

MHRTSN-WOA: Uma Estratégia Baseada em Whale Optimization para Navegação Robótica em Ambientes Estáticos e Dinâmicos

Micael Balza^{*†}, Vitor Y. F. Freitas^{*†}, Sérgio N. Silva^{*†§} e Marcelo A. C. Fernandes^{*†‡¶}

^{*}InovAI Lab, nPITI/IMD, UFRN, Natal, RN, Brasil

[†]Leading Advanced Technologies Center of Excellence (LANCE), nPITI/IMD, UFRN, Natal, RN, Brasil

[‡]Departamento de Engenharia Elétrica (DEE), UFCG, Campina Grande, PB, Brasil

[§]Departamento de Engenharia da Computação e Automação (DCA), UFRN, Natal, RN, Brasil

Email: micael.balza.108@ufrn.edu.br, vitor.freitas.123@ufrn.edu.br,

sergionatan@dee.ufcg.edu.br, mfernandes@dca.ufrn.br

Resumo—Este trabalho apresenta a estratégia MHRTSN-WOA, uma abordagem inovadora para o planejamento de trajetórias seguras e eficientes em ambientes parcialmente desconhecidos, com características estáticas e dinâmicas. A proposta integra o algoritmo Whale Optimization Algorithm (WOA) à estrutura MHRTSN (MetaHeuristic Real-Time Safe Navigation), que utiliza dados sensoriais em tempo real para navegar de forma reativa sem a necessidade de mapas prévios. A MHRTSN-WOA avalia pontos candidatos no espaço de atuação do robô com base em uma função objetivo composta por termos de atração, repulsão e penalidades. Cada indivíduo na população representa um único ponto (x, y) , permitindo que o tamanho da população se mantenha constante independentemente da complexidade do ambiente. A técnica foi implementada em uma plataforma integrada com o simulador Webots e o software MATLAB, e avaliada em dois ambientes de simulação: um estático e outro dinâmico, com obstáculos móveis. Os resultados experimentais demonstram reduções expressivas no tempo de processamento por deslocamento — de até 75,89% em relação à versão com algoritmo genético (GA) e de até 66,39% em relação à versão com PSO — sem comprometer a segurança ou a qualidade da trajetória. Adicionalmente, a MHRTSN-WOA mostrou-se mais simples de ajustar, exigindo apenas um parâmetro principal. Esses resultados indicam que a estratégia proposta é eficaz, escalável e adequada para aplicações em navegação robótica em cenários complexos e variáveis.

Index Terms—Navegação Robótica, Whale Optimization Algorithm, Planejamento de Trajetória, Ambientes Dinâmicos, Meta-Heurísticas

I. INTRODUÇÃO

Com o avanço da tecnologia, sistemas robóticos têm sido amplamente empregados em diversos cenários, desempenhando uma variedade de funções e tornando-se fundamentais para a execução de tarefas complexas. O desenvolvimento dessas aplicações está fortemente relacionado ao aumento da produtividade e à redução de riscos operacionais. A praticidade observada no uso desses sistemas deve-se, em grande parte, à sua capacidade de operar de forma autônoma, percebendo o ambiente ao redor e tomando decisões inteligentes para alcançar seus objetivos [1], [2].

A autonomia de robôs móveis é sustentada por diversas etapas interdependentes, como aquisição de dados por sensores, modelagem do ambiente, planejamento da trajetória e controle de movimento. Esses elementos atuam de forma integrada, permitindo ao robô construir um mapa do ambiente e traçar uma rota viável entre dois pontos, otimizando métricas como distância percorrida, consumo energético, tempo de execução e segurança. Nesse contexto, o planejamento de trajetórias destaca-se como uma etapa crítica, pois viabiliza a navegação segura e eficiente em ambientes com obstáculos e variações espaciais. A trajetória gerada deve ser seguida com precisão, a fim de evitar colisões e garantir a integridade da operação [3], [4]. Contudo, esse processo enfrenta desafios significativos, incluindo limitações na modelagem do ambiente, restrições computacionais e a frequência de amostragem dos sensores. Por tratar-se de um problema classificado como NP-difícil, o planejamento de trajetórias demanda algoritmos robustos e alto poder computacional para lidar com essas restrições [5], [6].

Diversas abordagens baseadas em inteligência artificial têm sido propostas na literatura para lidar com esses desafios [7]–[19]. Algumas dessas estratégias combinam diferentes técnicas, originando soluções híbridas que visam melhorar o desempenho da navegação. Por exemplo, os trabalhos em [7], [8] empregam algoritmos de aprendizado de máquina para navegação em ambientes com obstáculos. Em [7], propõe-se uma abordagem híbrida com aprendizado por reforço profundo e camadas LSTM, voltada para terrenos irregulares, dispensando o uso de mapas prévios. Já em [8], redes neurais são combinadas com aprendizado por reforço hierárquico, utilizando sensores para gerar mapas em grade e uma função de recompensa que favorece caminhos suaves e livres de colisões. No entanto, essas soluções requerem elevado poder computacional, o que limita sua aplicação em plataformas com recursos restritos.

Outros estudos adotam sistemas baseados em lógica fuzzy, especialmente em contextos dinâmicos e desconhecidos [9], [10]. Essas abordagens processam em tempo real os dados

sensoriais para identificar obstáculos e definir a trajetória. Em [9], por exemplo, é apresentado um sistema neuro-fuzzy adaptativo para robôs com limitações de hardware, empregando sensores ultrassônicos e ponderações baseadas na distância até o destino. A proposta de [10], por sua vez, combina campos potenciais com lógica fuzzy para evitar aprisionamento em mínimos locais. Entretanto, essas soluções podem ter seu desempenho comprometido caso as regras fuzzy não estejam adequadamente calibradas ao ambiente.

