷ Conjunto de sensores seguinte
20:  end for
21:   $\mathbf{X}_{\text{Pop}} \leftarrow \text{Organize}(\mathbf{X}_i)$  ▷ Divide a população em clãs
22:   $X_f = \text{EHO}(\mathbf{X}_{\text{Pop}}, \text{Param})$  ▷ Executa o método de EHO
23: end function

```

O algoritmo 4.1 sumaria o procedimento descrito. Note-se que a variância das observações é considerada conhecida, não sendo objeto de análise no presente trabalho. Considera-se uma tolerância em relação ao valor pretendido para a taxa de aceitação de 50% (Tol), de modo a criar uma janela de histerese em torno do

valor central. Quanto ao número máximo de iterações do método da bissecção, considera-se um limite (*MaxIter*), de modo a não entrar num ciclo infinito.

No entanto, o valor de σ_M^2 pode ser mantido constante para um determinado problema, diminuindo a complexidade do algoritmo. Deve ter-se em atenção que tal procedimento pode restringir a generalidade pretendida para a abordagem ao contexto de localização. O sucesso de tal abordagem dependerá das condições reais de implementação.

4.4 Resultados e Discussão

As condições de simulação já referidas na secção 4.3, foram agora estendidas para vários valores da variância do erro das observações ($\sigma_\nu^2 \in [-30, -5]$ dB), com passos de 5 dB. O valor de σ_M^2 foi mantido constante, conforme análise efetuada anteriormente (execução da linha 10 do algoritmo 4.1 e a não implementação do ciclo indicado na linha 8, do mesmo algoritmo).

Na figura 4.7 podemos visualizar o comportamento do método do ponto de vista do *RMSE* calculado.

A figura 4.7 compara os resultados do algoritmo EHO na sua forma original, o método SOCP, os resultados obtidos no capítulo 3.3 (iEHO) e a nova implementação em que a população é inicializada com o algoritmo MCMC.

No sentido de correlacionar o resultado do erro com a simplicidade de computação obtida com a nova metodologia, a figura 4.8 apresenta os histogramas do número de avaliações da função de custo. Esta é uma medida indireta do número de iterações realizadas e, portanto, do tempo de computação.

Perante os resultados obtidos, verifica-se que a simulação implementada representa dois cenários distintos. Por um lado, quando considerados níveis de ruído baixos, o método apenas executa uma geração de população, por outro, para níveis de ruído elevado, o número de gerações aumenta significativamente. Tal situação, deve-se ao comportamento que a sequência de *Markov* tem perante valores diferentes de ruído. Com ruído mais elevado o primeiro ponto da sequência obtido da inicialização dos clãs, está mais afastado da solução pretendida, ou seja, do ponto ótimo do

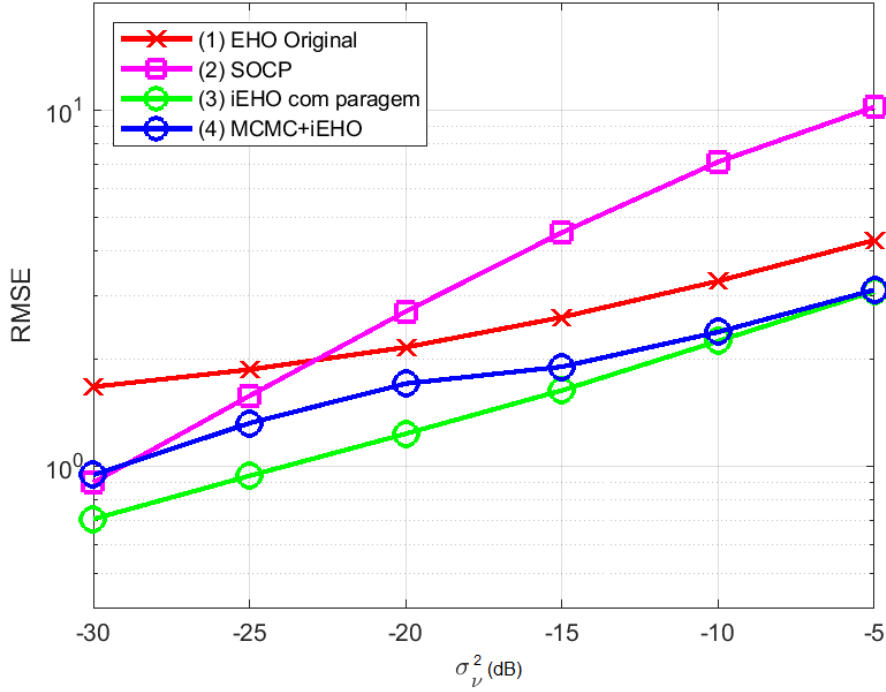


Figura 4.7: iEHO com inicialização Metropolis-Hastings, $N = 9$ sensores

modelo. Mantendo-se σ_M^2 , o nível de dispersão não será suficiente para obter uma concentração de pontos em torno do ótimo (análise apresentada na figura 4.5).

Como consequência, o método de EHO tem maior preponderância, executa mais iterações e o erro de estimação aproxima-se dos resultados obtidos com o método iEHO descrito no capítulo 3.3.

Por outro lado, com σ_M^2 devidamente ajustado, a sequência de *Monte Carlo* produz um conjunto de pontos utilizados como população inicial do método de EHO, em torno do ponto ótimo do modelo. Como se pode verificar pela análise da figura 4.8, o facto de a maioria das simulações se encontrar na classe 250 implica que o método de EHO apenas realizou uma iteração. Observa-se no entanto, que o erro é ligeiramente maior. Deve realçar-se que a diferença é muito pouco significativa e que a ordem de valores é de cerca de $0.3 m$.

A principal vantagem da abordagem implementada consiste precisamente no facto de estarmos perante a possibilidade de uma metodologia independente. Esta é baseada na criação de um conjunto de candidatos por estimação das distâncias,

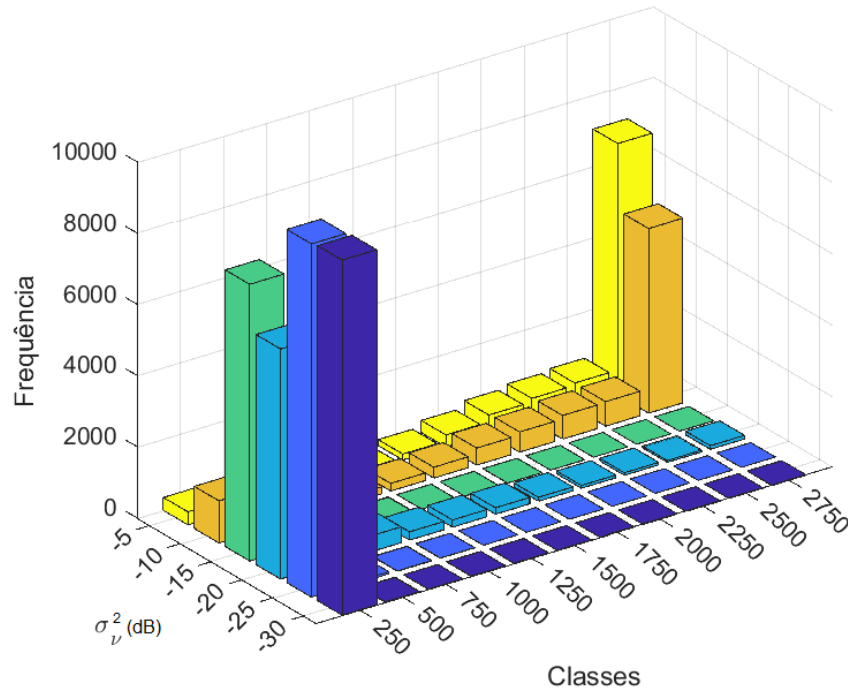


Figura 4.8: Número de avaliações (MCMC+iEHO), $N = 9$ sensores

aplicando as expressões 4.4 para gerar os pontos iniciais de uma sequência MCMC, pelo algoritmo *Metropolis-Hastings*. Os resultados assim obtidos apresentam uma solução sub-ótima que pode ser satisfatória para determinadas aplicações.

De salientar que, numa metodologia desta ordem, as instruções apresentadas como opcionais no pseudocódigo do algoritmo 4.1 seriam implementadas. O comentário apresentado na linha 10 do referido algoritmo não seria válido e o valor de σ_M^2 teria de ser analisado com rigor, nomeadamente através da metodologia proposta no algoritmo 4.1. A simplificação efetuada implicou um comportamento não uniforme nos resultados. Para valores do nível de ruído mais elevados, o valor de σ_M^2 necessitaria ser sintonizado novamente com a implementação das instruções consideradas opcionais no algoritmo 4.1.

4.5 Conclusão

Este capítulo fecha os trabalhos de investigação desenvolvidos no âmbito desta dissertação.

Além de propor uma nova metodologia para abordagem ao problema de localização acústica, assume também um papel de sistematização do cálculo dos pontos, para determinação da população inicial dos métodos.

O trabalho realizado para inicialização de uma população dividida em grupos é sistematizado do ponto de vista de notação matemática, permitindo uma melhor compreensão formal. No entanto, ao invés de gerar pontos aleatórios em torno de um conjunto de interseções determinadas com base na estimativa de distância, os mesmos são considerados como pontos iniciais de uma sequência de *Markov*, aplicando o método de *Metropolis–Hastings*.

Os resultados obtidos através de simulação computacional permitem demonstrar um compromisso entre erro (RMSE) e a taxa de convergência. A mesma simulação mostra os dois cenários, onde um compromisso entre um leve aumento do erro para aumentar a convergência contrasta com uma convergência mais lenta para obter erros mais reduzidos.

Sobretudo a principal vantagem deste capítulo consiste em poder considerar-se a proposta de uma nova metodologia de abordagem ao problema em estudo na presente dissertação, sem o recurso a métodos mais ao menos morosos em termos de computação.

Perante uma inicialização inteligente, obtêm-se pontos para o arranque de uma cadeia de *Markov*, aplicando o algoritmo de *Metropolis–Hastings* que, sem qualquer outro processamento, permite obter soluções com menor erro e menor complexidade em relação ao atual estado da arte.

5

Conclusão

Conteúdos

5.1 Conclusão	99
5.2 Trabalho Futuro	101

5.1 Conclusão

A presente dissertação traçou, como objetivo, a aplicação de métodos meta-heurísticos ao problema de localização de uma fonte acústica. Não sendo esta uma interpelação comum no estado da arte, a abordagem ao problema passou por várias fases, nomeadamente a validação da adequação dos métodos baseados em enxames de partículas ao problema de localização.

Numa segunda fase, com base num algoritmo concreto, foram propostas diversas metodologias que permitiram obter avanços ao nível do erro de estimação da posição, do rácio de convergência dos métodos (redução do número de iterações) e indiretamente, do tempo de computação.

No capítulo 4, foi apresentado um avanço que pode ser considerado significativo neste trabalho. De facto, os resultados demonstram o desenvolvimento de um novo método para a solução do problema de localização acústica. No entanto, todas

as metodologias propostas, nomeadamente a nova metodologia objeto de realce, podem ser estendidas a qualquer problema de localização baseado na medição (in)direta de distância (RSSI, ToA ou TDoA).

Em suma, é sabido que incorporar mais informação num modelo computacional irá incrementar o seu desempenho. A presente dissertação responde a uma das questões que se coloca quando se aborda um modelo que representa um determinado problema:

1. Que informação pode, ou deve ser incorporada de modo a melhorar o seu desempenho?
2. De que modo essa mesma informação deve ser considerada e aplicada?

Assim, apresentou-se desenvolvimento e validação de uma metodologia genérica para a inicialização de uma população candidata à solução de um problema de localização, baseada na medição (in)direta de distância (considerado de forma isolada).

