# Multi-robot Path Planning based on an Enhanced Harmony Search Algorithm

Pingfeng Zhang, and Fangyan Nie, *Member, IAENG*

***Abstract*—In multi-robot systems, the complexity and dynamics of path planning significantly impact task allocation efficiency and overall operational performance. This study presents an enhanced harmony search (EHS) algorithm designed to tackle these challenges. The EHS optimizes critical metrics, such as minimizing path length and reducing runtime, while generating collision-free and efficient routes in complex, dynamic environments. By integrating strategies based on the Lévy distribution and the best harmony attributes, the EHS greatly improves exploration and exploitation over traditional harmony search algorithms. Simulations show that EHS effectively manages varied robot numbers and environmental complexities, excelling in path length, step count, computational efficiency, and obstacle avoidance. Comparative tests demonstrate that EHS outperforms not only standard harmony search (HS) algorithms but also other leading optimization methods like grey wolf optimizer (GWO), whale optimization algorithm (WOA), and arithmetic optimization algorithm (AOA). These findings highlight EHS's potential for broader optimization challenges beyond multi-robot path planning (MRPP).**

***Index Terms*—multi-robot path planning, intelligent optimization, harmony search algorithm, Lévy distribution.**

## I. Introduction

ADVANCEMENTS in robotics technology have underscored the increasing importance of multi-robot systems (MRSs), distinguished by their efficiency, flexibility, and robustness in performing complex tasks [1]. An MRS consists of multiple autonomous or semi-autonomous robots designed to collaborate on goals that are beyond the reach of single robots. At the core of these systems lies the challenge of multi-robot path planning (MRPP) [2]–[4], which involves generating collision-free trajectories for each robot from initial to final positions, while simultaneously coordinating interactions among robots.

The importance of MRPP is highlighted by its dual focus on practical applications and the inherent technical challenges. For example, in logistics warehouses, automated guided vehicles (AGVs) must navigate through intricate operational environments to ensure efficient goods transport. Effective MRPP ensures not only operational efficiency but also precise task execution [1]. In manufacturing sectors, multi-robot systems are employed to perform a variety of production tasks. By optimizing path planning, these systems improve both productivity and product quality. Beyond industrial applications, MRPP plays a crucial role in areas such as intelligent traffic management, environmental monitoring, military reconnaissance, and agricultural automation, among other fields.

Despite its advantages, MRPP presents several critical challenges [5]–[9]. One primary challenge lies in computational complexity, as increasing robot numbers render optimal solution identification computationally expensive. Furthermore, operating within dynamically changing environments introduces significant difficulties, as obstacles exhibit unpredictable movements. This necessitates the development of adaptive algorithms capable of real-time response to environmental changes.

The algorithms employed in MRPP can be systematically categorized into three groups: traditional planning methods [10], intelligent optimization methods [3]–[5], [11]–[13], and others [14]. Traditional methods, exemplified by the A* algorithm and Dijkstra's algorithm, are particularly suitable for static environments. These approaches typically formulate environmental models as graphs and systematically search for optimal paths within this framework. In contrast, intelligent optimization methods, comprising genetic algorithms, ant colony optimization, particle swarm optimization, artificial bee colony algorithms, and reinforcement learning techniques, provide adaptive solutions. Notably, these advanced methodologies excel in resolving complex, nonlinear optimization problems, particularly those involving coordination and cooperation within multi-robot systems.

The Harmony Search (HS) algorithm is a metaheuristic optimization technique inspired by the improvisation mechanisms musicians employ to create harmonious compositions. First introduced in 2001 by Geem et al., this computational framework was developed to solve complex optimization problems across diverse domains [15]–[18]. Specifically, HS mimics the iterative refinement process musicians use to achieve aesthetic harmonies, applying analogous principles to discover optimal solutions within a defined parameter space.

The HS algorithm initiates by generating a population of random solutions, termed the Harmony Memory (HM). This initialization process is analogous to compiling a diverse repertoire of potential musical harmonies. The algorithm then proceeds through iterative cycles where new candidate solutions are constructed by manipulating optimization parameters according to three operational principles: memory exploitation, pitch modulation, and stochastic exploration. During each iteration, the newly proposed solution undergoes systematic evaluation. If it exhibits superior fitness, it displaces the weakest candidate in the HM archive. This iterative process continues until a predefined termination

Manuscript received January 20, 2025; revised May 18, 2025.

This work was supported by the Key project of the special subject (Master Program) of "Research and Interpretation of the Spirit of the Third Plenary Session of the 20th CPC Central Committee" of Guizhou Universeity of Commerce (No.2024XJSDZD12) and the Guizhou Provincial Science and Technology Projects (No.KXJZ[2024]006).

Pingfeng Zhang is a Technician of the College of Marxism, Guizhou University of Commerce, Guiyang, Guizhou 550014 China (phone: 86-851-84607689, e-mail: zhpingfeng@163.com).

Fangyan Nie is a Professor of Computer and Information Engineering College, Guizhou University of Commerce, Guiyang, Guizhou 550014 China (phone: 86-851-84862403, e-mail: niefyan@vip.163.com).

criterion is met, such as completing a specified number of computational cycles.

The HS algorithm has demonstrated significant applicability across multidisciplinary domains, attributed to its dual capability in managing both continuous and discrete optimization parameters, coupled with its computational efficiency. A primary strength of this algorithm resides in its ability to maintain equilibrium between exploration—the discovery of novel solution candidates—and exploitation—the enhancement of existing promising solutions. This equilibrium is strategically achieved through the dynamic maintenance of a harmony memory archive, which facilitates the generation of new solutions via parametric adjustments. The algorithm's effectiveness stems from its dual focus: thoroughly exploring the solution space while concurrently optimizing identified promising candidates. Notably, the HS framework requires only three critical parameters: harmony memory size (HMS), harmony memory consideration rate (HMCR), and pitch adjustment rate (PAR). These parameters can be tailored to problem-specific characteristics, thereby enabling the algorithm's adaptability across diverse optimization scenarios.

While the HS algorithm has proven effective across various optimization paradigms, it manifests inherent limitations [16] in complex, dynamic scenarios such as those presented by the MRPP. To overcome these challenges, this study proposes a novel HS variant specifically engineered to address the multidimensional complexities inherent to MRPP optimization. The algorithmic modifications presented herein are designed to: (1) mitigate premature convergence through adaptive parameter regulation, (2) reduce sensitivity to initial parameter configurations via self-adaptive mechanisms, and (3) enhance exploration efficiency through dynamic memory management.

To validate the proposed enhancements, we conducted comprehensive benchmarking against not only the original HS algorithm but also three cutting-edge metaheuristic algorithms-Grey Wolf Optimizer (GWO) [19]–[21], Whale Optimization Algorithm (WOA) [22]–[24], and Arithmetic Optimization Algorithm (AOA) [25]–[27]. Through rigorous experimental evaluation, our findings demonstrate that the modified HS formulation achieves statistically significant performance improvements across all benchmark metrics when solving complex MRPP instances.

## II. Modeling of MRPP Problem

In the simulation model, the operational environment is abstracted as a rectangular plane, which offers a simplified yet comprehensive representation of the workspace. This plane facilitates the coordinated deployment of multiple robotic units for task execution. It encompasses both static obstacles, defined as entities with fixed spatial coordinates, and dynamic obstacles characterized by time-varying positional parameters.

During task execution, robotic systems must autonomously navigate from predefined starting coordinates to specified target coordinates. This navigation process necessitates circumvention of static obstacles to maintain a continuous, unimpeded trajectory. To address dynamic operational challenges, advanced collision-avoidance algorithms are implemented to mitigate potential collision risks with dynamic obstacles or other coexisting robotic units within the shared workspace.

Figure 1 presents an illustrative example of the initial environmental configuration for a MRPP problem. As visualized in Figure 1, the operational workspace is modeled as a $100 \times 100$ unit square grid. This standardized experimental configuration is rigorously maintained across all subsequent test scenarios to ensure methodological consistency, enabling quantitative comparative analysis and performance benchmarking of the MRPP algorithms under examination.