Uma categoria significativa de soluções utiliza algoritmos meta-heurísticos para o planejamento em ambientes com obstáculos [11]–[19]. As propostas em [12], [13], por exemplo, operam em ambientes estáticos com conhecimento prévio. Em [12], o algoritmo *Sparrow Search* é aprimorado com aprendizado por oposição para gerar uma população inicial mais eficaz. Já [13] apresenta uma combinação entre *Particle Swarm Optimization* (PSO) e campos potenciais artificiais, otimizando a movimentação do robô. Em cenários dinâmicos, outras soluções realizam ajustes locais conforme mudanças no ambiente. Em [14], a trajetória global é calculada com *Ant Colony Optimization* (ACO) e refinada com o método *Dynamic Window Approach*. Em [16], uma versão modificada do *Simulated Annealing* introduz operações de seleção e exclusão de caminhos, reduzindo o custo computacional. Ambas as abordagens requerem conhecimento prévio do ambiente discretizado.

Por outro lado, as estratégias descritas em [11], [17]–[19] adotam navegação reativa, dispensando mapas globais. Em [11], é proposto o algoritmo híbrido *Hybrid Particle Swarm and Chemical Reaction Optimization* (HPCRO), que combina PSO com *Chemical Reaction Optimization* (CRO), gerando trajetórias contínuas e seguras em ambientes desconhecidos. O trabalho em [17] integra campos potenciais com algoritmos evolucionários bacterianos para escapar de mínimos locais. Já a solução *Dynamic Planning Navigation Algorithm optimized with Genetic Algorithm* (DPNA-GA) [18], [20] realiza o replanejamento em tempo real com base em dados sensoriais atualizados, utilizando algoritmos genéticos (GA) para selecionar o próximo ponto de deslocamento. A técnica *MetaHeuristic Real-Time Safe Navigation* (MHRTSN), apresentada em [19], introduz uma abordagem segura para navegação em tempo real, mantendo distância segura de obstáculos por meio da construção de um polígono de segurança e restrições mecânicas do robô. A solução emprega campos potenciais e avaliação por meta-heurísticas, como GA e PSO, demonstrando resultados superiores em ambientes complexos.

Diante desse contexto, a estratégia *MetaHeuristic Real-Time Safe Navigation integrated with Whale Optimization Algorithm* (MHRTSN-WOA) proposta neste artigo destaca-se por oferecer uma solução eficiente e segura para o planejamento de trajetórias em ambientes parcialmente desconhecidos, com características estáticas e dinâmicas. A proposta integra o algoritmo *Whale Optimization Algorithm* (WOA) à estrutura MHRTSN, mantendo os mecanismos que asseguram a distância segura do robô em relação aos obstáculos e superando limitações de técnicas que desconsideram zonas de risco. A técnica combina

a flexibilidade da navegação baseada em campos potenciais com a capacidade exploratória do WOA. Sua estrutura permite que cada indivíduo represente apenas um ponto no espaço (x, y) , mantendo o tamanho da população constante independentemente da complexidade do ambiente. Além disso, o WOA exige apenas um único parâmetro de ajuste (SC), o que simplifica sua aplicação prática. Os resultados obtidos indicam que a MHRTSN-WOA reduz significativamente o tempo de processamento por deslocamento em relação às variantes com GA e PSO, mantendo trajetórias seguras e de boa qualidade.

II. FUNDAMENTAÇÃO: ESTRATÉGIA MHRTSN

Nesta seção, será realizada uma descrição da estratégia de navegação MHRTSN (*MetaHeuristic Real-Time Safe Navigation*), a fim de fornecer uma melhor compreensão da proposta deste artigo. O MHRTSN foi inicialmente proposto em [19]. Essa estratégia visa guiar robôs móveis autônomos de maneira segura, por meio da minimização de uma função objetivo composta por termos de atração, repulsão e penalidades. Tal função objetivo considera a geração de um polígono em torno do robô garantindo uma distância segura dos obstáculos, bem como considera as restrições de cinemática do robô. Diferentemente de abordagens tradicionais, a MHRTSN não requer conhecimento prévio do ambiente, coletando os dados sensoriais dinamicamente a cada deslocamento. Sua operação ocorre de maneira iterativa, invocando uma meta heurística baseada em população a cada deslocamento para selecionar o melhor candidato a solução local dentro de um raio de atuação. De modo que, essa solução local pode ou não coincidir com a solução global.

O processo de navegação é dividido em quatro subetapas principais: coleta de dados, pré-processamento, avaliação meta-heurística e deslocamento (ver Figura 1). Em cada uma das N iterações (deslocamentos), o ambiente ao redor do robô é amostrado por P pontos uniformemente distribuídos. A partir disso, é gerado um polígono que representa o espaço livre navegável, a partir do qual os candidatos a ponto objetivo são avaliados por uma função de *fitness*.

Os tópicos abaixo descrevem resumidamente cada subetapa:

Coleta de dados: O robô realiza a leitura do ambiente ao seu redor com P amostras igualmente espaçadas. Essas leituras são convertidas em coordenadas cartesianas, formando um polígono que representa as regiões ocupadas por obstáculos e áreas livres.

Pré-processamento: A partir dos pontos coletados, são identificadas as regiões ocupadas por obstáculos, e a região navegável é construída. Essa região é representada por um polígono livre de colisões, adequado à geometria do robô.