Os resultados obtidos assumem ainda maior preponderância quando comparados com os métodos homólogos determinísticos. Além de terem sido alcançadas as melhorias já referidas, a principal conquista consiste nas possibilidades futuras de implementação ao nível aplicacional. Sendo este tipo de problemas normalmente implementado de forma centralizada, com o nível de simplificação obtida poderá ser possível implementações distribuídas e/ou sequenciais. Além disso, a possibilidade de utilização de arquiteturas computacionais menos complexas apresenta-se como um avanço no atual estado da arte.

Por fim, realça-se o facto do trabalho desenvolvido ter sido publicado numa revista científica de segundo quartil (indexação Scopus, ISI e WoS), numa revista de primeiro quartil (indexação Scopus, ISI e WoS), e apresentado numa conferência internacional indexada pelo *IEEE*.

5.2 Trabalho Futuro

Perante o trabalho desenvolvido no âmbito desta dissertação, uma das áreas mais promissoras de melhoramento, com desenvolvimento ao nível de trabalho futuro, diz respeito aos métodos descritos no capítulo 4. A abordagem apresentada pretendeu realçar o comportamento do método em duas vertentes diferentes, que contrabalançam o efeito do erro, com o efeito da convergência. Assim, automatizar este processo para um ponto intermédio com a implementação das propostas sugeridas, é um aspeto a ter em conta como área de atuação premente.

Aprofundar o desenvolvimento das metodologias desenvolvidas através da sua extensão à localização com base na medição de RSSI, ToA, TDoA, AoA deve ser uma prioridade do ponto de vista científico. Por outro lado, tendo em conta a proliferação de métodos atualmente propostos na literatura, continuar a desenvolver mais implementações noutros algoritmos meta-heurísticos reforçará as reivindicações já demonstradas.

Esta dissertação abre portas em relação a várias linhas de investigação que podem ser consideradas do ponto de vista de trabalho futuro. As mesmas podem assumir bases científicas de diversas natureza, apresentado-se de seguida diferentes propostas:

1. Processamento Digital de Sinal
 - (a) Tendo sido apenas considerada uma fonte acústica num contexto centralizado, pode ter-se em conta a generalização a múltiplas fontes, assim como à localização distribuída ou sequencial;
 - (b) Incorporação de modelos de movimento, no sentido da consideração de fontes de sinal estáticas;
 - (c) Integração de estimadores mais complexos e filtragem Bayesiana;
2. Computação
 - (a) Implementação e teste das metodologias propostas com dados reais, a partir de sinais de banda larga, para deteção e localização de eventos acústicos;

- (b) Implementação, teste, validação e obtenção de métricas de performance, com implementações em arquiteturas computacionais de baixa complexidade e baixo consumo (*Embedded Programming*).

Anexos



Demonstrações

Conteúdos

A.1 Demonstração da Proposição 3.1	105
A.2 Demonstração da Proposição 3.2	108

A.1 Demonstração da Proposição 3.1

Demonstração. Considere os dois triângulos retângulos formados por $\triangle S_4P_bP_x$ e $\triangle S_1P_bP_x$.

Aplicando o Teorema de Pitágoras, obtemos duas expressões. Função de a , b e h (figura. A.1). Como temos três variáveis, adicionamos uma terceira equação relacionando a e b com a distância conhecida dos pontos (sensores) P_4 e P_1 .

$$\begin{cases} a^2 + h^2 = R_4^2 \\ b^2 + h^2 = R_1^2 \\ d = a + b \end{cases} \quad (\text{A.1})$$

Substituindo a terceira expressão de A.1 pela primeira, obtemos:

$$a = \frac{R_1^2 - R_4^2 + d^2}{2d} \quad (\text{A.2})$$

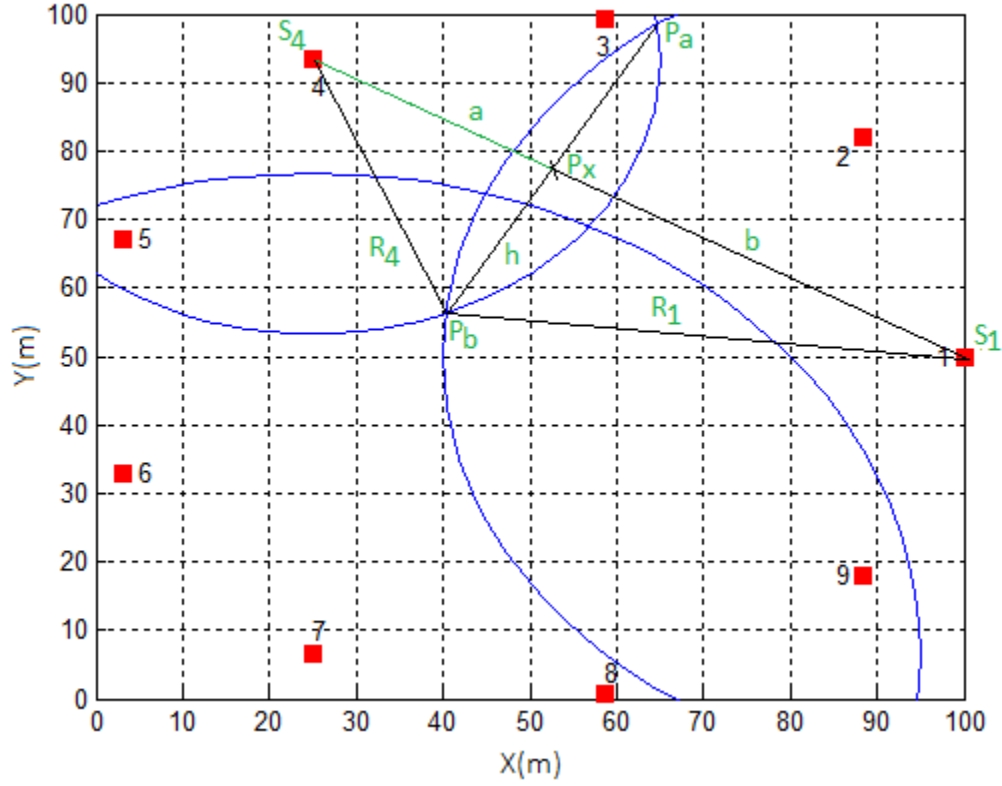


Figura A.1: Circunferências secantes (Proposição 3.1)

Observe que $d = \|P_1 - P_4\|$. Substituindo A.2 na primeira expressão de A.1 obtemos:

$$h^2 = R_1^2 - a^2 \quad (\text{A.3})$$

As coordenadas de P_x , o ponto intermédio do eixo radial que une os pontos de interceção dos dois círculos secantes, podem ser obtidas pela relação das distâncias:

$$P_x = P_4 + \frac{a}{d}(P_1 - P_4) \quad (\text{A.4})$$

Observe-se que P_x é o ponto intermédio do segmento de reta $\overline{P_a P_b}$. A partir deste ponto, podemos determinar as extremidades dos segmentos de reta, embora no presente caso, desejemos apenas determinar o ponto de menor distância em relação ao raio estimado do sensor S_7 (P_b correspondente a R_7).

$$\begin{cases} x_b = P_x(x) - h \frac{P_4(y) - P_1(y)}{d} \\ y_b = P_x(y) + h \frac{S_j(x) - S_i(x)}{d} \end{cases}$$

No caso em que $\|P_7 - P_b\| > R_7$, significa que estamos a obter o ponto do segmento de reta do eixo radial mais distante e devemos corrigir a expressão anterior.

$$\begin{cases} x_b = P_x(x) + h \frac{P_4(y) - P_1}{d} \\ y_b = P_x(y) - h \frac{P_4(x) - P_1(x)}{d} \end{cases}$$

Para obter os dois pontos restantes, repetimos o procedimento com os pares de pontos alternados. ■

A.2 Demonstração da Proposição 3.2

Demonstração. Considere o triângulo $\triangle S_7S_1E$ (fig. A.2). Como S_1 e S_7 são conhecidos, podemos calcular o ângulo α_p com a relação cosseno:

$$\alpha_p = \cos^{-1} \left(\frac{|S_1(x) - S_7(x)|}{\|S_1 - S_7\|} \right)$$

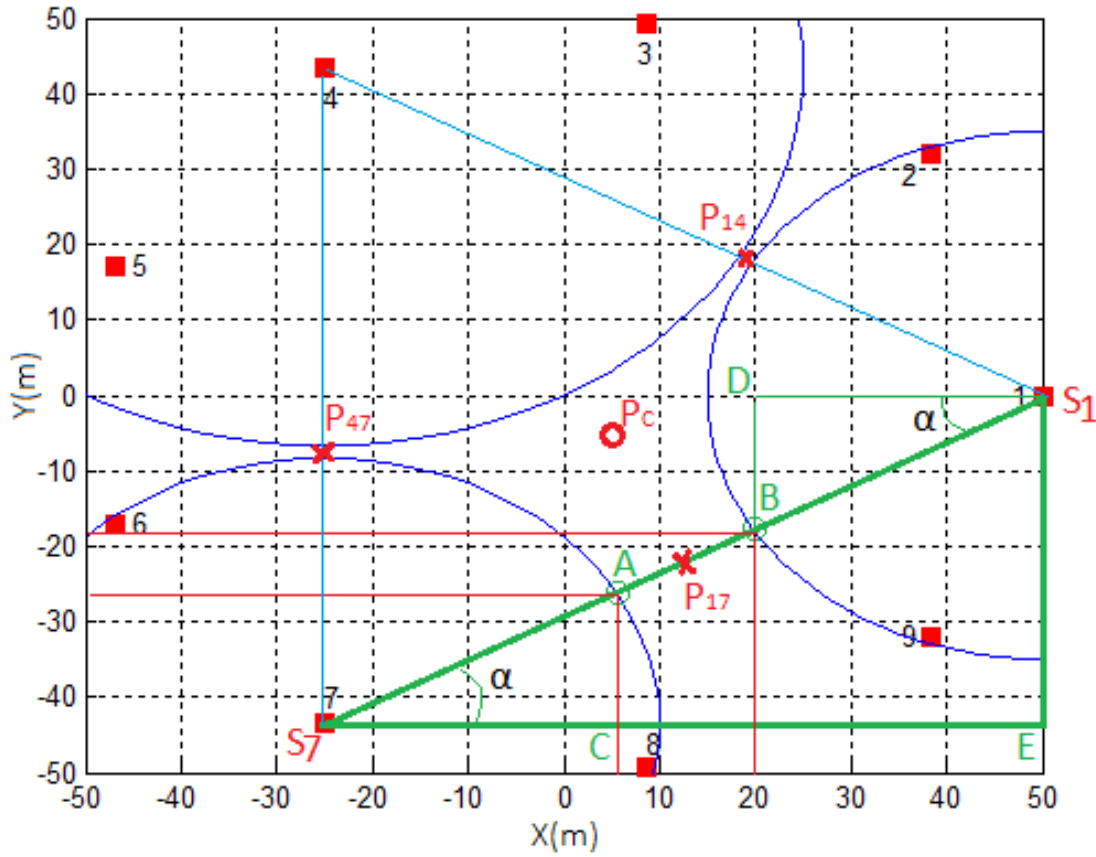


Figura A.2: Circunferências secantes (Proposição 3.2)

Os pontos P_A e P_B são calculados com a relação seno e cosseno dos triângulos $\triangle S_7P_C P_A$ e $\triangle S_1P_B P_D$, respetivamente.