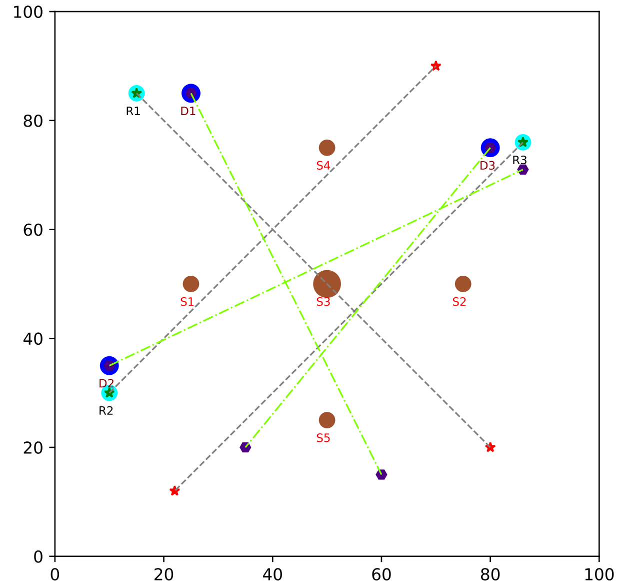



Fig. 1. Example configuration of a simulation environment for MRPP.

In this schematic representation, robotic agents are denoted by **R**, static obstacles by **S**, and dynamic obstacles by **D**, following the established color legend. Specifically, a cyan circle incorporating a pentagram symbol denotes each robot's initial position, while a red pentagram connected to this coordinate by a gray dashed line demarcates the target destination. The gray dashed lines represent computationally derived optimal trajectories for robotic navigation. Blue circles indicate the origination points of dynamic obstacles, indigo hexagons their respective target locations, and chartreuse dashed lines their projected movement vectors. Sienna-colored circular elements designate static obstacles within the operational environment.

Both robotic agents and obstacles-regardless of static or dynamic classification-are uniformly modeled as circular entities with a characteristic radius denoted as $r$. This dimensional parameter quantifies their physical extent within the operational plane.

The kinematic behavior of the robotic agent within the operational scenario visualized in Figure 1 is formally described by the differential equation presented in Equation (1).

$$\begin{cases} x_k^{t+1} = x_k^t + v_k cos(\alpha_k) \\ y_k^{t+1} = y_k^t + v_k sin(\alpha_k) \end{cases} \tag{1}$$

In Equation (1), $x_k^t$ and $y_k^t$ denote the current position coordinates of the $k$th robot, while $x_k^{t+1}$ and $y_k^{t+1}$ represent the coordinates of the next target position. The movement speed $v_k$ and movement angle $\alpha_k$ of robot $k$ are constrained within predefined intervals, i.e., $v_k \in [v_{min}, v_{max}]$ and $\alpha_k \in [\alpha_{min}, \alpha_{max}]$. In this study, we set $v_{min} = 1$, $v_{max} = 2$ and $\alpha_{min} = 0$, $\alpha_{max} = 360$.

To address the MRPP problem, planning optimal mobile paths for robots requires considering several critical factors, as outlined below:

1. Travel Distance Minimization: To enhance operational efficiency and reduce energy consumption, the total distance traversed by the robotic systems must be minimized. This objective is mathematically expressed by the function $F_1$, as defined in Equation (2).

$$F_1 = \sum_{k=1}^{nR} (d_1 + d_2) \tag{2}$$

$$\begin{aligned} d_1 &= \sqrt{(x_k^{t+1} - x_k^t)^2 + (y_k^{t+1} - y_k^t)^2} \\ d_2 &= \sqrt{(x_k^g - x_k^{t+1})^2 + (y_k^g - y_k^{t+1})^2} \end{aligned} \tag{3}$$

In the formulation, $nR$ denotes the total number of robots, while $x_k^g$ and $y_k^g$ represent the target coordinates of the $k$th robot in the 2D plane.

2. Obstacle Avoidance: To ensure safe navigation, robotic systems must dynamically adapt their paths to circumvent stationary obstacles in the environment and mobile obstacles traversing the operational zone. Specifically, this involves avoiding collisions with static obstacles while simultaneously responding to the movements of dynamic obstacles.

Avoiding static obstacles is represented by the function $F_2$, and avoiding dynamic obstacles is represented by the function $F_3$.

$$F_2 = \begin{cases} P, & \text{if}(d_k^s \leq rR) \\ 0, & \text{if}(d_k^s > rR) \end{cases} \tag{4}$$

$$d_k^s = \sum_{k=1}^{nR} \sum_{s=1}^{nS} \sqrt{(x_k^t - x_s)^2 + (y_k^t - y_s)^2} \tag{5}$$

In Equations (4) and (5), $nS$ denotes the number of static obstacles, $rR$ the robot's radius, $P$ a penalty factor (a relatively large positive real number), and $x_s$ and $y_s$ the position coordinates of the $s$th static obstacle.

$$F_3 = \begin{cases} P, & \text{if}(d_k^d \leq rR) \\ 0, & \text{if}(d_k^d > rR) \end{cases} \tag{6}$$

$$d_k^d = \sum_{k=1}^{nR} \sum_{d=1}^{nD} \sqrt{(x_k^t - x_d)^2 + (y_k^t - y_d)^2} \tag{7}$$

Here, $nD$ represents the number of dynamic obstacles, and $x_d$ and $y_d$ denote the position coordinates of the dth dynamic obstacle. The movement process of dynamic obstacles is described by Equation (8).

$$\begin{cases} x_d^{t+1} = x_d^t + v_d cos(\beta_d) \\ y_d^{t+1} = y_d^t + v_d sin(\beta_d) \end{cases} \tag{8}$$

Here, $v_d$ and $\beta_d$ represent the speed and angle of the $d$th dynamic obstacle, respectively.

3. Inter-Robot Collision Avoidance: Effective coordination mechanisms are essential to prevent collisions during both task execution and target navigation. This coordination is dynamically regulated by the collision avoidance function $F_4$, which is formally defined in Equations (9) and (10).

$$F_4 = \begin{cases} P, & \text{if}(d_k^o \leq rR) \\ 0, & \text{if}(d_k^o > rR) \end{cases} \tag{9}$$

$$d_k^o = \sum_{k=1}^{nR} \sum_{o=1}^{nR-1} \sqrt{(x_k^t - x_o^t)^2 + (y_k^t - y_o^t)^2} \tag{10}$$

Considering these factors ensures the path planning algorithm achieves two objectives: (1) identifying collision-free trajectories and (2) optimizing system-wide operational efficiency through minimized energy expenditure and travel time. The proposed solution model, which mathematically encapsulates these optimization criteria, is formally presented in Equation (11).

$$F = F_1 + F_2 + F_3 + F_4 \tag{11}$$

## III. Enhanced HS Algorithm for MRPP

In this section, we introduce the HS algorithm, detailing its algorithmic enhancements, followed by its application to the MRPP problem.

### A. Orignal HS Algorithm

The HS algorithm is a metaheuristic optimization technique inspired by the improvisational process of musicians adjusting harmonies during performances. In this framework, a solution to the optimization problem is represented as a harmony, with each decision variable corresponding to a musical note. The objective function is optimized by iteratively modifying the notes (i.e., the variables) within the harmony [15]. The HS algorithm encompasses several key steps, which are outlined below.

**Step 1:** HM Initialization. A population of candidate solutions is probabilistically generated to establish the foundational HM structure, where each solution represents a viable harmony configuration.

$$HM = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \tag{12}$$

Here, $x_{ij} = x_{min} + r \cdot (x_{max} - x_{min})$, $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, m$; $r$ is a random decimal number within the interval $(0, 1)$; $x_{min}$ and $x_{max}$ represent the lower and upper bounds of the decision variables, respectively.

**Step 2:** New Harmony Generation. Existing harmonies are selected from the HM based on specified strategies, including random selection and memory-based considerations. The selected harmonies undergo systematic perturbations governed by Equation (13), resulting in the creation of a novel harmony configuration.

$$x_{ij} = \begin{cases} x_{rj}, & R < HMCR \\ x_{min} + r \cdot (x_{max} - x_{min}), & R \geq HMCR \end{cases} \quad (13)$$

Here, $x_{rj}$ denotes a value extracted from the $j$th dimension of a randomly selected harmony in the HM. Let $R$ and $r$ represent two independent random decimals uniformly distributed in $(0, 1)$. When a new variable is sampled from HM, there exists a probability $PAR$ (Adjustment Rate) that the variable undergoes a perturbation to enhance solution diversity and expand the search scope. Mathematically, this adjustment is implemented as follows:

$$x_{rj} = \begin{cases} x_{rj} + (2 \cdot r_1 - 1) \cdot BW, & r_2 < PAR \\ x_{rj}, & r_2 \geq PAR \end{cases} \quad (14)$$

In Equation (14), $BW$ represents the bandwidth for fine-tuning the pitch, $r_1$ and $r_2$ denote two random decimal numbers within the interval $(0, 1)$.