Avaliação meta-heurística: Um conjunto de candidatos a ponto objetivo é gerado dentro de um raio de atuação R^{mh} ao redor do robô. Cada candidato é avaliado por

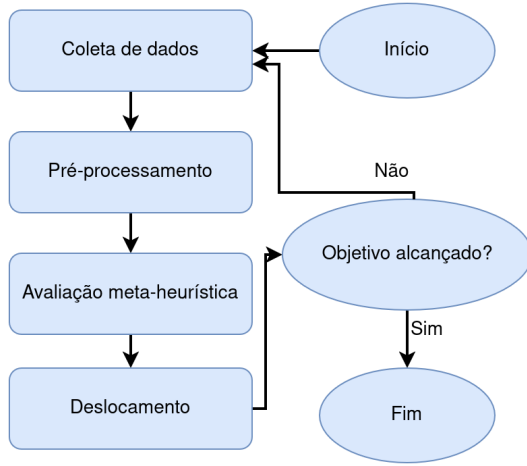


Figura 1: Diagrama de fluxo das subetapas associadas à estratégia MHRTSN.

meio da seguinte função de aptidão,

$$f_n(k, g) = F_{ed}^G(k, g) + \rho_n \frac{1}{F_{ed}^{op}(n, p)} + \rho_n A(n, k, g) + B(n, k, g), \quad (1)$$

onde F_{ed}^G representa a distância até o objetivo global, $\frac{1}{F_{ed}^{op}}$ atua como força repulsiva em relação aos obstáculos, $A(n, k, g)$ penaliza regiões já visitadas e $B(n, k, g)$ penaliza pontos inviáveis ou inacessíveis. O indivíduo com melhor valor de f é selecionado como o ponto objetivo local sp_n^{LG} .

Deslocamento: O robô se desloca até o ponto selecionado sp_n^{LG} e atualiza sua posição. Caso o objetivo global seja atingido, o algoritmo é encerrado; caso contrário, o processo é reiniciado com uma nova coleta de dados.

No presente trabalho, esses termos são apresentados de forma resumida para contextualizar o processo de avaliação. A formulação matemática detalhada, incluindo a dedução completa e a definição de todos os parâmetros envolvidos, encontra-se descrita em [19].

A etapa de avaliação meta-heurística é responsável por explorar de forma eficiente o espaço de configuração local a cada iteração, promovendo a escolha de pontos viáveis mesmo em ambientes complexos com descontinuidades e regiões de difícil acesso. A função objetivo avaliada apresenta características típicas de funções de campos potenciais com múltiplos mínimos locais, tornando o uso de meta-heurísticas uma escolha adequada. Além disso, a complexidade total pode ser expressa como $O(N_{Max} \cdot G \cdot K \cdot (P + N))$. Considerando que N_{Max} é um valor constante definido para limitar o número máximo de deslocamentos, a complexidade assintótica do algoritmo é $O(G \cdot K \cdot (P + N))$. Esse resultado reflete o impacto combinado dos principais parâmetros do algoritmo, com a etapa de avaliação sendo o fator predominante no comportamento computacional. Desta forma, a escolha de uma boa meta-heurística em conjunto com uma boa sintonia dos

parâmetros, influencia diretamente na busca e custo computacional na solução do problema.

III. MHRTSN-WOA

A proposta MHRTSN-WOA representa uma evolução significativa no planejamento de trajetórias reativas, ao incorporar a meta-heurística Whale Optimization Algorithm (WOA) na etapa de avaliação de pontos de navegação. Diferentemente de abordagens convencionais, em que o ponto objetivo é escolhido com base em regras fixas ou em estratégias locais simples, a MHRTSN-WOA utiliza o poder da inteligência de enxame para explorar, selecionar e refinar pontos de destino mais promissores dentro de um polígono de navegação gerado dinamicamente pelos sensores do robô.

O algoritmo WOA se inspira no comportamento coletivo das baleias jubarte durante a técnica de bolha espiral para caça. Essa inspiração permite modelar uma busca com exploração e intensificação balanceadas, alternando entre movimentos espirais em torno da melhor solução encontrada (exploração local) e saltos aleatórios guiados pela posição do grupo (exploração global). Na MHRTSN-WOA, cada indivíduo do enxame representa um ponto candidato (x, y) a ser avaliado como novo destino do robô.

A principal inovação reside na adaptação do WOA para operar de forma eficiente dentro dos limites operacionais e sensoriais impostos pelo ambiente. O polígono de navegação define dinamicamente a área de busca, e a função de avaliação utilizada considera fatores como a distância euclidiana ao objetivo global, o afastamento de obstáculos, e penalizações. Essa função mensura a qualidade de um determinado ponto do ambiente. Entretanto, essa função possui diversos pontos com mínimos locais e descontinuidades. Dessa forma, o WOA entra para selecionar o melhor ponto objetivo local com base nessa função de avaliação, obtendo resultados concisos e eficientes graças à simplicidade do algoritmo e sua capacidade de modelar uma inteligência de enxame.

Além disso, os parâmetros do WOA foram ajustados para maximizar seu desempenho em ambientes de alta variabilidade. O fator de redução a , que controla a transição entre as fases de exploração e intensificação, foi calibrado para garantir maior estabilidade ao longo das gerações, evitando convergência prematura. A constante espiral b e o coeficiente aleatório r também foram modificados para permitir um comportamento adaptativo, sensível à densidade de obstáculos e ao formato do polígono de navegação.

A escolha da meta-heurística WOA foi motivada por sua simplicidade de implementação, baixo custo computacional e boa capacidade de escapar de ótimos locais, característica essencial em ambientes parcialmente observáveis. A natureza colaborativa do algoritmo permite que diferentes regiões do espaço de busca sejam exploradas simultaneamente, aproveitando tanto a memória coletiva do enxame quanto a diversidade das soluções individuais. Isso resulta em uma navegação mais inteligente, resiliente e adaptada ao ambiente, superando as limitações de técnicas determinísticas ou exclusivamente baseadas em gradientes.

O processo geral da MHRTSN-WOA pode ser resumido no Algoritmo 1, que apresenta de forma simplificada as principais etapas: coleta e processamento dos dados sensoriais, avaliação meta-heurística via WOA e deslocamento do robô até o ponto objetivo local.