$$P_A(x, y) = \begin{cases} P_A(x) = S_7(x) + \cos(\alpha_p)R_7 \\ P_A(y) = S_7(y) + \sin(\alpha_p)R_7 \end{cases}$$

$$P_B(x, y) = \begin{cases} P_B(x) = S_1(x) - \cos(\alpha_p)R_1 \\ P_B(y) = S_1(y) - \sin(\alpha_p)R_1 \end{cases}$$

A. Demonstrações

Para determinar P_{17} , consideramos a distância intermédia entre $\overline{P_A P_B}$. Para obter P_{47} e P_{14} aplica-se a mesma metodologia com os sensores S_7 e S_4 e S_4 e S_1 , respetivamente. ■

B

Histogramas e Indicadores Estatísticos (Cap. 3)

B.1 Histogramas Correspondes ao Caso de Estudo

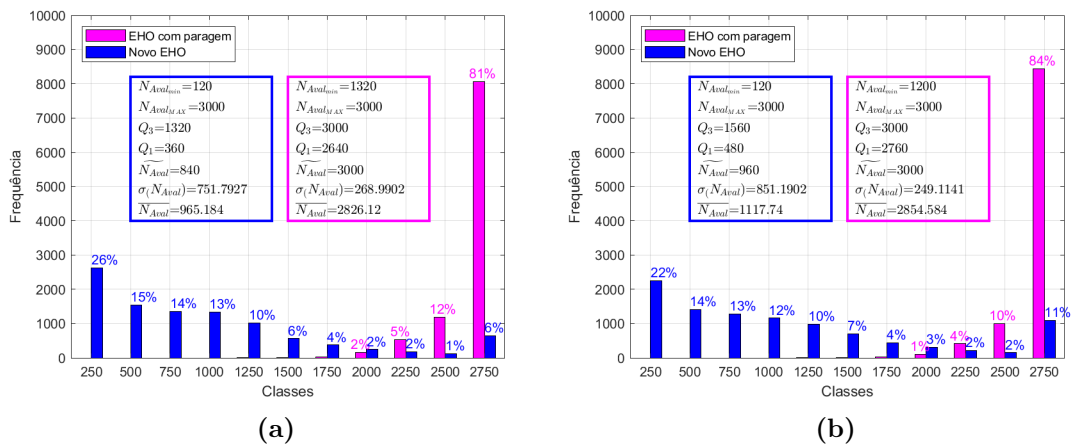


Figura B.1: Histogramas e dados estatísticos para $\sigma_v^2 = -40$ dB. (a) $N = 9$ sensores
(b) $N = 12$ sensores

B.1. Histogramas Correspondes ao Caso de Estudo

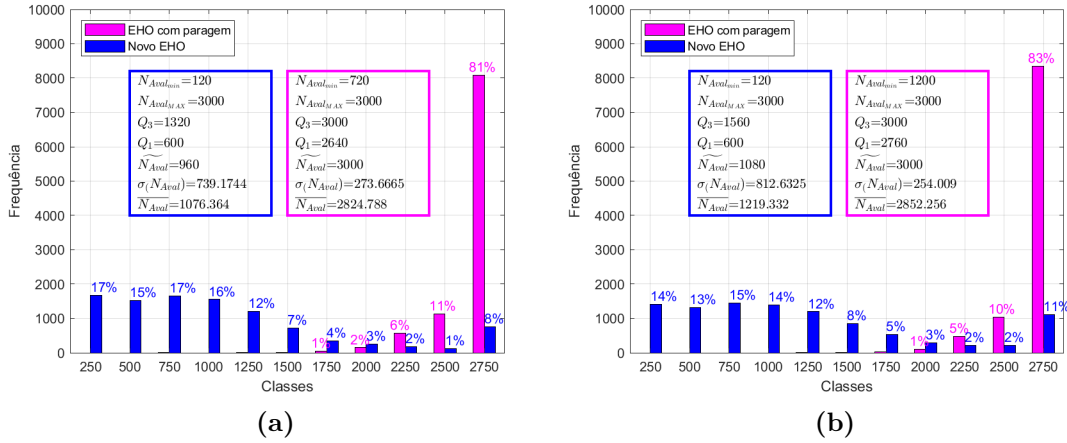


Figura B.2: Histogramas e dados estatísticos para $\sigma_v^2 = -35 \text{ dB}$. (a) $N = 9$ sensores (b) $N = 12$ sensores

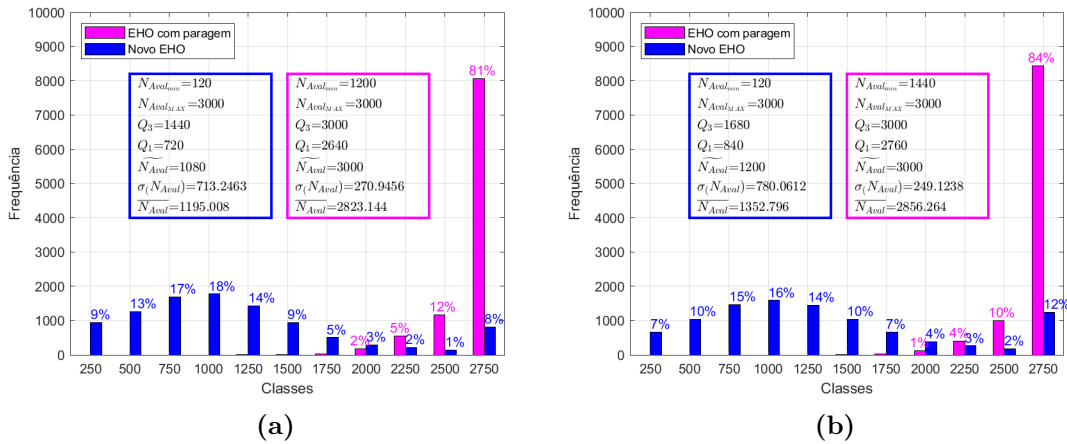


Figura B.3: Histogramas e dados estatísticos para $\sigma_v^2 = -30 \text{ dB}$. (a) $N = 9$ sensores (b) $N = 12$ sensores

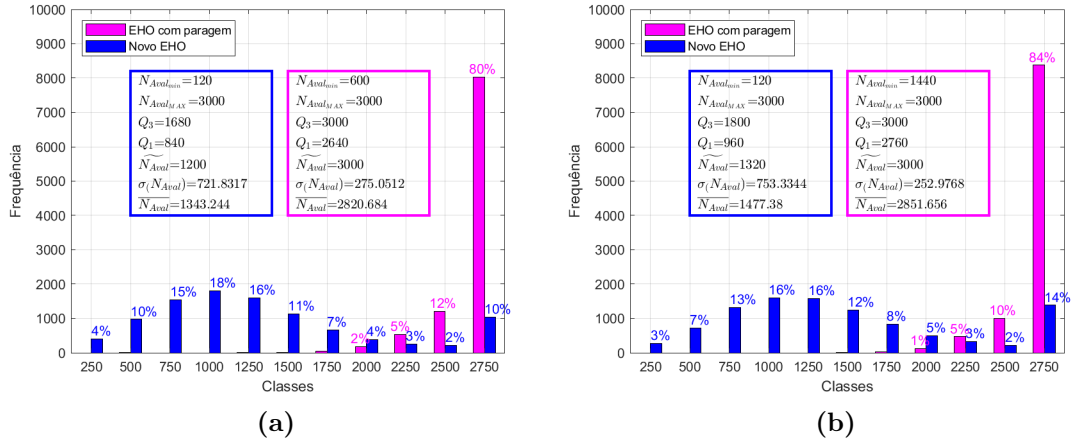


Figura B.4: Histogramas e dados estatísticos para $\sigma_v^2 = -25$ dB. (a) $N = 9$ sensores (b) $N = 12$ sensores

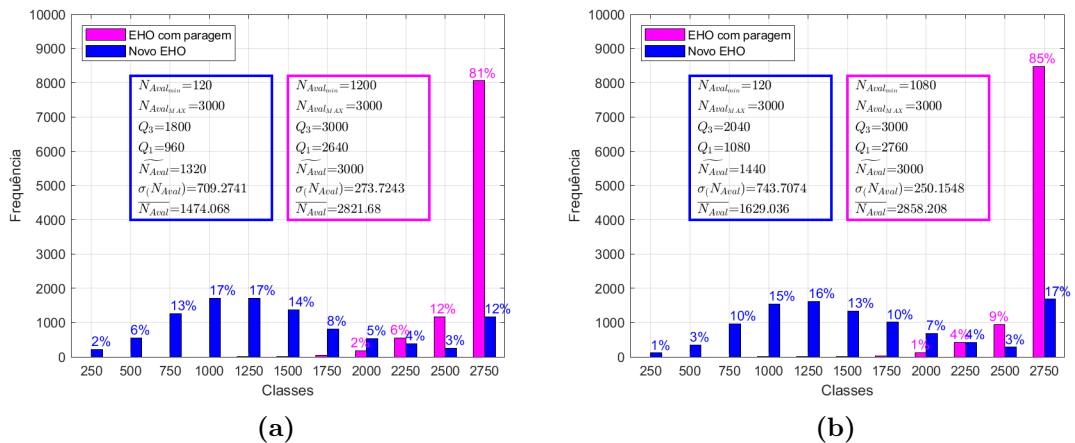


Figura B.5: Histogramas e dados estatísticos para $\sigma_v^2 = -20$ dB. (a) $N = 9$ sensores (b) $N = 12$ sensores

B.1. Histogramas Correspondes ao Caso de Estudo

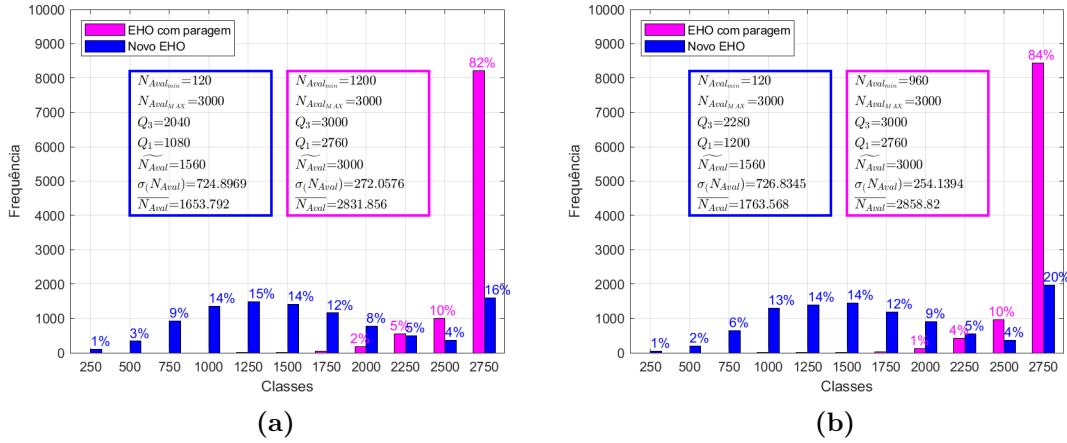


Figura B.6: Histogramas e dados estatísticos para $\sigma_v^2 = -15$ dB. (a) $N = 9$ sensores (b) $N = 12$ sensores

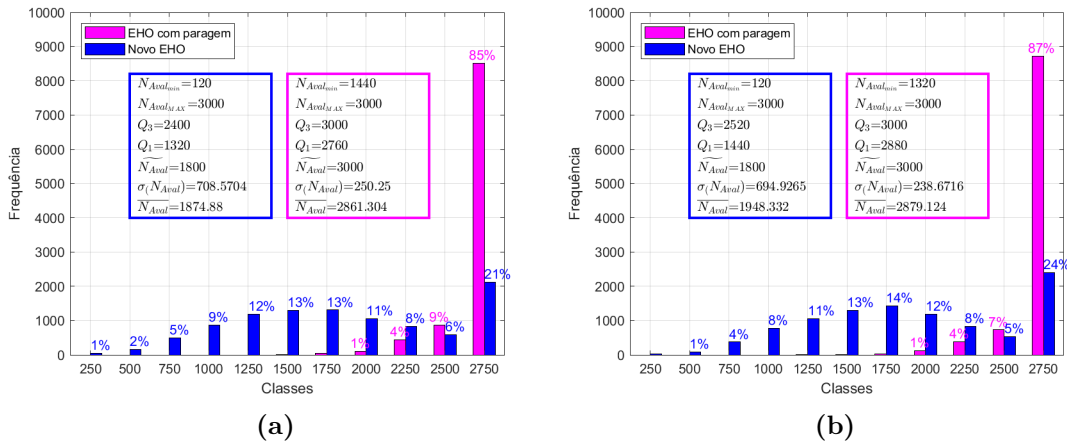


Figura B.7: Histogramas e dados estatísticos para $\sigma_v^2 = -10$ dB. (a) $N = 9$ sensores (b) $N = 12$ sensores

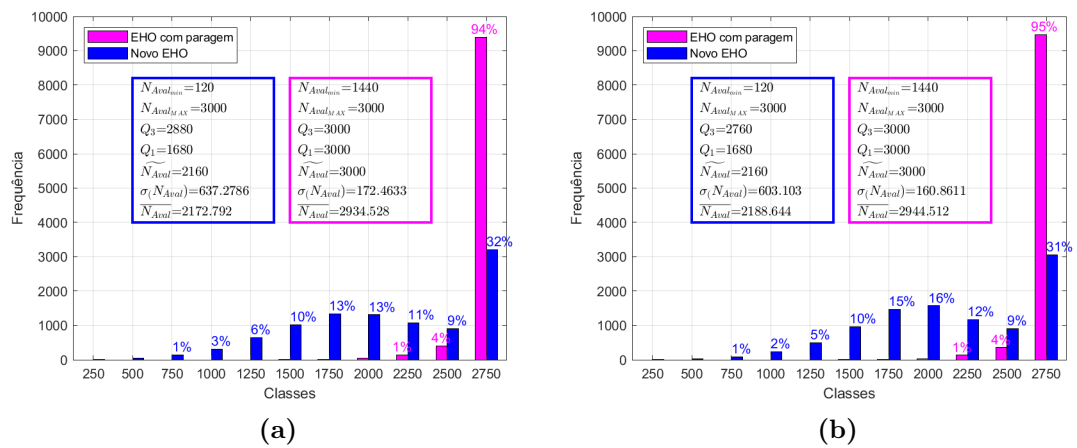


Figura B.8: Histogramas e dados estatísticos para $\sigma_v^2 = -5$ dB. (a) $N = 9$ sensores (b) $N = 12$ sensores

C

Histogramas e Indicadores Estatísticos

C.1 Extensão dos Resultados a $N=6$ e $N=15$

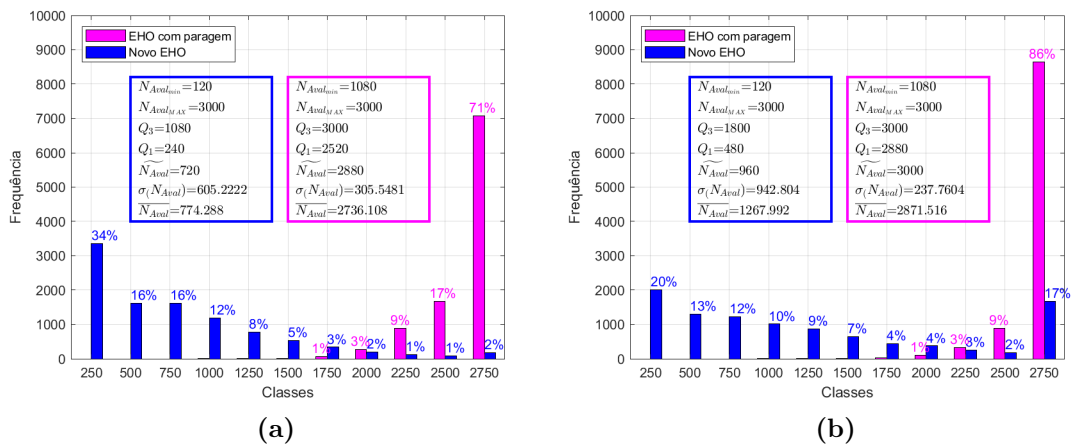


Figura C.1: Histogramas e dados estatísticos para $\sigma_v^2 = -40$ dB. (a) $N = 6$ sensores (b) $N = 15$ sensores

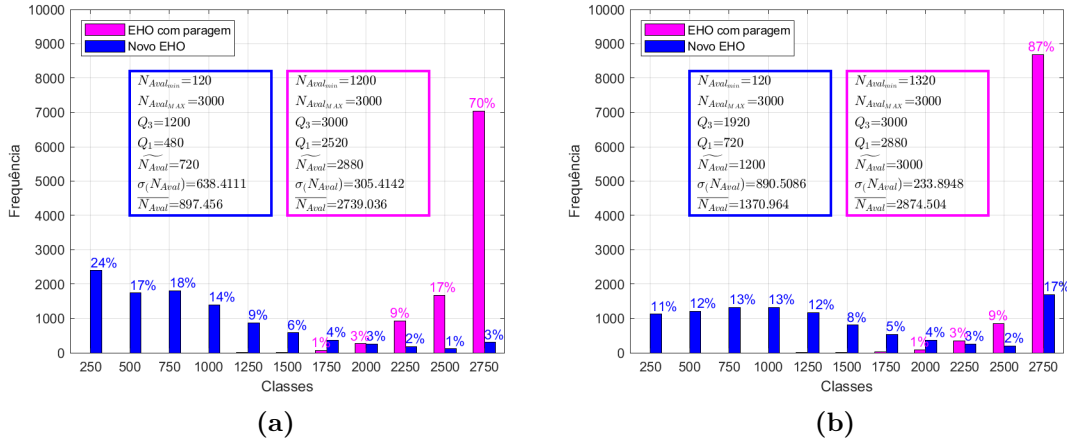


Figura C.2: Histogramas e dados estatísticos para $\sigma_v^2 = -35$ dB. (a) $N = 6$ sensores (b) $N = 15$ sensores

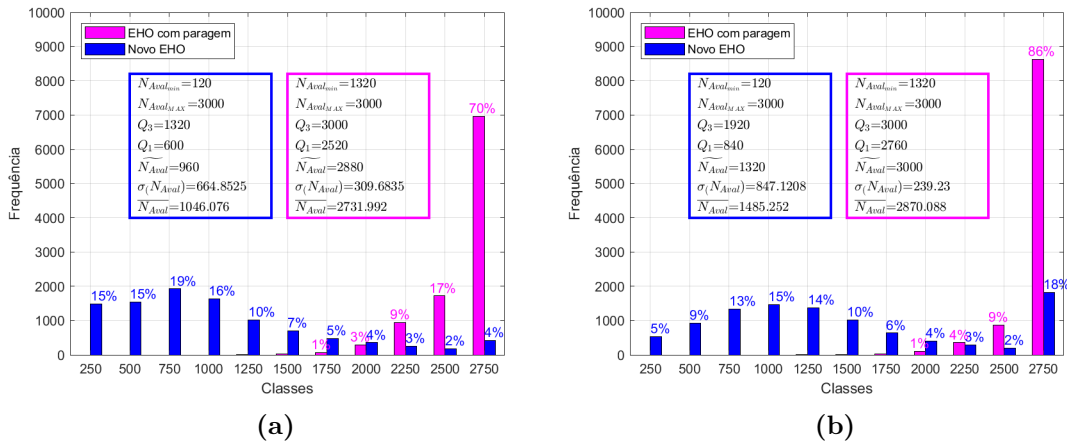


Figura C.3: Histogramas e dados estatísticos para $\sigma_v^2 = -30$ dB. (a) $N = 6$ sensores (b) $N = 15$ sensores

C. Histogramas e Indicadores Estatísticos

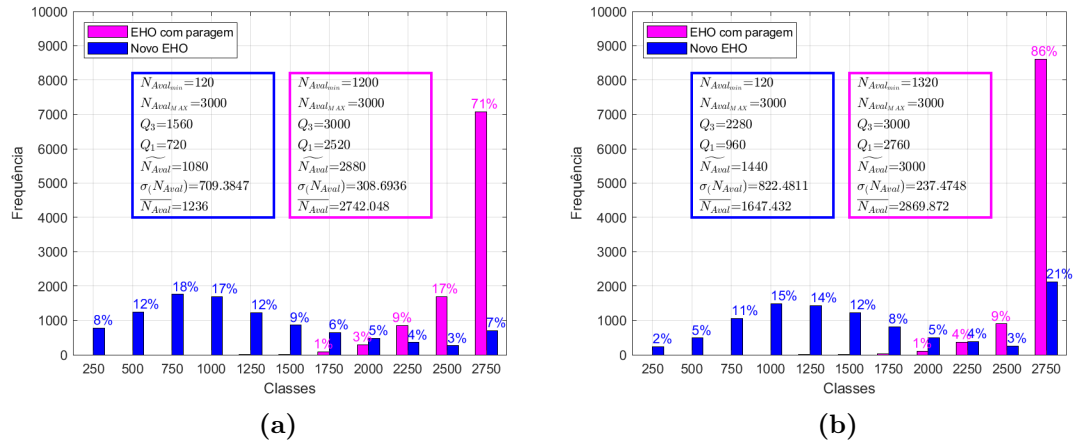


Figura C.4: Histogramas e dados estatísticos para $\sigma_v^2 = -25$ dB. (a) $N = 6$ sensores (b) $N = 15$ sensores

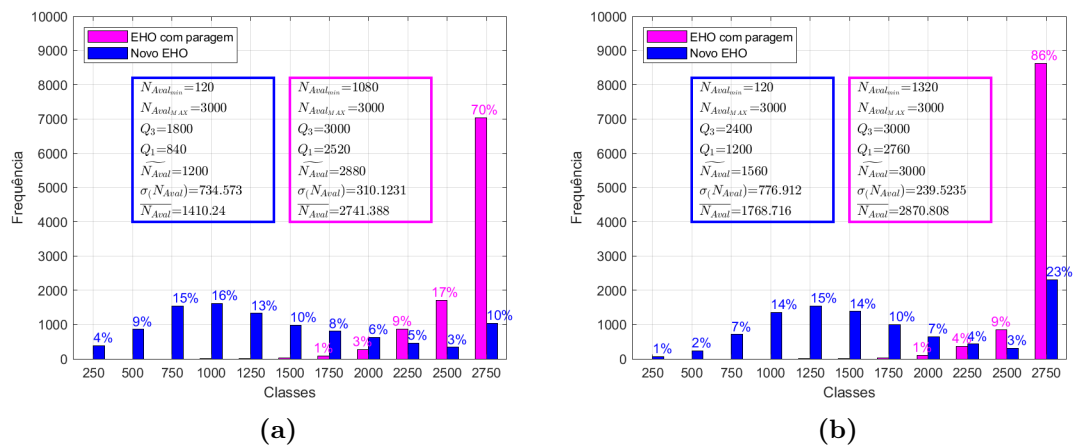


Figura C.5: Histogramas e dados estatísticos para $\sigma_v^2 = -20$ dB. (a) $N = 6$ sensores (b) $N = 15$ sensores

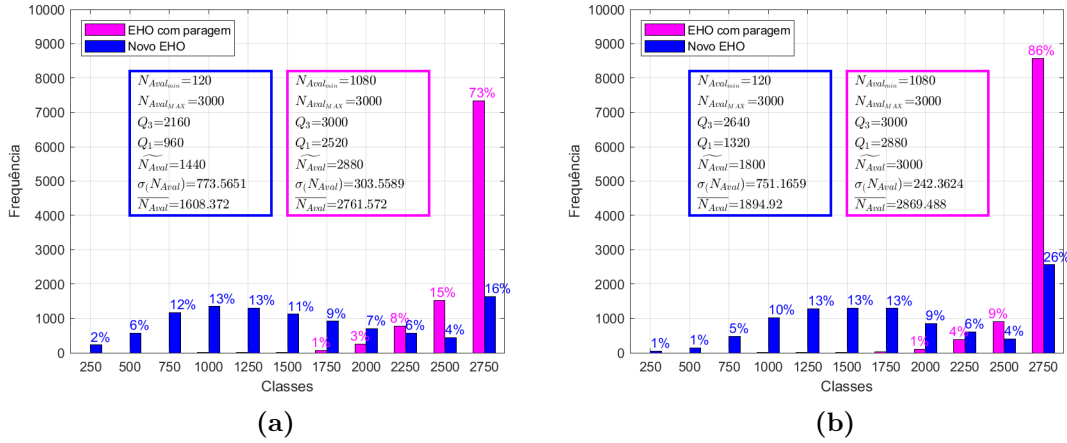