**Step 3:** HM Update. During this phase, the newly generated harmony undergoes evaluation. If its objective function value surpasses that of the lowest-fitness harmony currently residing in HM, the existing solution is replaced by the new candidate through a competitive exclusion mechanism.

$$X_{worst} = \begin{cases} X_{new}, & f(X_{new}) < f(X_{worst}) \\ X_{worst}, & f(X_{new}) \geq f(X_{worst}) \end{cases} \quad (15)$$

**Step 4:** Termination Condition Assessment. At this stage, the algorithm checks if the predefined iteration termination criterion has been met. In the event that the stopping condition remains unsatisfied, Steps 2 and 3 must be reiterated. Conversely, when the termination criterion is satisfied, the iterative search process terminates, and the algorithm reports the best-found harmony as the optimized solution.

The orignal HS algorithm is described in Algorithm 1.

**Algorithm 1** The pseudocode of the orignal HS algorithm.

**Input:** The parameters HMCR, PAR, BW, and $Maxiter$.
**Output:** The best harmony, i.e. the optimal solution.

```
Initialize HM with n harmonies, It = 1.
To evaluate harmonies using the fitness function f().
while (It ≤ Maxiter) do
    Generate a new harmony using (13).
    Update HM using (15).
    Select the best harmony encountered so far.
    It = It + 1
end while
```

### B. Enhanced HS Algorithm

The original HS algorithm, owing to its inherent tendency to stagnate in local optima and other methodological constraints, encounters difficulties in effectively identifying globally optimal solutions when addressing MRPP problems. This study proposes a systematic enhancement and adaptation of the conventional HS framework to more effectively accommodate the complex requirements of MRPP. To achieve this goal, several improvements have been introduced.

The proposed EHS algorithm introduces substantial methodological advancements, particularly in Steps 2 and 3 of the original HS framework. Specifically, the harmony generation mechanism in Step 2 has undergone fundamental redesign to integrate two distinct probabilistic strategies. While preserving the conventional memory-based value selection/adjustment process, we have developed two novel harmony generation paradigms. These innovations exploit: (1) the Lévy flight distribution [28]–[30] and (2) dynamic adaptation to characteristics of the current best-known harmony. The mathematical formulations for these strategies are presented in Equations (16) and (17), respectively.

$$X_{new} = X_{best} + k_1 \cdot X_i \cdot \text{Lévy(d)} \quad (16)$$

$$X_{new} = X_{best} + k_2 \cdot (X_{best} - X_i) \quad (17)$$

In Equations (16) and (17), $X_{new}$ denotes the newly generated harmony, $X_{best}$ represents the best harmony identified thus far, and $X_i$ corresponds to the $i$th harmony in the current HM. The remaining variables in these equations are defined as follows:

$$k_1 = 1 - It/Maxiter \quad (18)$$

$$\text{Lévy}(d) = \frac{u}{|v|^{\frac{1}{\beta}}} \left[ \frac{\Gamma(1+\beta) \times sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}}} \right]^{\frac{1}{\beta}} \quad (19)$$

$$k_2 = s \cdot e^{(It/Maxiter)} \quad (20)$$

In the aforementioned equations, $It$ denotes the current iteration number, while $Maxiter$ is the maximum number of iterations. The variables $u$, $v$, and $s$ are random numbers uniformly distributed in $(0, 1)$. The parameter $\beta$ is set to $1.5$, and $\Gamma(\cdot)$ represents the gamma function.

This study proposes an enhanced harmony update mechanism to improve the HS algorithm. Unlike conventional approaches that directly replace the worst harmony, the EHS employs a two-phase refinement strategy. In the secondary phase, EHS generates new harmonies matching the number of replacements, integrates them into the HM, ranks all harmonies by their objective values, and selects the top $n$ to update the HM. The complete EHS workflow is detailed in Algorithm 2.

### C. EHS for MRPP

To apply the EHS algorithm to the MRPP problem, each kinematic step of the robot must be formulated as an independent optimization subproblem. Specifically, Equation (11) computationally determines the optimal step size for every discrete robot movement, treating each locomotion decision as a distinct optimization scenario with unique constraints and objective criteria.

To determine each robot's optimal path from its initial to target position, multiple optimization rounds are required.

**Algorithm 2** The pseudocode of the EHS algorithm.

**Input:** The parameters HMCR, PAR, BW, and $Maxiter$.
**Output:** The best harmony, i.e. the optimal solution.

```
Initialize HM with n harmonies, It = 1.
To evaluate harmonies using the fitness function f().
while (It ≤ Maxiter) do
    oldHM = HM
    newHM = HM
    for i = 0 to n do
        Ps = rand() /*Strategy selection probability*/
        if (Ps < 1/3) then
            Generate a new harmony using (13).
        else
            if (Ps > 2/3) then
                Generate a new harmony using (16).
            else
                Generate a new harmony using (17).
            end if
        end if
        Replace the ith harmony in newHM with this
        new harmony.
    end for
    Evaluate the harmonies in newHM.
    Merge oldHM and newHM into tHM.
    Sort tHM based on the values of the fitness function.
    Select the top n harmonies from tHM to form the
    new HM.
    Select the best harmony encountered so far.
    It = It + 1
end while
```

The process involves iterative applications of the EHS algorithm to evaluate and select optimal movements at each trajectory step. Given the dynamic nature of the MRPP problem - where robots must navigate around each other and avoid obstacles while reaching their destinations - the number of required optimization steps may grow exponentially. Consequently, solving MRPP with EHS requires not a single optimization but rather a series of optimizations, making it a significantly complex computational challenge.

The methodology for solving the MRPP problem with the EHS algorithm consists of the following steps:

**Step 1:** Initialize the start and target positions for all robots and dynamic obstacles, specify the coordinates of static obstacles, and define the kinematic parameters (including velocities and dimensions) for all entities (robots, static obstacles, and dynamic obstacles).

**Step 2:** Initialize the HM for the MRPP problem. Each harmony in HM encodes a potential solution, representing robots' travel distances as harmony vector components. The mathematical structure of the harmony vector is formally defined in Equation (21).

$$X_i = [x_{i1}, x_{i2}, \cdots, x_{im}, y_{i1}, y_{i2}, \cdots, y_{im}] \tag{21}$$

In Equation (21), $X_i$ represents the $i$th harmony in the current HM, $m$ denotes the number of robots, and $(x_{i1}, y_{i1})$ denotes the displacement of the first robot in the $x$- and $y$-directions within the coordinate system.

**Step 3:** Determine optimal displacements for all robots using Algorithm 2. During EHS execution, Equation (11) serves as the fitness function for harmony evaluation.

**Step 4:** Update the position coordinates of robots and dynamic obstacles according to Equations (1) and (8), thereby adjusting each robot's movement trajectory.

**Step 5:** Check if all robots have reached their designated target positions. If not, iterate Steps 2-4 until convergence. Upon completion, output the optimal trajectories for all robots.

## IV. Experiments and Analysis

To evaluate the performance of the EHS algorithm for MRPP, we designed two distinct test scenarios. The proposed EHS was compared with: (1) the original HS algorithm; and (2) three state-of-the-art metaheuristics - the Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), and Arithmetic Optimization Algorithm (AOA) - all of which have demonstrated excellent performance in various optimization domains.

All algorithms were implemented in Python 3.12 and executed on a laptop with the following configuration: 13th Gen Intel® Core™ i9-13900H processor (2.60 GHz), 32GB RAM, and Windows 11 (version 24H2). The development environment utilized Visual Studio Code (v1.96.3).

### A. Algorithm Parameter Settings

To ensure fair and consistent algorithm comparisons, we maintained identical stopping criteria and population sizes across all methods, with a maximum of 1,000 iterations and a population size of 30 for each algorithm. Algorithm-specific parameters were configured according to the reference values provided in Table I.