Algorithm 1 MHRTSN-WOA

- 1: **while** objetivo global não alcançado e limite de deslocamentos não atingido **do**
- 2: **Coleta de dados:** ler sensores e obter pontos do ambiente
- 3: **Pré-processamento:** identificar obstáculos e espaço livre navegável
- 4: **Avaliação por WOA:**
Inicializar população de baleias (candidatos)
- 5: **while** critério de parada do WOA não atingido **do**
 Avaliar candidatos
 Atualizar baleia líder
 Atualizar baleias seguindo líder ou em espiral
- 6: **end while**
 Selecionar melhor candidato como ponto objetivo local
- 7: **Deslocamento:** mover robô até o ponto objetivo local
- 8: **end while**

A Figura 2 ilustra o fluxograma da geração e seleção de um ponto objetivo por parte do WOA na estratégia MHRTSN-WOA, enquanto o Algoritmo 1 fornece uma visão resumida em pseudocódigo das etapas executadas.

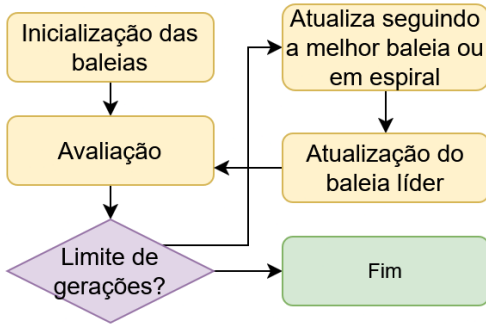


Figura 2: Ilustração das etapas da MHRTSN-WOA: geração dos candidatos, avaliação via WOA e definição do ponto objetivo no ciclo de sub-deslocamento.

IV. PLATAFORMA DE SIMULAÇÃO

Os experimentos realizados neste trabalho foram conduzidos em uma plataforma de simulação desenvolvida com base na integração entre o simulador 3D Webots e o software Matlab (licença 1080073) [21]. Essa personalização permite uma interação eficiente entre as plataformas e possibilita o rastreamento dos movimentos de navegação do robô. De modo que, a plataforma de simulação customizada está vinculada a estratégia de navegação MHRTSN conforme ilustrado na Figura 3.

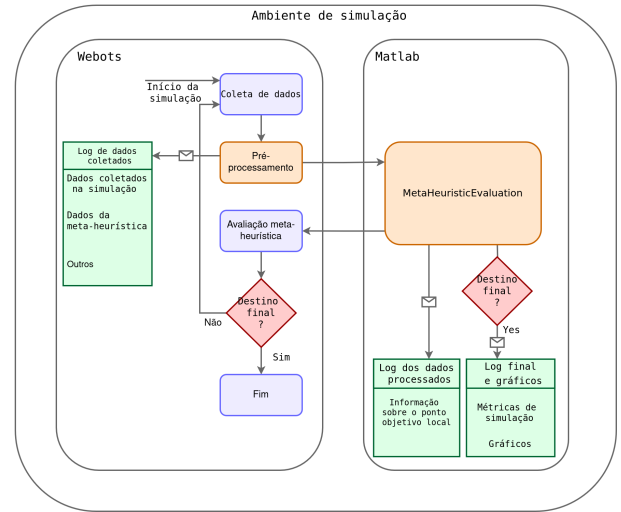


Figura 3: Fluxo de comunicação entre os módulos da plataforma de simulação.

A arquitetura da plataforma utilizada é composta por módulos que se comunicam entre si, conforme o fluxo representado na Figura 3. Os módulos indicados em laranja são responsáveis pelas funções relacionadas ao robô, como o controle de movimento e a coleta de informações sensoriais, e são implementados em linguagem C. Já o bloco destacado em azul refere-se ao processamento das estratégias meta-heurísticas, o qual é executado no Matlab, onde ocorre a escolha do ponto objetivo a partir dos dados coletados. Todo conteúdo visto na Figura 3 é denominado de ambiente de simulação.

Outro trecho da plataforma de simulação, denominado de programa orquestrador, é responsável por gerenciar todo o ciclo de simulação (ver Figura 4). Esse programa realiza a leitura de um arquivo de configuração no formato JSON, contendo informações como o ambiente de teste, o tipo de técnica de navegação a ser aplicada, os parâmetros dos algoritmos, o número de execuções e outras definições. Com base nesses dados, o orquestrador inicia, por meio de comandos do sistema operacional, a simulação no Webots e a execução do processamento no Matlab, assegurando a sincronização entre as plataformas e a coleta sistemática dos resultados.

Durante cada iteração da simulação, são registrados arquivos de log contendo dados como a posição atual do robô, os obstáculos detectados e os resultados da técnica de navegação. Esses registros são utilizados tanto para análise de desempenho quanto para reprodutibilidade dos testes. Para medir o tempo de execução das técnicas, foram empregadas duas funções nativas do Matlab: ‘cputime’, que fornece o tempo total de uso da CPU, e ‘tic/toc’, que permite medir o tempo decorrido entre dois pontos específicos no código. Essas medições são utilizadas para avaliar o desempenho computacional da estratégia MHRTSN. Todas as simulações foram conduzidas em um computador com sistema de 64 bits, equipado com

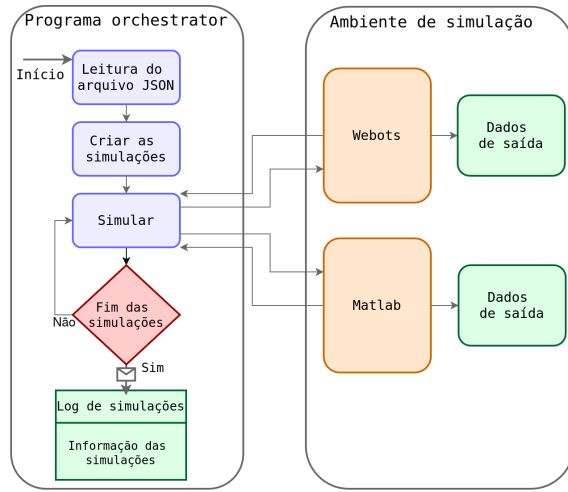


Figura 4: Ambiente tridimensional utilizado durante os testes de navegação.

um processador de 6 núcleos e 12 *threads*, com frequência de 2.30GHz e memória RAM de 20GB.