Figura C.6: Histogramas e dados estatísticos para $\sigma_v^2 = -15$ dB. (a) $N = 6$ sensores (b) $N = 15$ sensores

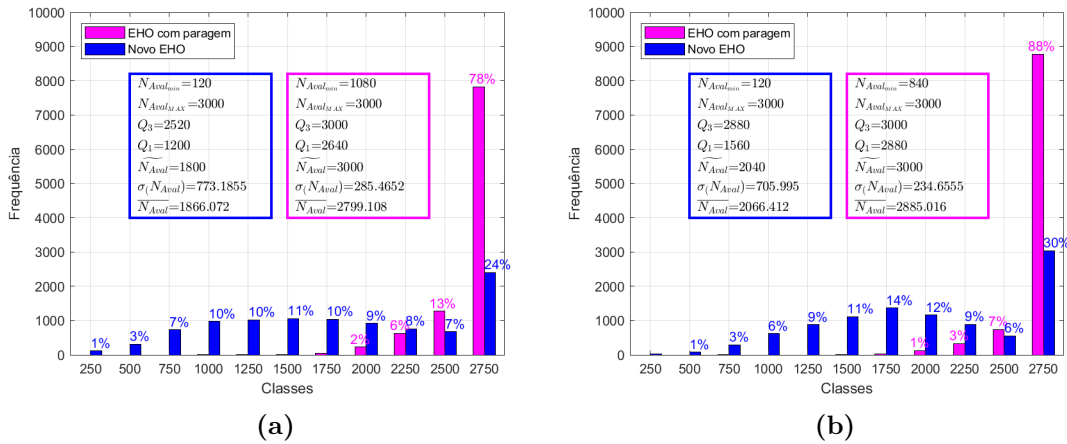


Figura C.7: Histogramas e dados estatísticos para $\sigma_v^2 = -10$ dB. (a) $N = 6$ sensores (b) $N = 15$ sensores

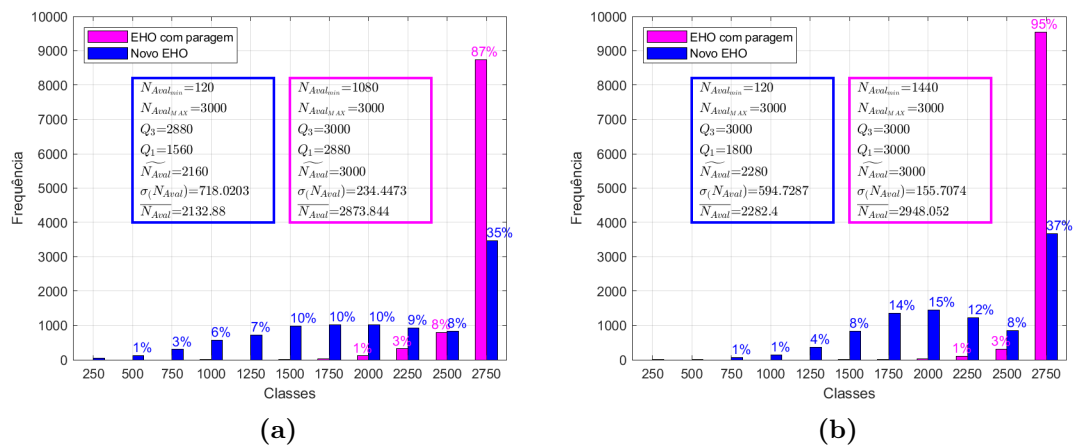


Figura C.8: Histogramas e dados estatísticos para $\sigma_v^2 = -5$ dB. (a) $N = 6$ sensores (b) $N = 15$ sensores

Referências

- [1] K. M. Larson. “Unanticipated Uses of the Global Positioning System”. In: *Annual Review of Earth and Planetary Sciences* 47.1 (2019), pp. 19–40.
- [2] A. LaMarca and E. de Lara. *Location Systems: An Introduction to the Technology Behind Location Awareness*. 3(1). Morgan & Claypool Publishers, 2008, pp. 1–122.
- [3] M. Weiser. *The Computer for the 21st Century*. Vol. 265. 3. 1991, pp. 94–104.
- [4] V. Realinho, T. Romão, and A. E. Dias. “An event-driven workflow framework to develop context-aware mobile applications”. In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia - MUM '12* (2012).
- [5] K. Kjær. “Designing middleware for context awareness in agriculture”. In: *Proceedings of the MDS'08* (2008), pp. 19–24.
- [6] N. D. Lane et al. “A survey of mobile phone sensing”. In: *IEEE Communications Magazine* 48.9 (July 2010), pp. 140–150.
- [7] N. Sabor, S. Sasaki, and M. Abo-Zahhad M. S. Ahmed. “A Comprehensive Survey on Hierarchical-Based Routing Protocols for Mobile Wireless Sensor Networks: Review, Taxonomy, and Future Directions”. In: *Wireless Communications and Mobile Computing 2017* (2017), pp. 1–23.
- [8] L. Atzori, A. Iera, and G. Morabito. “The Internet of Things: A survey”. In: *Computer Networks* 54.15 (2010), pp. 2787–2805.
- [9] E. Mumolo, M. Nolich, and G. VerCELLI. “Algorithms for acoustic localization based on microphone array in service robotics”. In: *Robotics and Autonomous Systems* 42.2 (2003), pp. 69–88.
- [10] M. Erol-Kantarci, H. T. Mouftah, and S. Oktug. “A Survey of Architectures and Localization Techniques for Underwater Acoustic Sensor Networks”. In: *IEEE Communications Surveys Tutorials* 13.3 (Mar. 2011), pp. 487–502.
- [11] Z. Wu and X. Li. “An improved underwater acoustic network localization algorithm”. In: *China Communications* 12.3 (Mar. 2015), pp. 77–83.
- [12] J. Choi et al. “Robust Directional Angle Estimation of Underwater Acoustic Sources Using a Marine Vehicle”. In: *Sensors* 18.9 (2018).
- [13] S. E. Küçükbay, M. Sert, and A. Yazici. “Use of Acoustic and Vibration Sensor Data to Detect Objects in Surveillance Wireless Sensor Networks”. In: *2017 21st International Conference on Control Systems and Computer Science (CSCS)*. May 2017, pp. 207–212.
- [14] J. Lim. “Detection Algorithm for Multi-Vehicular Target Tracking in Wireless Acoustic Sensor Networks”. In: *2006 International Forum on Strategic Technology*. Oct. 2006, pp. 142–145.

-