TABLE I
Specific parameter settings for each Algorithm.

| Algorithm | Specific parameters |
|---|---|
| EHS | HMCR = 0.9, PAR = 0.1, BW = 1 |
| HS | HMCR = 0.9, PAR = 0.1, BW = 1 |
| GWO | Parameter $a$ linearly decreases from 2 to 0. |
| WOA | Parameter $a$ linearly decreases from 2 to 0. |
| AOA | $MOA\_max = 1$, $MOA\_min = 0.2$, $\alpha = 5$, $\mu = 0.499$ |

### B. Performance Evaluation Metrics

To ensure an unbiased comparison of algorithms, we assess performance through (1) visual representations of path-planning results and (2) quantitative metrics defined in Equations (22)–(25).

$$TL = \sum_{i=1}^{m} L_i \tag{22}$$

$$TS = \sum_{i=1}^{m} S_i \tag{23}$$

In Equations (22) and (23), $m$ denotes the robot count, $L_i$ the $i$th robot's path length, and $TL$ the total path length. Similarly, $S_i$ represents the $i$th robot's step count, with $TS$ being the cumulative steps.

$$TC = \sum_{t=1}^{T} C_t \tag{24}$$

In Equation (24), $T$ denotes the total number of iterations, $C_t$ represents the fitness value computed by Equation (11) at iteration $t$, and $TC$ corresponds to the total cost.

$$AT = \frac{1}{mR} \sum_{i=1}^{mR} T_i \tag{25}$$

In Equation (25), $mR$ denotes the number of independent runs, $T_i$ indicates the computational time of run $i$ ($1 \leq i \leq mR$), and $AT$ corresponds to the average time across all runs. The parameter $mR$ was set to 40 in this study.

For all four metrics, smaller values indicate better algorithm performance.

### C. Scenario 1 for MRPP

Figure 1 illustrates the initial configuration for Scenario 1, which consists of 3 robots (R1-R3), 5 static obstacles (S1-S5), and 3 dynamic obstacles (D1-D3).

The robots' initial positions are R1(15,85), R2(10,30), and R3(86,76), with corresponding target positions at (80,20), (70,90), and (22,12) respectively.

The five static obstacles are positioned at: S1(25,50), S2(75,50), S3(50,50), S4(50,75), and S5(50,25).

The three dynamic obstacles have initial positions D1(25,85), D2(10,35), and D3(80,75), with corresponding target positions at (60,15), (86,71), and (35,20) respectively.

All robots (R1-R3) have a radius of 1 unit. Static obstacles S1, S2, S4 and S5 have a 3-unit radius, while S3 has a 4-unit radius. Dynamic obstacles (D1-D3) share a 1.25-unit radius, with velocities of 1.5 units/time step (D1, D3) and 1.8 units/time step (D2).

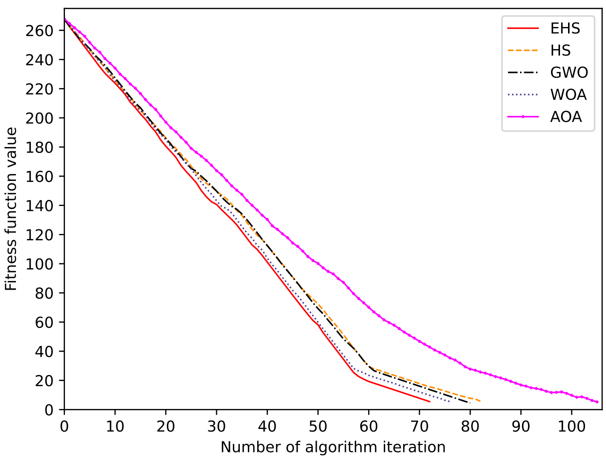



Fig. 2. Convergence curve of fitness function for MRPP of Scenario 1.

Figure 2 illustrates the convergence curve of the fitness function for five different algorithms operating under the conditions outlined in Scenario 1. From Figure 2, we can observe that the EHS algorithm achieves convergence within approximately 70 iterations. The WOA algorithm requires nearly 80 iterations, while the GWO algorithm requires approximately 80 iterations, the HS algorithm requires slightly more than 80 iterations, and the AOA requires more than 100 iterations to converge. Therefore, in terms of the convergence performance of the fitness function, the proposed EHS algorithm exhibits the fastest convergence rate.

Figure 3 demonstrates an example of paths generated by employing various algorithms for multi-robot path planning in Scenario 1.

Path planning results in Figure 3 indicate the EHS algorithm produces smoother trajectories and shorter paths than comparative algorithms. While GWO and WOA generate moderately jagged paths, HS shows intermediate performance. AOA yields the longest paths with least optimal trajectories.

Figure 4 displays boxplots comparing the total path lengths (TL) obtained by each algorithm over 40 independent executions.

The boxplots in Figure 4 demonstrate tightly clustered TL distributions with few outliers for EHS, GWO and WOA algorithms, among which EHS achieves the lowest mean total path length.

In contrast, the HS and AOA exhibit significantly poorer performance characteristics. Specifically, the AOA algorithm manifests a considerably greater variability in path lengths and produces a substantially higher mean path length relative to the other algorithms.

Figure 4 provides evidence that the proposed EHS algorithm surpasses the HS, GWO, WOA, and AOA with respect to the TL metric. Specifically, the EHS algorithm not only yields shorter mean path lengths but also demonstrates greater performance consistency compared to the other algorithms.

Figure 5 presents boxplots illustrating the distribution of step counts for individual robot movements and total step counts for all robots, following 40 independent runs of each algorithm used to solve the MRPP problem in Scenario 1.

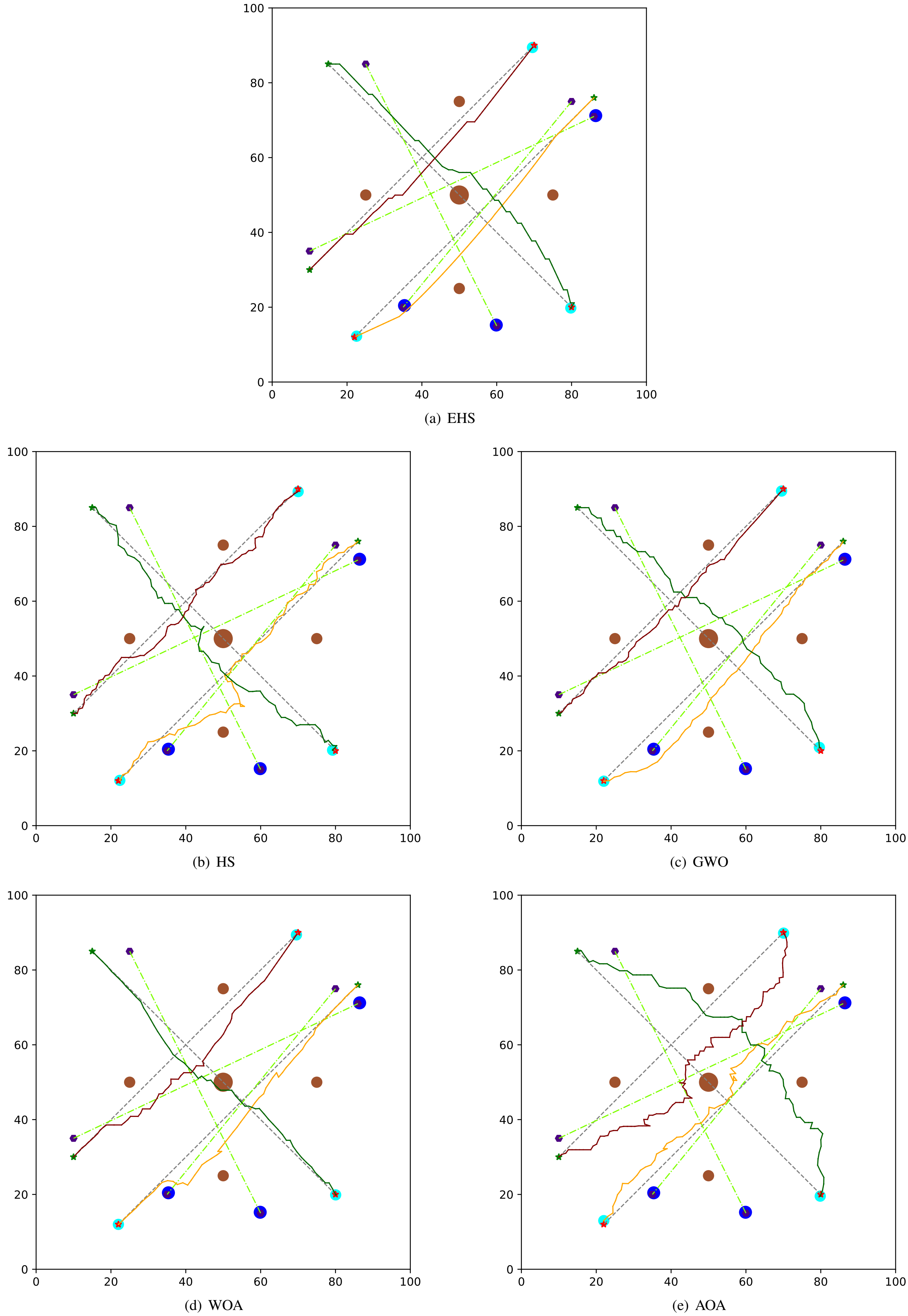
As illustrated in Figure 5, the EHS algorithm yields a lower median for both individual robot step counts and the aggregate step count across all robots compared to alternative algorithms. Furthermore, the EHS algorithm's data distribution is significantly more compact, with fewer outliers. Consequently, with respect to the movement efficiency metric—defined as the number of steps—the EHS algorithm exhibits superior performance relative to the other evaluated algorithms.