Nas simulações, foi utilizado um robô móvel não-holonômico de corpo circular, com dimensões de raio de 0,15 m e altura de 0,2 m, equipado com acionamento diferencial. Neste robô, são dispostos igualmente 4 sensores de distância. Os parâmetros comuns utilizados nas simulações, como o número e o alcance dos sensores, o espaçamento angular, o raio de geração da população e os fatores de penalidade e tolerância, estão apresentados na Tabela I. Esses parâmetros determinam a granularidade da percepção e a área de busca local durante cada ciclo de otimização e deslocamento.

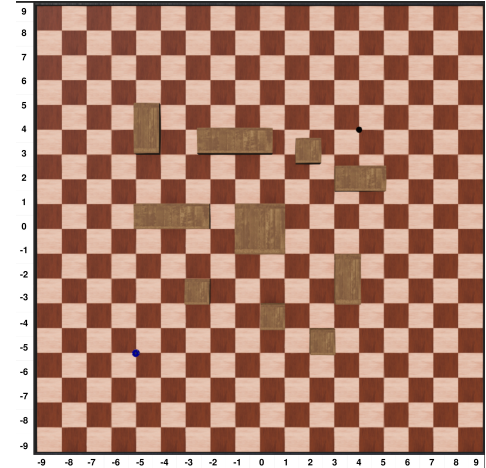
Tabela I: Parâmetros gerais do MHRTSN utilizados nas simulações com a estratégia MHRTSN-WOA.

Parâmetro	Valor
Número de sensores (s)	4
Alcance máximo dos sensores (Φ)	3 m
Deslocamento angular entre sensores (θ)	6°
Raio de geração dos indivíduos (R^{mh})	1 m
Fator de penalidade (τ)	1000
Fator de tolerância (ϵ)	0,2
Fator de ajuste (α)	1,4
Codificação dos indivíduos	Valor real

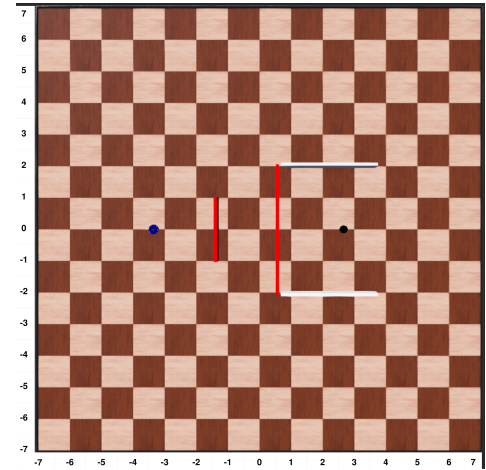
V. RESULTADOS

Com o objetivo de avaliar a capacidade e o desempenho da técnica proposta, MHRTSN-WOA, foram construídos dois ambientes de simulação. O primeiro ambiente, denominado AMB-1, corresponde a um cenário externo densamente povoado por obstáculos. Os pontos inicial e final desse ambiente são definidos por $(-5, 0; -5, 0)$ e $(4, 0; 4, 0)$, respectivamente (ver Figura 5b). O segundo ambiente, denominado AMB-2, também é externo, mas com a adição de duas paredes em momentos específicos, a fim de testar a capacidade da técnica de adaptação a mudanças dinâmicas. Os pontos inicial e final

são definidos por $(-4, 0; 0, 0)$ e $(2, 0; 0, 0)$, respectivamente. As duas paredes são representadas em vermelho na Figura 5b e suas características estão detalhadas na Tabela II. Ambos os ambientes são baseados nos propostos por [19], sendo os resultados obtidos nesta seção comparados com os descritos em [19], conforme a Seção VI. O vídeo demonstrativo do deslocamento do robô utilizando a técnica proposta pode ser acessado em https://drive.google.com/drive/folders/1oFeoXBVSggwiOcjdc82Hiim0_C_FfBHP.



(a) AMB-1.



(b) AMB-2.

Figura 5: Representação dos ambientes utilizados nas simulações do MHRTSN-WOA. O robô é indicado por um círculo azul na posição inicial, enquanto o ponto final de navegação é representado por um círculo preto. Nesses ambientes, o piso é marcado com quadrados de 1 por 1 metro em padrão quadriculado. Na Figura (a), os obstáculos estáticos são representados por caixas marrons. Na Figura (b), os obstáculos estáticos são representados por caixas brancas, e os obstáculos dinâmicos são mostrados como paredes vermelhas.

A Tabela III apresenta os parâmetros utilizados nas simulações com a estratégia MHRTSN-WOA. Os testes foram realizados nos ambientes AMB-1 e AMB-2, variando o

Tabela II: Dimensão, localização e tempo de adição das paredes no AMB-2.

Parede	Posição (x, y)	Dimensão (x, y, z)	Tempo de adição (s)
Parede 1	(-2.0, 0.0)	(0.1, 2, 1)	25
Parede 2	(-0.1, 0.0)	(0.1, 4, 1)	180

tamanho da população, o número máximo de gerações e o coeficiente de espiral (SC), que influencia diretamente o comportamento exploratório do algoritmo. Essas configurações visam analisar o impacto de tais parâmetros na eficiência da trajetória gerada.

Tabela III: Configurações dos parâmetros utilizados nas simulações da estratégia MHRTSN-WOA.

Simulação	Ambiente	População	Max. Ger.	SC
S1	AMB-1	20	30	0,9
S2	AMB-1	20	30	1,4
S3	AMB-1	30	10	0,9
S4	AMB-1	30	10	1,4
S5	AMB-2	20	30	0,9
S6	AMB-2	20	30	1,4
S7	AMB-2	30	10	0,9
S8	AMB-2	30	10	1,4

Em cada simulação foi gerado um conjunto de dados contendo os pontos detectados, distância percorrida, tempo de processamento, entre outras métricas. Para avaliação da estratégia MHRTSN-WOA, foram utilizados os seguintes indicadores: tempo de CPU, tempo de relógio, número de deslocamentos e distância percorrida. A Tabela IV apresenta os valores médios (μ) e respectivos desvios padrão (σ) calculados a partir de 60 amostras. Esse número de simulações foi obtido por meio do cálculo do tamanho mínimo de amostra necessário, considerando nível de confiança, margem de erro e variância preliminar dos dados.