- [15] T. C. Collier, A. N. G. Kirschel, and C. E. Taylor. “Acoustic localization of antbirds in a Mexican rainforest using a wireless sensor network”. In: *Acoustical Society of America Journal* 128.1 (Jan. 2010), pp. 182–189.
- [16] I. Strumberger et al. “Static drone placement by elephant herding optimization algorithm”. In: *2017 25th Telecommunication Forum (TELFOR)* (2017).
- [17] D. Pedro et al. “Localization of Static Remote Devices Using Smartphones”. In: *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. June 2018, pp. 1–5.
- [18] D. Pedro et al. “Algorithms for Estimating the Location of Remote Nodes Using Smartphones”. In: *IEEE Access* 7 (2019), pp. 33713–33727.
- [19] S. Tomic and M. Beko. “A Robust NLOS Bias Mitigation Technique for RSS-TOA-Based Target Localization”. In: *IEEE Signal Processing Letters* 26.1 (Jan. 2019), pp. 64–68.
- [20] S. Tomic et al. “Target Localization in NLOS Environments Using RSS and TOA Measurements”. In: *IEEE Wireless Communications Letters* 7.6 (Dec. 2018), pp. 1062–1065.
- [21] S. Tomic et al. “On Hybrid RSS/TOA Target Localization in NLOS Environments”. In: *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*. June 2018, pp. 1471–1476.
- [22] S. Tomic and M. Beko. “A bisection-based approach for exact target localization in NLOS environments”. In: *Signal Processing* 143 (2018), pp. 328–335.
- [23] E. Shakshuki et al. “Comparative Study on Range Free Localization Algorithms”. In: *Procedia Computer Science* 151 (2019), pp. 501–510.
- [24] S. P. Singh and S. C. Sharma. “Range Free Localization Techniques in Wireless Sensor Networks: A Review”. In: *Procedia Computer Science* 57 (2015). 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), pp. 7–16.
- [25] Y. Ahmadi, N. Neda, and R. Ghazizadeh. “Range Free Localization in Wireless Sensor Networks for Homogeneous and Non-Homogeneous Environment”. In: *IEEE Sensors Journal* 16.22 (Nov. 2016), pp. 8018–8026.
- [26] R. Stoleru, T. He, and J. A. Stankovic. “Range-Free Localization”. In: *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*. Ed. by R. Poovendran, S. Roy, and C. Wang. Boston, MA: Springer US, 2007, pp. 3–31.
- [27] J. Zheng and A. Deghani. “Range-Free Localization in Wireless Sensor Networks with Neural Network Ensembles”. In: *Journal of Sensor and Actuator Networks* 1.3 (2012), pp. 254–271.
- [28] J. Ming et al. “Study on the personnel localization algorithm of the underground mine based on RSSI technology”. In: *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*. May 2017, pp. 408–411.
- [29] X. Shen et al. “Improved localization algorithm based on RSSI in low power Bluetooth network”. In: *2016 2nd International Conference on Cloud Computing and Internet of Things (CCIOT)*. Nov. 2016, pp. 134–137.
-

- [30] Y. Zou, Q. Wan, and H. Liu. “Semidefinite Programming for Tdoa Localization with Locally Synchronized Anchor Nodes”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2018, pp. 3524–3528.