Table II presents the statistical results derived from the total cost (TC) of the fitness function following 40 independent executions of each algorithm. Figure 6 visualizes the boxplots for TC.

As illustrated in Table II, the EHS algorithm demonstrates significantly lower statistical values compared to alternative algorithms. This advantage is further corroborated by the graphical analysis in Figure 6. Consequently, with respect to the TC metric, the EHS algorithm demonstrates superior performance over the other evaluated algorithms.

Figure 7 illustrates the average computation time (AT) demanded by each algorithm during the resolution of the MRPP problem in Scenario 1, following 40 independent executions of each algorithm.

As shown in Figure 7, the EHS algorithm demands less computational time compared to the GWO, WOA, and AOA.



(a) EHS

(b) HS

(c) GWO

(d) WOA

(e) AOA

Fig. 3. Example paths derived from different algorithms applied to Scenario 1 MRPP.

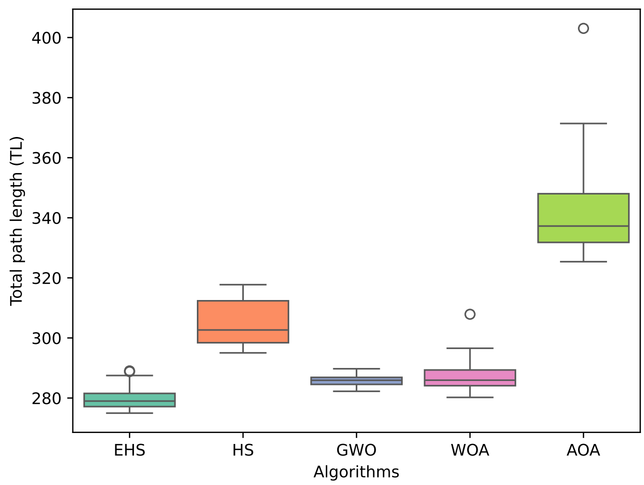
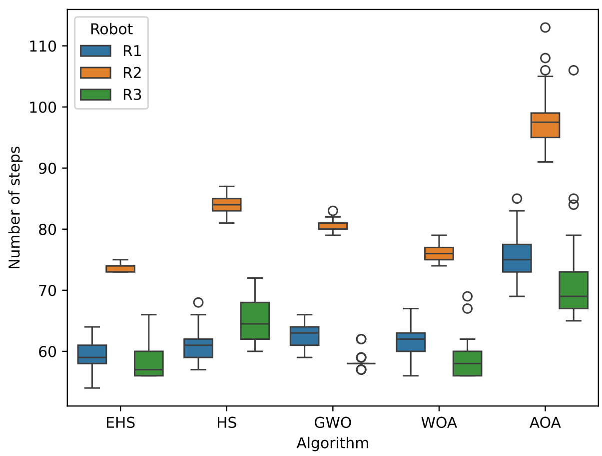
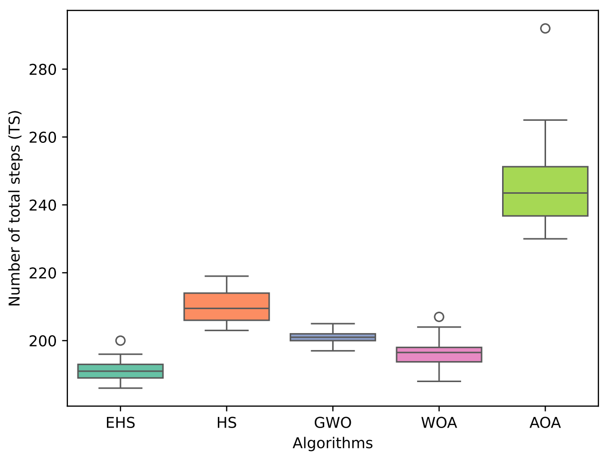



Fig. 4. Boxplots of total path length (TL) for MRPP in Scenario 1.




(a) Steps for each robot




(b) Steps for all robots

Fig. 5. Boxplots of steps for different MRPP algorithms in Scenario 1.

TABLE II
COMPARISON OF TOTAL COST (TC) STATISTICAL RESULTS FOR SCENARIO 1 ACROSS DIFFERENT ALGORITHMS.

| Measure | EHS | HS | GWO | WOA | AOA |
|---|---|---|---|---|---|
| Best | 8.48E+03 | 9.19E+03 | 9.10E+03 | 8.58E+03 | 1.04E+04 |
| Worst | 8.89E+03 | 9.95E+03 | 9.54E+03 | 9.29E+03 | 1.27E+04 |
| Median | 8.72E+03 | 9.58E+03 | 9.27E+03 | 8.98E+03 | 1.11E+04 |
| Mean | 8.72E+03 | 9.60E+03 | 9.28E+03 | 8.97E+03 | 1.12E+04 |
| Std | 1.00E+02 | 1.83E+02 | 1.04E+02 | 1.64E+02 | 4.56E+02 |

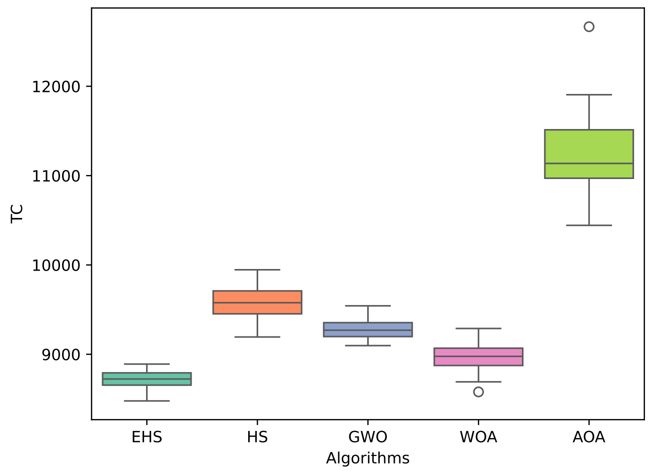



Fig. 6. Boxplots of total cost (TC) for MRPP of Scenario 1.

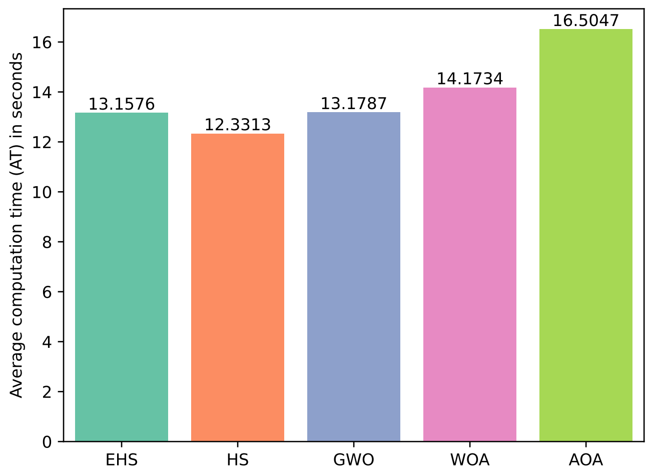



Fig. 7. Average computation time for MRPP in Scenario 1.