Ao analisar os resultados da Tabela IV, observa-se que, para um mesmo ambiente, a variação do parâmetro SC teve pouca influência no tempo, na distância e no número de deslocamentos. Por outro lado, os parâmetros relacionados ao número de gerações e ao tamanho da população impactaram significativamente o tempo de processamento, sem, contudo, trazer melhorias no trajeto percorrido. Esse comportamento foi observado em ambos os ambientes.

Adicionalmente, foram definidos dois indicadores para ampliar a análise: o indicador I_1 (Figura 6), que representa o tempo médio de processamento por evento de deslocamento, e o indicador I_2 (Figura 7), que corresponde à distância média percorrida em cada deslocamento. Essas métricas visam proporcionar maior robustez frente a variações no percurso ou na quantidade total de deslocamentos.

Nas Figuras 6 e 7, as simulações no ambiente AMB-1 são representadas na cor azul, enquanto as do AMB-2 aparecem em laranja. As barras indicam os valores médios, sendo que as linhas verticais representam os desvios padrão. A Figura 6 mostra redução no tempo médio de processamento para as configurações S3 e S4, em comparação com S1 e S2. Não foram observadas diferenças significativas entre simulações

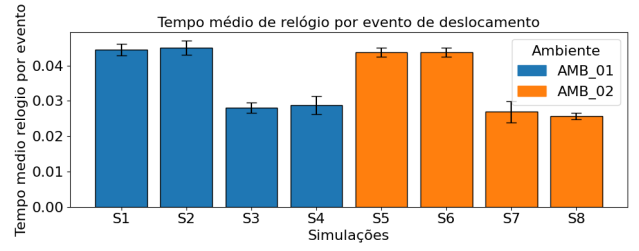


Figura 6: Média e desvio padrão do tempo de processamento por evento de deslocamento (indicador I_1) nos ambientes AMB-1 e AMB-2.

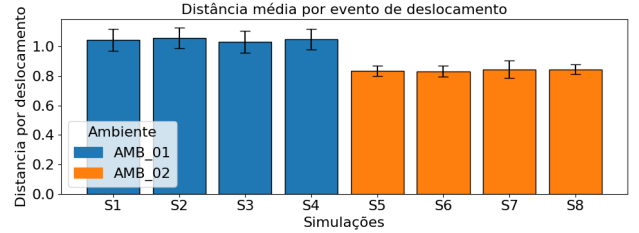


Figura 7: Média e desvio padrão da distância percorrida por evento de deslocamento (indicador I_2) nos ambientes AMB-1 e AMB-2.

que alteram apenas o parâmetro SC, como entre S1 e S2, ou S5 e S6. A Figura 7 reforça que, para um mesmo ambiente, a variação dos parâmetros analisados não influenciou a distância média por deslocamento. Tais observações, corroboradas pelos indicadores I_1 , I_2 e pela Tabela IV, indicam que a escolha por uma população ligeiramente maior com menos gerações não compromete a qualidade da solução, ao mesmo tempo em que reduz significativamente o tempo necessário para o processamento. Esse padrão foi consistente tanto em ambientes estáticos quanto dinâmicos.

VI. COMPARAÇÃO COM OUTRAS TÉCNICAS

A técnica MHRTSN-WOA também foi avaliada em comparação com outras abordagens da literatura. Os ambientes de teste AMB-1 e AMB-2 foram os mesmos utilizados no estudo apresentado em [19], no qual a técnica MHRTSN foi aplicada nas variantes MHRTSN-GA e MHRTSN-PSO. Dessa forma, foi possível realizar uma comparação direta entre o desempenho das três abordagens. As Figuras 8 e 9 apresentam os resultados comparativos com base nos indicadores I_1 e I_2 , respectivamente.

A Figura 8 evidencia uma redução significativa no tempo de processamento por evento de deslocamento com o uso da técnica baseada em WOA. No ambiente AMB-1, a redução média foi de aproximadamente 66,39% em relação ao PSO e de 75,89% em relação ao GA. Já no ambiente AMB-2, as reduções observadas foram de 62,00% e 72,62%, respectivamente.

A Figura 9 apresenta os valores médios de distância percorrida por evento de deslocamento (indicador I_2) para os

Tabela IV: Resultados das simulações para os ambientes AMB-1 e AMB-2: Tempos de CPU, tempos de relógio, deslocamentos e distâncias.

Simulação	Tempos de CPU (s)		Tempos de relógio (s)		Deslocamentos		Distância (m)	
	μ	σ	μ	σ	μ	σ	μ	σ
S1	1,2723	0,1551	0,7576	0,1168	17,1000	2,9816	17,7223	2,2153
S2	1,2827	0,1283	0,7594	0,0963	16,9167	2,6476	17,7880	2,0063
S3	0,9102	0,0974	0,4829	0,0671	17,3333	2,9078	17,7498	2,2741
S4	0,8872	0,0958	0,4872	0,0618	17,0333	2,6202	17,7623	2,0609
S5	1,5388	0,1079	0,9485	0,0930	21,7167	2,3459	18,1255	1,7827
S6	1,5409	0,1424	0,9729	0,1212	22,2807	3,0190	18,5083	2,3406
S7	1,0258	0,0876	0,5631	0,0605	21,1500	2,8097	17,7540	2,2145
S8	1,0125	0,0684	0,5671	0,0506	22,0962	2,2469	18,6519	1,6870

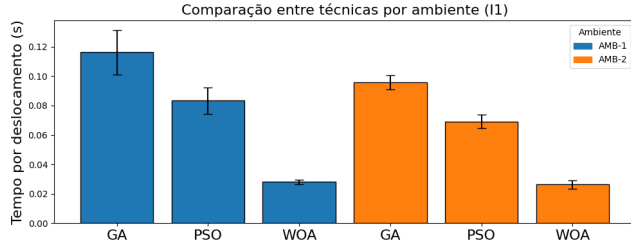


Figura 8: Comparativo do indicador I_1 entre as variantes MHRTSN-WOA, MHRTSN-PSO e MHRTSN-GA.