- [31] F. Nesta, P. Svaizer, and M. Omologo. “Robust two-channel TDOA estimation for multiple speaker localization by using recursive ICA and a state coherence transform”. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. Apr. 2009, pp. 4597–4600.
- [32] S. Tomic et al. “A Robust Bisection-Based Estimator for TOA-Based Target Localization in NLOS Environments”. In: *IEEE Communications Letters* 21.11 (Nov. 2017), pp. 2488–2491.
- [33] R. P and M. L. Sichitiu. “Angle of Arrival Localization for Wireless Sensor Networks”. In: *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*. Vol. 1. July 2006, pp. 374–382.
- [34] P. Kulakowski et al. “Angle-of-arrival localization based on antenna arrays for wireless sensor networks”. In: *Computers and Electrical Engineering* 36.6 (2010), pp. 1181–1186.
- [35] S. Tomic, M. Beko, and M. Tuba. “A Linear Estimator for Network Localization Using Integrated RSS and AOA Measurements”. In: *IEEE Signal Processing Letters* 26.3 (Mar. 2019), pp. 405–409.
- [36] S. Tomic and M. Beko. “Target Localization via Integrated and Segregated Ranging Based on RSS and TOA Measurements”. In: *Sensors* 19.2 (2019).
- [37] S. Tomic et al. “On Target Localization Using Combined RSS and AoA Measurements”. In: *Sensors* 18.4 (2018).
- [38] S. Tomic et al. “Bayesian methodology for target tracking using combined RSS and AoA measurements”. In: *Physical Communication* 25 (2017), pp. 158–166.
- [39] S. Tomic et al. “Target Tracking with Sensor Navigation Using Coupled RSS and AoA Measurements”. In: *Sensors* 17.11 (2017).
- [40] S. Tomic, M. Beko, and R. Dinis. “RSS-Based Localization in Wireless Sensor Networks Using Convex Relaxation: Noncooperative and Cooperative Schemes”. In: *IEEE Transactions on Vehicular Technology* 64.5 (May 2015), pp. 2037–2050.
- [41] R. W. Ouyang, A. K. Wong, and C. Lea. “Received Signal Strength-Based Wireless Localization via Semidefinite Programming: Noncooperative and Cooperative Schemes”. In: *IEEE Transactions on Vehicular Technology* 59.3 (Mar. 2010), pp. 1307–1318.
- [42] N. Patwari et al. “Locating the nodes: cooperative localization in wireless sensor networks”. In: *IEEE Signal Processing Magazine* 22.4 (July 2005), pp. 54–69.
- [43] P. T. Daely and D. Kim. “Bio-Inspired Cooperative Localization in Industrial Wireless Sensor Network”. In: *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)*. May 2019, pp. 1–4.
- [44] H. Wymeersch, J. Lien, and M. Z. Win. “Cooperative Localization in Wireless Networks”. In: *Proceedings of the IEEE* 97.2 (Feb. 2009), pp. 427–450.

-
- [45] S. Tomic et al. “Cooperative localization in wireless sensor networks using combined measurements”. In: *2015 23rd Telecommunications Forum Telfor (TELFOR)*. Nov. 2015, pp. 488–491.
- [46] C. Alippi and G. Vanini. “A RSSI-based and calibrated centralized localization technique for wireless sensor networks”. In: *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW’06)*. Mar. 2006, pp. 5–305.
- [47] Q. Zhang et al. “A new centralized localization algorithm for wireless sensor network”. In: *2008 Third International Conference on Communications and Networking in China*. Aug. 2008, pp. 625–629.
- [48] W. Yu, N. D. Gaubitch, and R. Heusdens. “Distributed Tdoa-Based Indoor Source Localisation”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2018, pp. 6887–6891.
- [49] C. H. Yu, S. H. Min, and J. W. Choi. “Sensor localization-based distributed target tracking filter in underwater sensor networks”. In: *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. July 2015, pp. 706–711.
- [50] L. Dan and W. Xian-bin. “An improved PSO algorithm for distributed localization in wireless sensor networks”. In: *International Conference on Software Intelligence Technologies and Applications International Conference on Frontiers of Internet of Things 2014*. Dec. 2014, pp. 184–189.
- [51] S. Tomic, M. Beko, and R. Dinis. “Distributed RSS-Based Localization in Wireless Sensor Networks Based on Second-Order Cone Programming”. In: *Sensors* 14.10 (2014), pp. 18410–18432.
- [52] J. Fang et al. “Sequential Localization of Sensor Networks”. In: *SIAM Journal on Control and Optimization* 48.1 (2009), pp. 321–350.
- [53] J. Fang, D. Duncan, and A. S. Morse. “Sequential localization with inaccurate measurements”. In: *2009 American Control Conference*. June 2009, pp. 1970–1975.
- [54] J. Kennedy and R. Eberhart. “Particle swarm optimization”. In: *Proceedings of ICNN’95 - International Conference on Neural Networks*. Vol. 4. Nov. 1995, pp. 1942–1948.
- [55] M. R. Bonyadi and Z. Michalewicz. “Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review”. In: *Evolutionary Computation* 25.1 (Mar. 2017), pp. 1–54.
- [56] T. O. Ting et al. “Hybrid Metaheuristic Algorithms: Past, Present, and Future”. In: *Recent Advances in Swarm Intelligence and Evolutionary Computation*. Ed. by X. S. Yang. Springer International Publishing, 2015, pp. 71–83.
- [57] M. Jain, V. Singh, and A. Rani. “A novel nature-inspired algorithm for optimization: Squirrel search algorithm”. In: *Swarm and Evolutionary Computation* 44 (2019), pp. 148–175.
- [58] S. Shadravan, H. R. Naji, and V. K. Bardsiri. “The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems”. In: *Engineering Applications of Artificial Intelligence* 80 (2019), pp. 20–34.
-

- [59] Q. Luo, X. Yang, and Y. Zhou. “Nature-inspired approach: An enhanced moth swarm algorithm for global optimization”. In: *Mathematics and Computers in Simulation* 159 (2019), pp. 57–92.
- [60] I. Strumberger et al. “Wireless Sensor Network Localization Problem by Hybridized Moth Search Algorithm”. In: *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*. June 2018, pp. 316–321.
- [61] I. Strumberger et al. “Hybridized moth search algorithm for constrained optimization problems”. In: *2018 International Young Engineers Forum (YEF-ECE)*. May 2018, pp. 1–5.
- [62] G. Wang, S. Deb, and Z. Cui. “Monarch butterfly optimization”. In: *Neural Computing and Applications* 31.7 (July 2019), pp. 1995–2014.
- [63] I. Strumberger et al. “Modified Monarch Butterfly Optimization Algorithm for RFID Network Planning”. In: *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*. May 2018, pp. 1–6.
- [64] I. Strumberger et al. “Monarch butterfly optimization algorithm for localization in wireless sensor networks”. In: *2018 28th International Conference Radioelektronika (RADIOELEKTRONIKA)*. Apr. 2018, pp. 1–6.
- [65] L. Cao, Y. Cai, and Y. Yue. “Swarm Intelligence-Based Performance Optimization for Mobile Wireless Sensor Networks: Survey, Challenges, and Future Directions”. In: *IEEE Access* (2019), pp. 161524–161553.
- [66] M. Dorigo and T. Stützle. *Ant Colony Optimization*. Scituate, MA, USA: Bradford Company, 2004.
- [67] D. Karaboga and B. Basturk. “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”. In: *Journal of Global Optimization* 39.3 (Nov. 2007), pp. 459–471.
- [68] X. S. Yang. “A New Metaheuristic Bat-Inspired Algorithm”. In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Ed. by J. R. González et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 65–74.
- [69] D. Teodorović. “Bee Colony Optimization (BCO)”. In: *Innovations in Swarm Intelligence*. Ed. by C. P. Lim, L. C. Jain, and S. Dehuri. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 39–60.
- [70] R. Tang et al. “Wolf search algorithm with ephemeral memory”. In: *Seventh International Conference on Digital Information Management (ICDIM 2012)*. Aug. 2012, pp. 165–172.
- [71] S. C. Chu, P. W. Tsai, and J. S. Pan. “Cat Swarm Optimization”. In: *PRICAI 2006: Trends in Artificial Intelligence*. Ed. by Q. Yang and G. Webb. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 854–858.
- [72] X. Yang and Suash Deb. “Cuckoo Search via Lévy flights”. In: *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*. Dec. 2009, pp. 210–214.
- [73] X. S. Yang. “Firefly Algorithm, Stochastic Test Functions and Design Optimisation”. In: *Int. J. Bio-Inspired Comput.* 2.2 (Mar. 2010), pp. 78–84.

-
- [74] A. R. Kammerdiner, A. Mucherino, and P. M. Pardalos. “Application of Monkey Search Meta-heuristic to Solving Instances of the Multidimensional Assignment Problem”. In: *Optimization and Cooperative Control Strategies*. Ed. by M. J. Hirsch et al. Springer Berlin Heidelberg, 2009, pp. 385–397.
- [75] M. Beko. “Energy-based localization in wireless sensor networks using semidefinite relaxation”. In: *2011 IEEE Wireless Communications and Networking Conference*. 2011, pp. 1552–1556.
- [76] M. Marko. “Energy-Based Localization in Wireless Sensor Networks Using Second-Order Cone Programming Relaxation”. In: *Wireless Personal Communications* 77.3 (Feb. 2014), pp. 1847–1857.
- [77] K. C. Ho and M. Sun. “An accurate algebraic closed-form solution for energy-based source localization”. In: *IEEE Transactions on Audio, Speech and Language Processing* 15.8 (2007), pp. 2542–2550.
- [78] C. Meesookho, U. Mitra, and S. Narayanan. “On energy-based acoustic source localization for sensor networks”. In: *IEEE Transactions on Signal Processing* 56.1 (2008), pp. 365–377.
- [79] G. Wang and K. Yang. “A Semidefinite Relaxation Method for Energy-based Source Localization in Sensor Networks”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2009)*. 4. 2009, pp. 2257–2260.
- [80] M. Rudafshani and S. Datta. “Localization in Wireless Sensor Networks”. In: *6th International Symposium on Information Processing in Sensor Networks (IPSN 2007)*. 2007, pp. 51–60.
- [81] G. Wang. “A Semidefinite Relaxation Method for Energy-Based Source Localization in Sensor Networks”. In: *IEEE Transactions on Vehicular Technology* 60.5 (2011), pp. 2293–2301.
- [82] D. Li and Y. H. Hu. “Energy Based Collaborative Source Localization Using Acoustic Micro-Sensor Array”. In: *Applied Signal Processing* 2003.4 (2003), pp. 321–337.
- [83] X. Sheng and Y. H. Hu. “Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks”. In: *IEEE Transactions on Signal Processing* 53.1 (2005), pp. 44–53.
- [84] G. Wang, S. Deb, and L. Dos Santos Coelho. “Elephant Herding Optimization”. In: *3rd International Symposium on Computational and Business Intelligence (ISCBI2015)*. 2016, pp. 1–5.
- [85] G. Wang et al. “A New Metaheuristic Optimisation Algorithm Motivated by Elephant Herding Behaviour”. In: *Int. J. Bio-Inspired Comput.* 8.6 (2017), pp. 394–409.
- [86] S. Deb. “Elephant Search Algorithm for Optimization Problems”. In: *10th International Conference on Digital Information Management (ICDIM)*. 2015, pp. 249–255.