The value is only slightly higher than that of the standard HS algorithm. These results indicate that with respect to computational efficiency, the proposed EHS algorithm demonstrates competitive performance.

### D. Scenario 2 for MRPP

The initial configuration for Scenario 2 of the MRPP problem is visualized in Figure 8. This scenario comprises 5 robots (R1-R5), 9 static obstacles (S1-S9), and 5 dynamic obstacles (D1-D5). Specifically, nine static obstacles are positioned within the environment, while five dynamic obstacles with time-varying positions are present. The robots are designated as R1 through R5.

The initial positions of the robots are as follows: R1 is located at coordinates (65,20), R2 at (15,85), R3 at (10,30), R4 at (86,76), and R5 at (85,35). Their respective target positions are set to be: for R1 (55,85), R2 (80,20), R3 (70,90), R4 (22,12), and R5 (10,50).

Static obstacles occupy fixed locations within the scene. These are positioned at coordinates: S1 (25,50), S2 (25,75), S3 (25,25), S4 (75,50), S5 (50,50), S6 (50,75), S7 (50,25), S8 (75,25), and S9 (75,75).

Dynamic obstacles start at specific locations and move towards designated targets. The initial positions of these dynamic obstacles are: D1 at (45,15), D2 at (25,85), D3 at

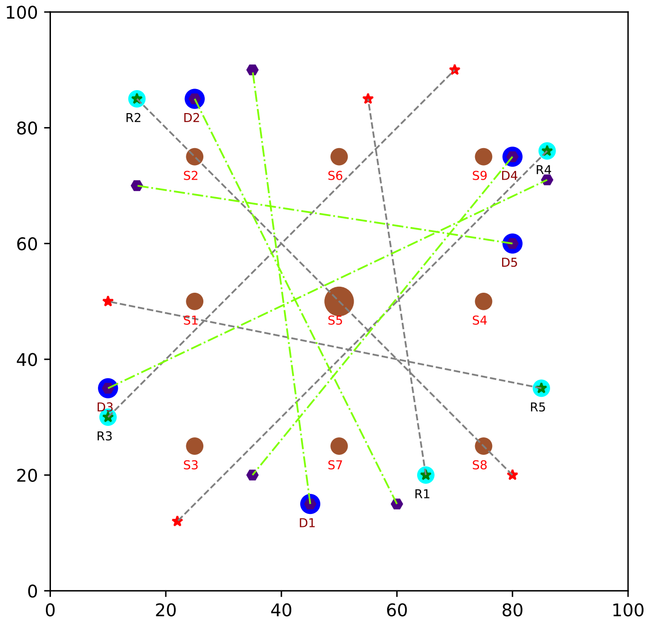



Fig. 8. Initial configuration for Scenario 2 of the MRPP.

(10,35), D4 at (80,75), and D5 at (80,60). They are expected to reach the following target positions: for D1 (35,90), D2 (60,15), D3 (86,71), D4 (35,20), and D5 (15,70).

All robots have a radius size of 1. For the static obstacles, the radius size is 3 except for S5, which has a larger radius of 4. Each dynamic obstacle has a radius size of 1.25 and moves with varying speeds: D1 moves at a speed of 1.25, D2 at 1.5, D3 at 1.8, D4 also at 1.5, and D5 at 1.25.

Figure 9 illustrates the convergence curves of the fitness function (as defined by Equation (11)) for the MRPP problem under the conditions of Scenario 2 for various algorithms.

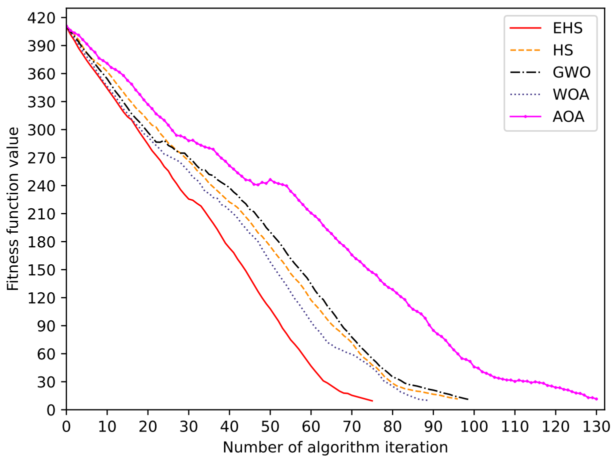



Fig. 9. Convergence curve of fitness function for MRPP of Scenario 2.

As illustrated in Figure 9, the EHS algorithm achieves a satisfactory convergence result with a notably reduced number of iterations. It is also evident from Figure 9 that the EHS algorithm requires fewer iterations compared to other algorithms, reaching convergence at approximately the 75th iteration. In comparison, the HS, GWO, and WOA require between 90 to 100 iterations for convergence, whereas the AOA demands over 130 iterations. From this, it can be inferred that when using function convergence performance as an evaluation criterion, the proposed EHS algorithm exhibits superior performance.

Figure 10 demonstrates the results obtained by applying various algorithms to solve the MRPP problem in Scenario 2.

As illustrated in Figure 10, the movement trajectories generated by the EHS algorithm for robots are smoother and more direct compared to those produced by other methods. In contrast, the routes derived from the alternative approaches exhibit greater curvature and complexity. This comparison visually demonstrates the efficiency and optimality of the EHS algorithm in addressing the MRPP problem presented in Scenario 2.

Figure 11 presents a boxplot depicting the total travel distances of robots solving the MRPP problem in Scenario 2, based on 40 independent runs of each algorithm.

Similar to Scenario 1, Figure 11 shows that the EHS algorithm produces a tightly clustered data distribution with minimal outliers, and its median is notably lower than those of other algorithms. Algorithm performances are consistent with those observed in Scenario 1. The EHS algorithm outperforms other algorithms in terms of the TL metric in Scenario 2.

Figure 12 presents boxplots of the steps taken by each robot in Scenario 2 of the MRPP problem, along with the total number of steps taken by all robots. The data shown in Figure 12 are derived from 40 independent runs for each algorithm.

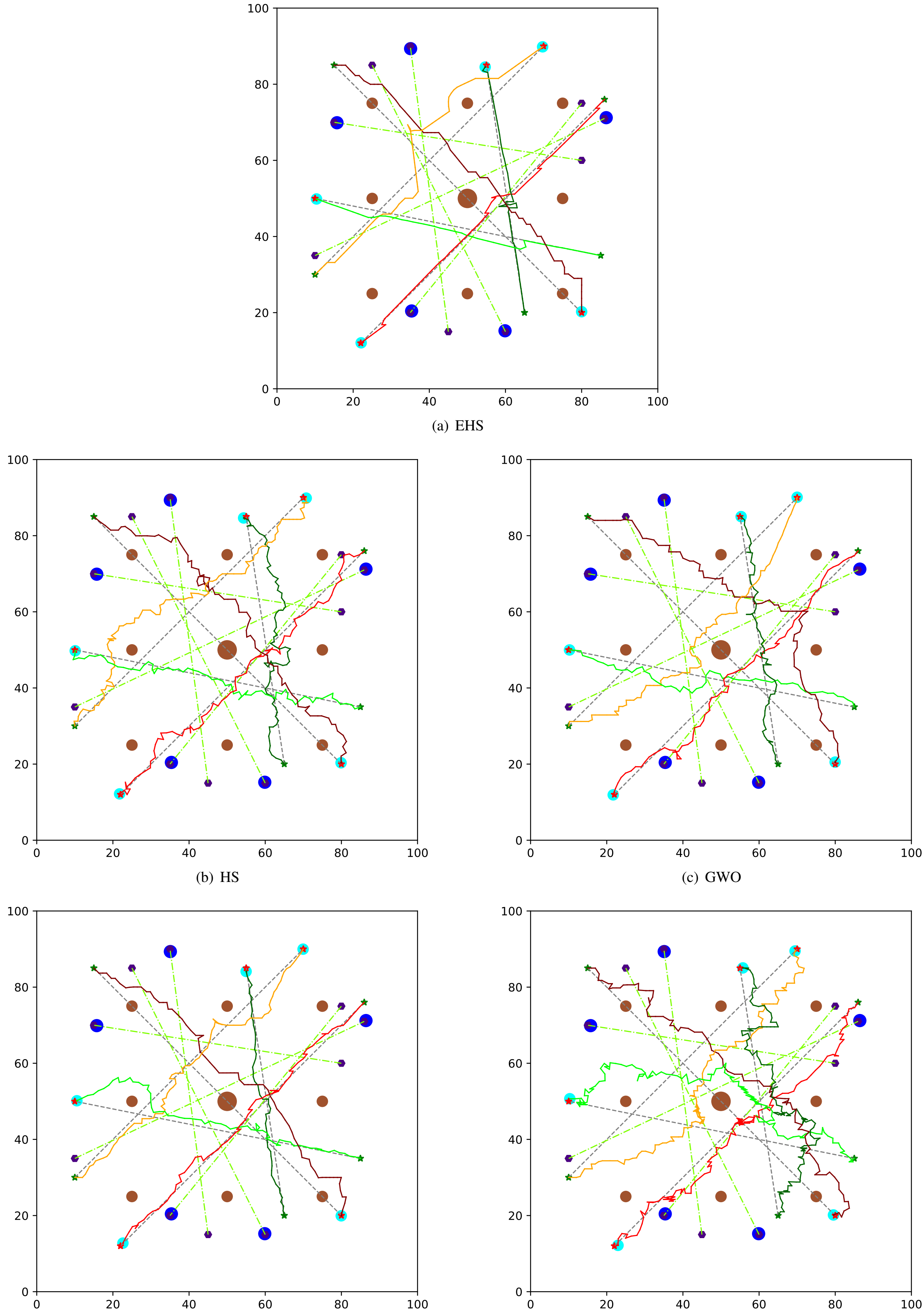
Each robot using the EHS algorithm generally reaches its target position within 100 steps, requiring fewer steps compared to the other algorithms. In terms of median values, the EHS algorithm outperforms others in efficiency. Conversely, the AOA algorithm involves more movement by the robots, with the median number of steps for each robot exceeding 100. Additionally, across the 40 independent trials, the data distribution from the EHS algorithm shows a notably tighter cluster with fewer outliers when compared to other algorithms.