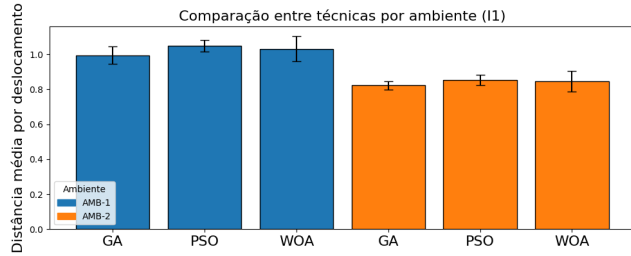


Figura 9: Comparativo do indicador I_2 entre as variantes MHRTSN-WOA, MHRTSN-PSO e MHRTSN-GA.

ambientes AMB-1 e AMB-2. Observa-se que, no ambiente AMB-1, todas as técnicas apresentaram desempenhos semelhantes, com médias próximas a uma unidade de distância. Já no ambiente AMB-2, a distância média por deslocamento foi consideravelmente menor para todas as variantes, o que está relacionado à configuração espacial do ambiente. A técnica MHRTSN-WOA manteve desempenho comparável ao das demais abordagens, indicando que sua maior eficiência no tempo de processamento (indicador I_1) não comprometeu a qualidade do trajeto gerado.

Para verificar a existência de diferenças estatisticamente significativas nos indicadores, foram aplicados testes estatísticos. O teste ANOVA foi utilizado quando os dados apresentaram distribuição normal, enquanto o teste de Kruskal-Wallis (KW) foi adotado nos casos em que a distribuição não foi normal. As comparações foram organizadas em grupos: o grupo G_1 reúne as simulações S3 e S4 (deste artigo) e as simulações S13, S14, S29 e S30 do trabalho [19]; o grupo G_2 inclui as simulações S7 e S8 (deste artigo) e S35 e S36 do trabalho [19]. A Tabela V apresenta os resultados dos testes estatísticos. Em

todos os casos, a hipótese nula (H_0) foi rejeitada, uma vez que os valores de p ficaram abaixo do nível de significância estabelecido ($\alpha = 0,05$), indicando a presença de diferenças estatisticamente significativas entre as abordagens comparadas.

Tabela V: Resultados dos testes estatísticos para os indicadores I_1 e I_2 .

Grupo	Indicador	Teste aplicado	p -valor
G_1	I_1	KW	$3,0825 \times 10^{-66}$
G_1	I_2	KW	$6,0056 \times 10^{-4}$
G_2	I_1	KW	$4,2859 \times 10^{-43}$
G_2	I_2	ANOVA	$1,4246 \times 10^{-4}$

A Figura 10 apresenta um comparativo entre o caminho percorrido pela técnica MHRTSN-WOA e os percursos gerados pela proposta descrita em [13], no ambiente AMB-3. Os dados foram obtidos a partir do mapeamento aproximado dos pontos extraídos das imagens disponibilizadas no referido trabalho. Observa-se que, diferentemente das abordagens propostas em [13], a estratégia MHRTSN-WOA manteve uma distância segura em relação aos obstáculos durante todo o trajeto. As demais abordagens não consideram as dimensões físicas do robô, o que pode levar a trajetórias excessivamente próximas aos obstáculos, comprometendo a segurança da navegação.

A Tabela VI apresenta uma comparação entre os caminhos percorridos pelas abordagens rPSO, apfrPSO e MHRTSN-WOA. Os dados demonstram que a estratégia MHRTSN-WOA gerou trajetórias com desempenho comparável às abordagens baseadas em conhecimento prévio do ambiente, mas com a vantagem de operar de forma totalmente online, sem qualquer informação antecipada sobre o cenário. Além disso, a MHRTSN-WOA mantém constante a estrutura dos indivíduos, independentemente das dimensões do ambiente. Essa característica reduz a complexidade computacional à medida que o ambiente cresce, evitando os problemas de escalabilidade enfrentados por outras técnicas.

Tabela VI: Comparação entre as técnicas rPSO, apfrPSO e MHRTSN-WOA no ambiente AMB-3.

Técnica	Trajeto (m)	Mapa conhecido	Segurança	Indivíduo fixo
rPSO	27,21	Sim	Não	Não
apfrPSO	24,38	Sim	Não	Não
MHRTSN-WOA	24,12	Não	Sim	Sim

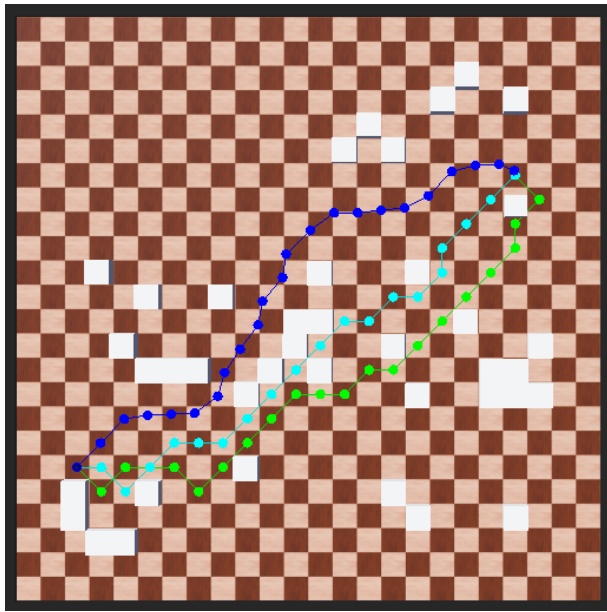


Figura 10: Comparação de trajetórias no ambiente AMB-3, proposto em [13]. As cores representam as técnicas: rPSO (verde), apfrPSO (ciano) e MHRTSN-WOA (azul). Obstáculos são indicados por regiões brancas. O ponto de partida está à esquerda e o destino no canto superior direito.