- [87] I. Strumberger, N. Bacanin, and M. Tuba. “Hybridized Elephant Herding Optimization Algorithm for Constrained Optimization”. In: *Hybrid Intelligent Systems Advances in Intelligent Systems and Computing* (2018), pp. 158–166.
-

- [88] I. Strumberger et al. “Performance of Elephant Herding Optimization and Tree Growth Algorithm Adapted for Node Localization in Wireless Sensor Networks”. In: *Sensors* 19.11 (2019).
- [89] S. Gupta et al. “Elephant herding optimization based PID controller tuning”. In: *International Journal of Advanced Technology and Engineering Exploration* 3.24 (2016), pp. 2394–5443.
- [90] S. S. Chibani and A. Tari. “Elephant Herding Optimization for Service Selection in QoS-Aware Web”. In: *International Journal of Computer and Information Engineering* 4.10 (2017).
- [91] M. Ghasemi et al. “Solving non-linear, non-smooth and non-convex optimal power flow problems using chaotic invasive weed optimization algorithms based on chaos”. In: *Energy* 73 (2014), pp. 340–353.
- [92] E. Tuba et al. “Support Vector Machine Optimized by Elephant Herding Algorithm for Erythematous-Squamous Diseases Detection”. In: *Procedia Computer Science* 122 (2017), pp. 916–923.
- [93] A. Alihodzic et al. “Unmanned aerial vehicle path planning problem by adjusted elephant herding optimization”. In: *2017 25th Telecommunication Forum (TELFOR)* (2017).
- [94] E. Tuba and Z. Stanimirovic. “Elephant herding optimization algorithm for support vector machine parameters tuning”. In: *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)* (2017).
- [95] E. Tuba, A. Alihodzic, and M. Tuba. “Multilevel image thresholding using elephant herding optimization algorithm”. In: *2017 14th International Conference on Engineering of Modern Electric Systems (EMES)* (2017).
- [96] N. K. Meena et al. “Improved Elephant Herding Optimization for Multiobjective DER Accommodation in Distribution Systems”. In: *IEEE Transactions on Industrial Informatics* 14.3 (2018), pp. 1029–1039.
- [97] I. Fatima et al. “Efficient Demand Side Management Using Hybridization of Elephant Herding Optimization and Firefly Optimization”. In: *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. IEEE Computer Society, May 2018, pp. 839–845.
- [98] A. E. Hassanien et al. “Intelligent human emotion recognition based on elephant herding optimization tuned support vector regression”. In: *Biomedical Signal Processing and Control* 45 (2018), pp. 182–191.
- [99] K. Parvez et al. “Scheduling of Appliances in HEMS Using Elephant Herding Optimization and Harmony Search Algorithm”. In: *Advances on Broad-Band Wireless Computing, Communication and Applications*. Ed. by L. Barolli, F. Khafa, and J. Conesa. Springer International Publishing, 2018, pp. 62–72.
- [100] F. Chakraborty, P. K. Roy, and D. Nandi. “Oppositional elephant herding optimization with dynamic Cauchy mutation for multilevel image thresholding”. In: *Evolutionary Intelligence* 12.3 (2019), pp. 445–467.
- [101] M. Jafari, E. Salajegheh, and J. Salajegheh. “An efficient hybrid of elephant herding optimization and cultural algorithm for optimal design of trusses”. In: *Engineering with Computers* 35.3 (2019), pp. 781–801.

-
- [102] A. H. Alavi W. Li Gai-Ge Wang. “Learning-based elephant herding optimization algorithm for solving numerical optimization problems”. In: *Knowledge-Based Systems* (2020).
- [103] R. M. Sahoo and S. K. P. Kumari. “Elephant Herding Optimization for Multiprocessor Task Scheduling in Heterogeneous Environment”. In: *Computational Intelligence in Pattern Recognition Advances in Intelligent Systems and Computing* (2020), pp. 217–229.
- [104] I. Strumberger et al. “Hybrid Elephant Herding Optimization Approach for Cloud Computing Load Scheduling”. In: *Communications in Computer and Information Science Swarm, Evolutionary, and Memetic Computing and Fuzzy and Neural Computing* (2020), pp. 201–212.
- [105] S. S. Dhillon et al. “Automatic Generation Control of Interconnected Power Systems Using Elephant Herding Optimization”. In: *Intelligent Computing Techniques for Smart Energy Systems*. Singapore: Springer Singapore, 2020, pp. 9–18.
- [106] H. Moayedi, M. A. Mu’azu, and L. K. Foong. “Novel swarm-based approach for predicting the cooling load of residential buildings based on social behavior of elephant herds”. In: *Energy and Buildings* 206 (2020), p. 109579.
- [107] L. E. Kinsler et al. *Fundamentals of Acoustics*. 4th. 1999.
- [108] G. Sun et al. “Signal processing techniques in network-aided positioning: A survey of state-of-the-art positioning designs”. In: *IEEE Signal Processing Magazine* 22.4 (2005), pp. 12–23.
- [109] Z. Deng et al. “Situation and development tendency of indoor positioning”. In: *China Communications* 10.3 (2013), pp. 42–55.
- [110] D. Blatt and A. O. Hero. “Energy Based Sensor Network Source Localization via Projection onto Convex Sets (POCS)”. In: *IEEE Transactions on Signal Processing* 54.2 (2006), pp. 3614–3619.
- [111] Charles Darwin. “Ch. VIII. - Darwin, and the Origin of Species”. In: *The Foundations of Zoölogy* (1899), pp. 183–214.
- [112] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [113] M. Mitchell. *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.
- [114] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. “Optimization by Simulated Annealing”. In: *Science* 220.4598 (1983), pp. 671–680.
- [115] L. Ingber. “Simulated annealing: Practice versus theory”. In: *Mathematical and Computer Modelling* 18.11 (1993), pp. 29–57.
- [116] L. Ingber. “Very fast simulated re-annealing”. In: *Mathematical and Computer Modelling* 12.8 (1989), pp. 967–973.
- [117] L. Ingber and B. Rosen. “Genetic Algorithms and Very Fast Simulated Reannealing: A comparison”. In: *Mathematical and Computer Modelling* 16.11 (1992), pp. 87–100.
-

- [118] D. Yuret and M. de la Maza. “Dynamic Hill Climbing: Overcoming the limitations of optimization techniques”. In: *In The Second Turkish Symposium on Artificial Intelligence and Neural Networks*. 1993, pp. 208–212.
- [119] D. Yuret. *From Genetic Algorithms to Efficient Optimization*. Tech. rep. Cambridge, MA, USA, 1994.
- [120] K. Bousson. “Efficient global optimization based on dynamic canonical descent”. In: *Systems Science* 26 (Jan. 2000), pp. 61–78.
- [121] K. Bousson. “Trajectory Optimization by Canonical Dynamic Programming”. In: *55th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*. 2004.
- [122] K. Bousson and S. D. Correia. “Optimization algorithm based on densification and dynamic canonical descent”. In: *Journal of Computational and Applied Mathematics* 191.2 (2006). The International Conference on Computational Methods in Sciences and Engineering 2004, pp. 269–279.
- [123] Y. Tan and Y. Zhu. “Fireworks Algorithm for Optimization”. In: *Advances in Swarm Intelligence*. Ed. by Y. Tan, Y. Shi, and K. C. Tan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 355–364.
- [124] S. Zheng, A. Janecek, and Y. Tan. “Enhanced Fireworks Algorithm”. In: *2013 IEEE Congress on Evolutionary Computation*. June 2013, pp. 2069–2077.
- [125] S. Zheng et al. “Dynamic search in fireworks algorithm”. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. July 2014, pp. 3222–3229.
- [126] J. Li, S. Zheng, and Y. Tan. “Adaptive Fireworks Algorithm”. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. July 2014, pp. 3214–3221.
- [127] S. Zheng et al. “Exponentially decreased dimension number strategy based dynamic search fireworks algorithm for solving CEC2015 competition problems”. In: *2015 IEEE Congress on Evolutionary Computation (CEC)*. May 2015, pp. 1083–1090.
- [128] C. Yu, L. C. Kelley, and Y. Tan. “Dynamic search fireworks algorithm with covariance mutation for solving the CEC 2015 learning based competition problems”. In: *2015 IEEE Congress on Evolutionary Computation (CEC)*. May 2015, pp. 1106–1112.
- [129] C. Yu and Y. Tan. “Fireworks algorithm with covariance mutation”. In: *2015 IEEE Congress on Evolutionary Computation (CEC)*. May 2015, pp. 1250–1256.
- [130] S. Zheng et al. “A Cooperative Framework for Fireworks Algorithm”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 14.1 (Jan. 2017), pp. 27–41.
- [131] J. Guerreiro et al. “Using the Fireworks Algorithm for ML Detection of Nonlinear OFDM”. In: *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. Sept. 2017, pp. 1–5.
- [132] E. Tuba, M. Tuba, and M. Beko. “Node localization in ad hoc wireless sensor networks using fireworks algorithm”. In: *2016 5th International Conference on Multimedia Computing and Systems (ICMCS)*. Sept. 2016, pp. 223–229.

-
- [133] E. Tuba, M. Tuba, and M. Beko. “Support Vector Machine Parameters Optimization by Enhanced Fireworks Algorithm”. In: *Advances in Swarm Intelligence*. Ed. by Y. Tan, Y. Shi, and B. Niu. Springer International Publishing, 2016, pp. 526–534.
- [134] M. Tuba, N. Bacanin, and M. Beko. “Fireworks algorithm for RFID network planning problem”. In: *2015 25th International Conference Radioelektronika (RADIOELEKTRONIKA)*. Apr. 2015, pp. 440–444.
- [135] H. M. Ali et al. “Optimising the power using firework-based evolutionary algorithms for emerging IoT applications”. In: *IET Networks* 8.1 (2019), pp. 15–31.
- [136] Y. Miao et al. “Mobile Robot Odor Source Localization Based on Modified FWA”. In: *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. July 2018, pp. 854–860.
- [137] N. Bacanin, M. Tuba, and M. Beko. “Hybridized Fireworks Algorithm for Global Optimization”. In: vol. 41. 2015, pp. 108–114.
- [138] I. Strumberger et al. “Hybridized Artificial Bee Colony Algorithm for Constrained Portfolio Optimization Problem”. In: *2018 IEEE Congress on Evolutionary Computation (CEC)*. July 2018, pp. 1–8.
- [139] D. Karaboga and B. Gorkemli. “A combinatorial Artificial Bee Colony algorithm for traveling salesman problem”. In: *2011 International Symposium on Innovations in Intelligent Systems and Applications*. July 2011, pp. 50–53.
- [140] C. Ozturk and D. Karaboga. “Hybrid Artificial Bee Colony algorithm for neural network training”. In: *2011 IEEE Congress of Evolutionary Computation (CEC)*. June 2011, pp. 84–88.
- [141] P. Bhattacharjee et al. “Multi-robot path-planning using artificial bee colony optimization algorithm”. In: *2011 Third World Congress on Nature and Biologically Inspired Computing*. Nov. 2011, pp. 219–224.
- [142] M. Tuba, N. Bacanin, and M. Beko. “Multiobjective RFID Network Planning by Artificial Bee Colony Algorithm with Genetic Operators”. In: *Advances in Swarm and Computational Intelligence*. Ed. by Y. Tan et al. Springer International Publishing, 2015, pp. 247–254.
- [143] V. Tuba, M. Beko, and M. Tuba. “Performance of Elephant Herding Optimization Algorithm on CEC 2013 Real Parameter Single Objective Optimization”. In: *WSEAS TRANSACTIONS ON SYSTEMS* 16 (2017), pp. 100–105.