Table III presents the statistical results of the total cost (TC) incurred by the fitness functions of each algorithm, obtained over 40 independent runs. Figure 13 provides a boxplot illustrating the distribution of these total costs across the different algorithms.

TABLE III
COMPARISON OF TOTAL COST (TC) STATISTICAL RESULTS FOR SCENARIO 2 ACROSS DIFFERENT ALGORITHMS.

| Measure | EHS | HS | GWO | WOA | AOA |
|---|---|---|---|---|---|
| Best | 1.33E+04 | 1.63E+04 | 1.53E+04 | 1.44E+04 | 2.04E+04 |
| Worst | 3.47E+04 | 3.74E+04 | 9.64E+04 | 1.37E+05 | 4.33E+04 |
| Median | 1.49E+04 | 1.76E+04 | 1.67E+04 | 1.69E+04 | 2.30E+04 |
| Mean | 1.86E+04 | 1.89E+04 | 1.88E+04 | 2.33E+04 | 2.44E+04 |
| Std | 7.40E+03 | 4.48E+03 | 1.25E+04 | 2.07E+04 | 4.42E+03 |

As indicated in Table III, the EHS algorithm produces lower values than the other algorithms for all statistical metrics, with the exception of the standard deviation. This observation is further supported by the boxplot presented in Figure 13.




(a) EHS

(b) HS

(c) GWO

(d) WOA

(e) AOA

Fig. 10. Example paths derived from different algorithms applied to Scenario 2 MRPP.

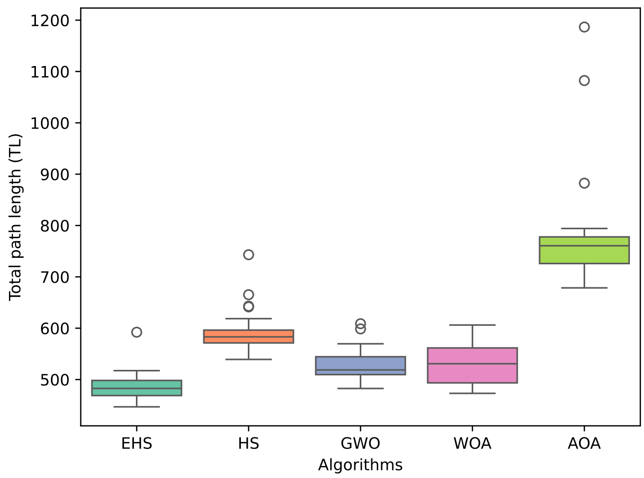
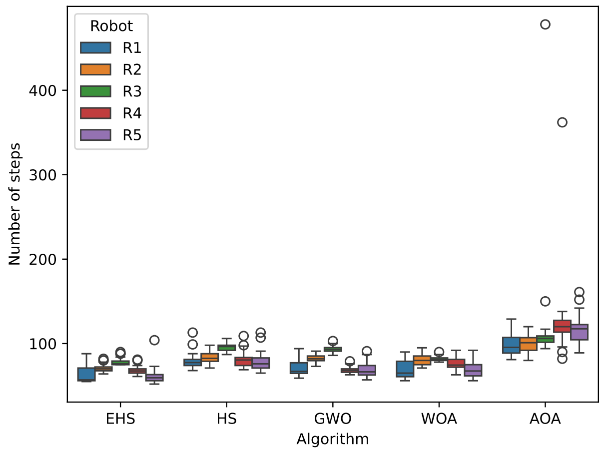
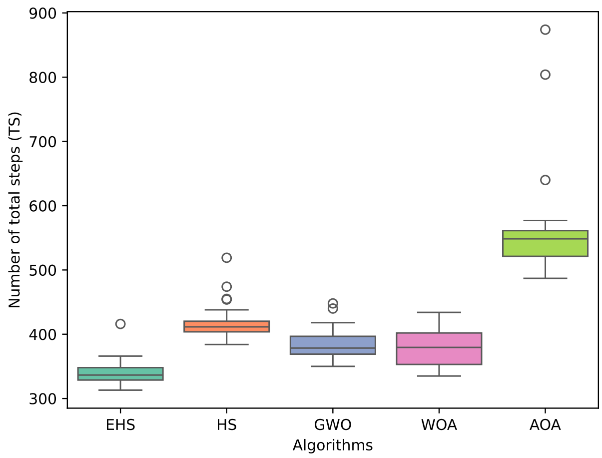



Fig. 11. Boxplots of TL for MRPP of Scenario 2.




(a) Steps for each robot




(b) Steps for all robots

Fig. 12. Boxplots of steps for different MRPP algorithms in Scenario 2.

Figure 14 depicts the average computation time (AT) needed for each algorithm to solve the MRPP problem in Scenario 2, based on 40 independent runs of each algorithm.

Figure 14 shows that the EHS algorithm requires less computation time compared to other algorithms. Thus, in terms of computational efficiency, the proposed EHS algorithm exhibits competitive performance, consistent with the findings from Scenario 1.

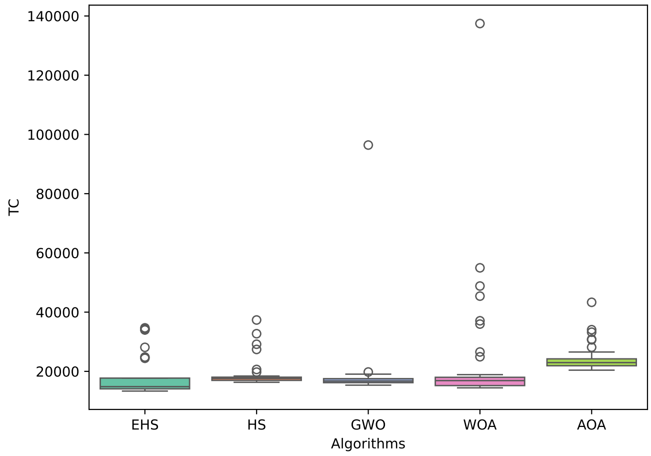



Fig. 13. Boxplots of TC for MRPP of Scenario 2.