VII. CONCLUSÃO

Os resultados demonstraram que a integração do algoritmo WOA à estratégia MHRTSN resultou em melhorias significativas no desempenho computacional. A técnica MHRTSN-WOA apresentou reduções expressivas no tempo de processamento por evento de deslocamento, superando as variantes MHRTSN-GA e MHRTSN-PSO, com destaque para reduções superiores a 70% em alguns cenários.

Além da eficiência, a MHRTSN-WOA se destaca pela simplicidade de ajuste, exigindo apenas o coeficiente de espiral (SC) como parâmetro. Essa característica facilita sua aplicação prática e reduz o custo de experimentação. A técnica também manteve a qualidade das trajetórias geradas, sem comprometer a segurança ou a distância percorrida. Dessa forma, a MHRTSN-WOA mostrou-se uma abordagem eficaz, simples e escalável para navegação de robôs móveis em ambientes complexos.

REFERÊNCIAS

- [1] S. Lin, A. Liu, J. Wang, and X. Kong, "A review of path-planning approaches for multiple mobile robots," *Machines*, vol. 10, no. 9, 2022.
- [2] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao, "Review of autonomous path planning algorithms for mobile robots," *Drones*, vol. 7, no. 3, 2023.
- [3] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," *Procedia Computer Science*, vol. 127, pp. 180–189, 2018.
- [4] Q. He, Z. Wang, K. Li, Y. Zhang, and M. Li, "Research on autonomous navigation of mobile robots based on ia-dwa algorithm," *Scientific Reports*, vol. 15, no. 1, p. 2099, 2025.
- [5] Z. Zhu, F. Wang, S. He, and Y. Sun, "Global path planning of mobile robots using a memetic algorithm," *International Journal of Systems Science*, vol. 46, no. 11, pp. 1982–1993, 2015.

- [6] Y. Xu, Q. Li, X. Xu, J. Yang, and Y. Chen, "Research progress of nature-inspired metaheuristic algorithms in mobile robot path planning," *Electronics*, vol. 12, no. 15, 2023.
- [7] S. Josef and A. Degani, "Deep reinforcement learning for safe local planning of a ground vehicle in unknown rough terrain," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6748–6755, 2020.
- [8] J. Yu, Y. Su, and Y. Liao, "The path planning of mobile robot by neural networks and hierarchical reinforcement learning," *Frontiers in Neurobotics*, vol. 14, p. 63, 2020.
- [9] M. Samadi Gharajeh and H. B. Jond, "An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system," *Ain Shams Engineering Journal*, vol. 13, no. 1, p. 101491, 2022.
- [10] T. A. Teli and M. A. Wani, "A fuzzy based local minima avoidance path planning in autonomous robots," *International Journal of Information Technology*, vol. 13, no. 1, pp. 33–40, 2021.
- [11] Pattnaik, S. K., Mishra, D., and Panda, S., "A comparative study of meta-heuristics for local path planning of a mobile robot," *Engineering Optimization*, vol. 54, no. 1, pp. 134–152, 2022.
- [12] Z. Yan, W. Luo, L. Yu, D. Zhang, B. Wang, and Q. Li, "A mobile robot path planning based on modified sparrow search algorithm," in *2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2022, pp. 1537–1541.
- [13] L. Zheng, W. Yu, G. Li, G. Qin, and Y. Luo, "Particle swarm algorithm path-planning method for mobile robots based on artificial potential fields," *Sensors*, vol. 23, no. 13, 2023.
- [14] Q. Wang, J. Li, L. Yang, Z. Yang, P. Li, and G. Xia, "Distributed multi-mobile robot path planning and obstacle avoidance based on aco-dwa in unknown complex terrain," *Electronics*, vol. 11, no. 14, 2022.
- [15] J. E. Cardoza Plata, M. Olgún Carbajal, J. C. Herrera Lozada, J. Sandoval Gutierrez, I. Rivera Zarate, and J. F. Serrano Talamantes, "Simulation and implementation of a mobile robot trajectory planning solution by using a genetic micro-algorithm," *Applied Sciences*, vol. 12, no. 21, 2022.
- [16] K. Shi, Z. Wu, B. Jiang, and H. R. Karimi, "Dynamic path planning of mobile robot based on improved simulated annealing algorithm," *Journal of the Franklin Institute*, vol. 360, no. 6, pp. 4378–4398, 2023.
- [17] O. Montiel, U. Orozco-Rosas, and R. Sepúlveda, "Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles," *Expert Systems with Applications*, vol. 42, no. 12, pp. 5177–5191, 2015.
- [18] C. A. D. Silva, A. V. F. M. De Oliveira, and M. A. C. Fernandes, "Validation of a dynamic planning navigation strategy applied to mobile terrestrial robots," *Sensors*, vol. 18, no. 12, 2018.
- [19] M. Balza, M. A. Goldbarg, S. N. Silva, L. M. D. Silva, and M. A. C. Fernandes, "A real-time safe navigation proposal for mobile robots in unknown environments using meta-heuristics," *IEEE Access*, vol. 13, pp. 23 987–24 013, 2025.
- [20] A. V. F. M. de Oliveira and M. A. C. Fernandes, "Dynamic planning navigation strategy for mobile terrestrial robots," *Robotica*, vol. 34, no. 3, pp. 568–583, 2016.
- [21] T. M. Inc., "Matlab version: 9.12.0 (r2022a) license 596681," Natick, Massachusetts, United States, 2022.