- [144] R. Zhao et al. “Selfish herd optimizer with levy-flight distribution strategy for global optimization problem”. In: *Physica A: Statistical Mechanics and its Applications* 538 (2020), pp. 122–687.
- [145] B. V. Chawda and J. M. Patel. “Natural Computing Algorithms – A Survey”. In: *International Journal of Emerging Technology and Advanced Engineering* 6.6 (2016), pp. 221–227.
- [146] A. Beck, P. Stoica, and J. Li. “Exact and Approximate Solutions of Source Localization Problems”. In: 56.5 (2008), pp. 1770–1778.
- [147] L. E. Kinsler, A. R. Frey, and G. S. Bennett. “Fundamentals of Acoustics”. In: *American Journal of Physics* 19.4 (1951).
-

- [148] Leo L. Beranek and Tim J. Mellow. In: *Acoustics: Sound Fields and Transducers*. Academic Press, 2012.
- [149] A. R. Barnard. “Quantifying acoustic sources through sound power measurements”. In: *Sound and Vibration* 48 (2014), pp. 8–11.
- [150] Yue Kan et al. “Passive Acoustic Source Localization at a Low Sampling Rate Based on a Five-Element Cross Microphone Array”. In: *Sensors* 15.6 (2015), pp. 13326–13347.
- [151] S. Amir Hoseini Sabzevari and Majid Moavenian. “Locating the acoustic source in thin glass plate using low sampling rate data”. In: *Ultrasonics* 70 (2016), pp. 1–11.
- [152] A. V. Oppenheim, S. Hamid Nawab, and Alan S. Willsky. *Signals and Systems*. Pearson Education (US), 1996.
- [153] Z. Ma and G. A. E. Vandenbosch. “Impact of Random Number Generators on the performance of particle swarm optimization in antenna design”. In: *6th European Conference on Antennas and Propagation (EUCAP)* (2012), pp. 925–929.
- [154] F. Koeune. “Pseudo-random number generator”. In: *Encyclopedia of Cryptography and Security*. Ed. by H. C. A. Van Tilborg. Boston, MA: Springer US, 2005, pp. 485–487.
- [155] M. Zhang, W. Zhang, and Y. Sun. “Chaotic co-evolutionary algorithm based on differential evolution and particle swarm optimization”. In: *IEEE International Conference on Automation and Logistics (ICAL’09)* (2009), pp. 885–889.
- [156] J. L. Guerrero, A. Berlanga, and J. M. Molina. “Initialization Procedures for Multiobjective Evolutionary Approaches to the Segmentation Issue”. In: *Lecture Notes in Computer Science Hybrid Artificial Intelligent Systems* (2012), pp. 452–463.
- [157] Qian Li, San-Yang Liu, and Xin-She Yang. “Influence of initialization on the performance of metaheuristic optimizers”. In: *Applied Soft Computing* (2020), p. 106193.
- [158] J. Kiefer and J. Wolfowitz. “Stochastic Estimation of the Maximum of a Regression Function”. In: *The Annals of Mathematical Statistics* 23.3 (1952), pp. 462–466.
- [159] German Correa Velez, Fernando Mesa, and Pedro Pablo Cardenas Alzate. “Linear search optimization through the Armijo rule method”. In: *Contemporary Engineering Sciences* 11.16 (2018), pp. 771–778.
- [160] J. Snyman. *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Applied Optimization. Springer US, 2005.
- [161] J. A. Snyman and D. N. Wilke. *Practical Mathematical Optimization: Basic Optimization Theory and Gradient-Based Algorithms*. Springer Optimization and Its Applications. Springer International Publishing, 2018.
- [162] G. B. Arfken. *Mathematical Methods for Physicists, Chapter: The Method of Steepest Descents*. Elsevier Science, 2013.
- [163] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. 1996.

-
- [164] P. Wolfe. “Convergence Conditions for Ascent Methods”. In: *SIAM Review* 11.2 (1969), pp. 226–235.
- [165] B. Christianson. “Global convergence using de-linked Goldstein or Wolfe linesearch conditions”. In: *Advanced Modeling and Optimization* 11.1 (2009), pp. 25–31.
- [166] M. Avriel. *Nonlinear Programming: Analysis and Methods*. Dover Books on Computer Science Series. Dover Publications, 2003.
- [167] R. Fletcher. “A new approach to variable metric algorithms”. In: *The Computer Journal* 13.3 (1970), pp. 317–322.
- [168] R. Fletcher. “An Overview of Unconstrained Optimization”. In: *Algorithms for Continuous Optimization* (1994), pp. 109–143.
- [169] R. Fletcher and M. J. D. Powell. “A Rapidly Convergent Descent Method for Minimization”. In: *The Computer Journal* 6.2 (1963), pp. 163–168.
- [170] R. Fletcher. “Function minimization by conjugate gradients”. In: *The Computer Journal* 7.2 (Jan. 1964), pp. 149–154.
- [171] A. Beck and M. Teboulle. “Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems”. In: *IEEE Transactions on Image Processing* 18.11 (2009), pp. 2419–2434.
- [172] V. S. Petrovic and C. S. Xydeas. “Gradient-Based Multiresolution Image Fusion”. In: *IEEE Transactions on Image Processing* 13.2 (2004), pp. 228–237.
- [173] Z. Wang and Y. Dai. “A New Gradient Annealing Algorithm (GAA) and its Applications in Path Planning of Mobile Robot”. In: *2007 IEEE International Conference on Automation and Logistics* (2007).
- [174] Z. Dongmei, W. Pei, and Z. Xu. “Reactive power optimization by genetic algorithm integrated with reduced gradient method”. In: *2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA)* (2014).
- [175] O. V. Barabash et al. “Dynamic models of decision support systems for controlling UAV by two-step variational-gradient method”. In: *2017 IEEE 4th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)* (2017).
- [176] I. Jusoh, M. Mamat, and M. Rivaie. “A new family of conjugate gradient methods for small-scale unconstrained optimization”. In: *AIP Conference Proceedings* 1522.1 (2013), pp. 1360–1365.
- [177] M. Rivaie, M. Fauzi, and M. Mamat. “New modifications of conjugate gradient coefficient with global convergence properties”. In: *2012 IEEE Symposium on Humanities, Science and Engineering Research* (2012).
- [178] B. Jiao, J. Han, and L. Chen. “A Modified Conjugate Gradient Algorithm with Sufficient Descent”. In: *2011 Fourth International Joint Conference on Computational Sciences and Optimization* (2011).
- [179] H. Gao, Z. Sun, and T. Zhu. “Another hybrid conjugate gradient method and its global convergence for unconstrained optimization”. In: *2011 Chinese Control and Decision Conference (CCDC)* (2011).
-

- [180] J. Yin, L. Wang, and X. Jiang. “A modified PRP conjugate gradient method with Armijo line search for large-scale unconstrained optimization”. In: *2017 36th Chinese Control Conference (CCC)* (2017).
- [181] A. Taflove and S. C. Hagness. *Computational Electrodynamics: The Finite-difference Time-domain Method*. Artech House antennas and propagation library. Artech House, 2005.
- [182] J. Noye. “Finite Difference Techniques for Partial Differential Equations”. In: *Computational Techniques for Differential Equations*. Ed. by J. Noye. Vol. 83. North-Holland Mathematics Studies. North-Holland, 1984, pp. 95–354.
- [183] S. D. Correia et al. “Elephant Herding Optimization for Energy-Based Localization”. In: *Sensors* 18.9 (2018).
- [184] A. A. K. Ismael et al. “Enhanced Elephant Herding Optimization for Global Optimization”. In: *IEEE Access* 7 (2019), pp. 34738–34752.
- [185] J. Pierozan and L. Dos Santos Coelho. “Coyote Optimization Algorithm: A New Metaheuristic for Global Optimization Problems”. In: *2018 IEEE Congress on Evolutionary Computation (CEC)*. July 2018, pp. 1–8.
- [186] X. Yuan et al. “Hybrid parallel chaos optimization algorithm with harmony search algorithm”. In: *Applied Soft Computing* 17 (2014), pp. 12–22.
- [187] E. Tuba et al. “Chaotic elephant herding optimization algorithm”. In: *2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMII)*. Feb. 2018, pp. 213–216.
- [188] A. Cheraghali, M. Hajiaghayi-Keshteli, and M. M. Paydar. “Tree Growth Algorithm (TGA): A novel approach for solving optimization problems”. In: *Engineering Applications of Artificial Intelligence* 72 (2018), pp. 393–414.
- [189] I. Strumberger et al. “Dynamic Search Tree Growth Algorithm for Global Optimization”. In: *Technological Innovation for Industry and Service Systems*. Ed. by L. M. Camarinha-Matos, R. Almeida, and J. Oliveira. Springer International Publishing, 2019, pp. 143–153.
- [190] J. Li et al. “Enhancing Elephant Herding Optimization with Novel Individual Updating Strategies for Large-Scale Optimization Problems”. In: *Mathematics* 7.5 (2019).
- [191] A. E. Mostafa et al. “On the performance improvement of elephant herding optimization algorithm”. In: *Knowledge-Based Systems* 166 (2019), pp. 58–70.
- [192] J. F. Gonçalves and M. G. C. Resende. “Biased random-key genetic algorithms for combinatorial optimization”. In: *Journal of Heuristics* 17.5 (2011), pp. 487–525.
- [193] H. Q. Mohammed et al. “Coyote optimization algorithm for parameters extraction of three-diode photovoltaic models of photovoltaic modules”. In: *Energy* 187 (2019), pp. 1–8.
- [194] U. Güvenç and E. Kaymaz. “Economic Dispatch Integrated Wind Power Using Coyote Optimization Algorithm”. In: *2019 7th International Istanbul Smart Grids and Cities Congress and Fair (ICSIG)*. Apr. 2019, pp. 179–183.
- [195] Z. Li et al. “Optimal Power Flow for Transmission Power Networks Using a Novel Metaheuristic Algorithm”. In: *Energies* 12.22 (2019).

-
- [196] S. Chib and E. Greenberg. “Understanding the Metropolis-Hastings Algorithm”. In: *The American Statistician* 49.4 (1995), pp. 327–335.
- [197] C. P. Robert and G. Casella. “The Metropolis—Hastings Algorithm”. In: *Springer Texts in Statistics Monte Carlo Statistical Methods* (1999), pp. 231–283.
- [198] L. Holden. “Geometric convergence of the Metropolis-Hastings simulation algorithm”. In: *Statistics and Probability Letters* 39.4 (1998), pp. 371–377.
- [199] F. J. Medina-Aguayo, A. Lee, and G. O. Roberts. “Stability of noisy Metropolis–Hastings”. In: *Statistics and Computing* 26.6 (Nov. 2016), pp. 1187–1211.
- [200] G. O. Roberts and A. F. M. Smith. “Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms”. In: *Stochastic Processes and their Applications* 49.2 (1994), pp. 207–216.
- [201] D. E. Knuth. *The Art of Computer Programming: Sorting and Searching*. 2nd. Vol. 3. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1998.
- [202] L. F. Williams. “A Modification to the Half-interval Search (Binary Search) Method”. In: *Proceedings of the 14th Annual Southeast Regional Conference*. ACM-SE 14. Birmingham, Alabama: ACM, 1976, pp. 95–101.
- [203] S. Bar and J. Tabrikian. “New observations on efficiency of variance estimation of white Gaussian signal with unknown mean”. In: *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*. July 2016, pp. 1–5.
- [204] P. J. G. Teunissen and A. R. Amiri-Simkooei. “Least-squares variance component estimation”. In: *Journal of Geodesy* 82.2 (2008), pp. 65–82.