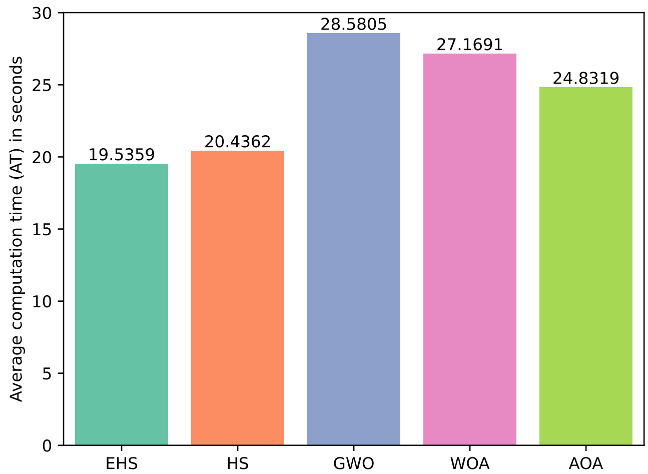



Fig. 14. Average computation time for MRPP of Scenario 2.

## V. Conclusion

In this study, we propose an enhanced harmony search (EHS) algorithm to overcome the limitations inherent in the traditional harmony search (HS) method. Additionally, we apply this enhanced algorithm to the problem of multi-robot path planning (MRPP), thereby illustrating its efficacy in addressing complex nonlinear optimization challenges.

The simulation results presented in this paper across two scenarios demonstrate that the EHS algorithm generates paths for robots which are not only smoother and more direct but also exhibit superior performance metrics compared to those generated by other algorithms. Specifically, the EHS algorithm outperforms several widely-used optimization algorithms, such as the original harmony search algorithm, grey wolf optimizer (GWO), arithmetic optimization algorithm (AOA), and whale optimization algorithm (WOA), in various critical aspects. The EHS demonstrates enhanced performance through: fewer steps taken by the robots to reach their destinations; shorter total travel distances; lower overall cost of the fitness function; and reduced computational time.

This paper presents a preliminary investigation into the effectiveness of the proposed EHS algorithm for the MRPP problem. Future research will delve deeper into the practical application of the EHS algorithm, specifically within the MRPP domain. Additionally, there are plans to extend the application of the EHS algorithm to other fields where path planning and optimization are critical.

## REFERENCES


[1] Z. Qiu, J. Long, Y. Yu, and S. Chen, "Integrated task assignment and path planning for multi-type robots in an intelligent warehouse system," *Transportation Research Part E: Logistics and Transportation Review*, vol. 194, p. 103883, 2025.

[2] B. Sahu, P. K. Das, and R. Kumar, "A modified cuckoo search algorithm implemented with SCA and PSO for multi-robot cooperation and path plannin," *Cognitive Systems Research*, vol. 79, pp. 24–42, 2023.

[3] R. Akay and M. Y. Yildirim, "Multi-strategy and self-adaptive differential sine–cosine algorithm for multi-robot path planning," *Expert Systems with Applications*, vol. 232, p. 120849, 2023.

[4] Y. Cui, W. Hu, and A. Rahmani, "Multi-robot path planning using learning-based artificial bee colony algorithm," *Engineering Applications of Artificial Intelligence*, vol. 129, p. 107579, 2024.

[5] S. Wu, A. Dong, Q. Li, W. Wei, Y. Zhang, and Z. Ye, "Application of ant colony optimization algorithm based on farthest point optimization and multi-objective strategy in robot path planning," *Applied Soft Computing*, vol. 167, Part C, p. 112433, 2024.

[6] P. Lv, G. Luo, Z. Ma, S. Li, and X. Yin, "Optimal multi-robot path planning for cyclic tasks using petri nets," *Control Engineering Practice*, vol. 138, p. 105600, 2023.

[7] S. Wen, Z. Wen, D. Zhang, H. Zhang, and T. Wang, "A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning based on transfer learning," *Applied Soft Computing*, vol. 110, p. 107605, 2021.

[8] P. K. Das and P. K. Jena, "Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators," *Applied Soft Computing*, vol. 92, p. 106312, 2020.

[9] P. K. Das, H. S. Behera, and B. K. Panigrahi, "Intelligent-based multi-robot path planning inspired by improved classical q-learning and improved particle swarm optimization with perturbed velocity," *Engineering Science and Technology, an International Journal*, vol. 19, no. 1, pp. 651–669, 2016.

[10] O. O. M. PhD, A. A. Adekunle, O. M. Olaniyan, and B. O. Bolaji, "An improved multi-objective a-star algorithm for path planning in a large workspace: design, implementation, and evaluation," *Scientific African*, vol. 15, p. e01068, 2022.

[11] X. Ji, Q. Qu, Y.-L. Ren, J.-X. Lian, and T.-R. Jiang, "Improved DBO algorithm incorporating disorienting behavior and dynamic population strategy for engineering problem solving," *Engineering Letters*, vol. 33, no. 1, pp. 90–103, 2025.

[12] P. D. Kusuma, "Best couple algorithm: A new metaheuristic with two types of equal size swarm splits," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 8, pp. 1615–1623, 2024.

[13] X. Ru, "An improved butterfly optimization algorithm for numerical optimization and parameter identification of photovoltaic model," *Engineering Letters*, vol. 33, no. 1, pp. 169–184, 2025.

[14] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "Expected-mean gamma-incremental reinforcement learning algorithm for robot path planning," *Expert Systems with Applications*, vol. 249, Part A, p. 123539, 2024.

[15] Z. W. Geem, J. H. Kim, , and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation: Transactions of The Society for Modeling and Simulation International*, vol. 76, no. 2, pp. 60–68, 2001.

[16] F. Qin, A. M. Zain, and K.-Q. Zhou, "Harmony search algorithm and related variants: A systematic review," *Swarm and Evolutionary Computation*, vol. 74, p. 101126, 2022.

[17] S. Dadashzadeh and M. Aghaie, "Performance evaluation of uranium enrichment cascades using fuzzy based harmony search algorithm," *Engineering Applications of Artificial Intelligence*, vol. 140, p. 109710, 2025.

[18] X. Sun, H. He, and L. Ma, "Harmony search meta-heuristic algorithm based on the optimal sizing of wind-battery hybrid micro-grid power system with different battery technologies," *Journal of Energy Storage*, vol. 75, p. 109582, 2024.

[19] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[20] L. Zhang and X. Chen, "Elite-driven grey wolf optimization for global optimization and its application to feature selection," *Swarm and Evolutionary Computation*, vol. 92, p. 101795, 2025.

[21] H.-C. Tsai and J.-Y. Shi, "Potential corrections to grey wolf optimizer," *Applied Soft Computing*, vol. 161, p. 111776, 2024.

[22] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[23] L. Ye, H. Ding, H. Xu, B. Xiang, Y. Wu, and M. Gong, "MFWOA: Multifactorial whale optimization algorithm," *Swarm and Evolutionary Computation*, vol. 91, p. 101768, 2024.

[24] J. Xu and B. Li, "Uncertain utility portfolio optimization based on two different criteria and improved whale optimization algorithm," *Expert Systems with Applications*, vol. 268, p. 126281, 2025.

[25] L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, p. 113609, 2021.

[26] R. Zhang, S. Mao, S. Zhao, and C. Liu, "An arithmetic optimization algorithm with balanced diversity and convergence for multimodal multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 91, p. 101724, 2024.

[27] S. Chauhan, G. Vashishtha, R. Kumar, R. Zimroz, M. K. Gupta, and A. Kumar, "A quasi-reflected and gaussian mutated arithmetic optimisation algorithm for global optimisation," *Information Sciences*, vol. 677, p. 120823, 2024.

[28] M. Elhoseny, "Intelligent firefly-based algorithm with Levy distribution (FF-L) for multicast routing in vehicular communications," *Expert Systems with Applications*, vol. 140, p. 112889, 2020.

[29] R. Zhao, Y. Wang, C. Liu, P. Hu, Y. Li, H. Li, and C. Yuan, "Selfish herd optimizer with levy-flight distribution strategy for global optimization problem," *Physica A: Statistical Mechanics and its Applications*, vol. 538, p. 122687, 2020.

[30] E. H. Houssein, M. R. Saad, F. A. Hashim, H. Shaban, and M. Hassaballah, "Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 94, p. 103731, 2020.



**Fangyan Nie** received his Ph.D degree in instrument science and technology from the Chongqing University, Chongqing, China, in 2010. His research interests include swarm intelligence optimization algorithm, data mining, computer vision, and machine learning.

He is a professor of Computer and Information Engineering College, Guizhou University of Commerce, Guiyang, Guizhou 550014 China, and a Researcher of Guizhou Academician Innovation Workstation for Disaster Remote Sensing and Control, Guiyang, Guizhou 550014 China.