



OPEN ACCESS

EDITED BY

Hassen Fourati,
Université Grenoble Alpes, France

REVIEWED BY

Mario Versaci,
Mediterranea University of Reggio Calabria, Italy
Fangwen Yu,
Tsinghua University, China

*CORRESPONDENCE

Ling Zheng
✉ lingzheng@whu.edu.cn

RECEIVED 01 August 2023

ACCEPTED 25 September 2023

PUBLISHED 17 October 2023

CITATION

Zheng L, Hong C, Song H and Chen R (2023) An autonomous mobile robot path planning strategy using an enhanced slime mold algorithm. *Front. Neurobot.* 17:1270860. doi: 10.3389/fnbot.2023.1270860

COPYRIGHT

© 2023 Zheng, Hong, Song and Chen. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

An autonomous mobile robot path planning strategy using an enhanced slime mold algorithm

Ling Zheng^{1,2*}, Chengzhi Hong³, Huashan Song⁴ and Rong Chen⁵

¹School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China, ²Shenzhen Research Institute of Central China Normal University, Shenzhen, China, ³State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan, China, ⁴Space-Time Information Department, China Mobile Intelligent Mobility Network Technology Co., Ltd., Wuhan, China, ⁵Institute of Traffic Engineering, Wuhan Technical College of Communications, Wuhan, China

Introduction: Autonomous mobile robot encompasses modules such as perception, path planning, decision-making, and control. Among these modules, path planning serves as a prerequisite for mobile robots to accomplish tasks. Enhancing path planning capability of mobile robots can effectively save costs, reduce energy consumption, and improve work efficiency. The primary slime mold algorithm (SMA) exhibits characteristics such as a reduced number of parameters, strong robustness, and a relatively high level of exploratory ability. SMA performs well in path planning for mobile robots. However, it is prone to local optimization and lacks dynamic obstacle avoidance, making it less effective in real-world settings.

Methods: This paper presents an enhanced SMA (ESMA) path-planning algorithm for mobile robots. The ESMA algorithm incorporates adaptive techniques to enhance global search capabilities and integrates an artificial potential field to improve dynamic obstacle avoidance.

Results and discussion: Compared to the SMA algorithm, the SMA-AGDE algorithm, which combines the Adaptive Guided Differential Evolution algorithm, and the Lévy Flight-Rotation SMA (LRSMA) algorithm, resulted in an average reduction in the minimum path length of (3.92%, 8.93%, 2.73%), along with corresponding reductions in path minimum values and processing times. Experiments show ESMA can find shortest collision-free paths for mobile robots in both static and dynamic environments.

KEYWORDS

autonomous mobile robots, path planning, slime mold algorithm, dynamic environment, artificial potential field

1. Introduction

Autonomous vehicles and mobile robots represent pivotal advancements in modern artificial intelligence, enabling applications across many industries, including manufacturing, agriculture, and healthcare. Developing fully autonomous mobile robot systems requires integrating diverse functionalities, from environmental perception and planning to decision-making, behavior control, and execution. Path planning serves as a foundation for autonomous mobile robot decision-making and controls, especially in

intricate and unknown dynamic environments. Optimal path planning algorithms lay the groundwork for safe navigation. However, ensuring safety as robots traverse the real world demands balancing global optimization with local reaction (Wang et al., 2018). Significant research has focused on mobile robot path planning, developing numerous theories and methods addressing its multifaceted challenges (Hewawasam et al., 2022). Path planning presents difficulties due to its non-linear, multi-constraint nature within uncertain environments, making optimal solutions elusive using traditional techniques (Sanchez-Ibanez et al., 2021). Conventional algorithms like the A* (Jin et al., 2022), probabilistic roadmaps (Kavraki et al., 1996), fuzzy reasoning (Liu Z. X. et al., 2022), artificial potential field (Das et al., 2022), and rapidly-exploring random trees (Zhang et al., 2021) have limitations addressing complex environments. Swarm intelligence algorithms excel through self-organizing population interactions informed by mathematical rulesets (Xu et al., 2020). They generate optimal solutions via biological cluster emulation (Rafai et al., 2022). For example, the monarch butterfly optimization (MBO) algorithm simulates butterfly swarms' behavior (Bao et al., 2020). Artificial bee colony (ABC) modeling imitates bee colonies (Liang and Lee, 2015), while the moth flame optimization (MFO) algorithm recreates moths' spiral motions (Zhang et al., 2020). Such bio-inspired techniques offer novel approaches for complex path planning problems (Versaci et al., 2020).

Several studies confirm swarm intelligence's benefits for autonomous mobile robot decentralization (Fragapane et al., 2021). Swarm intelligence algorithms establish mathematical models using rule sets and elements. These algorithms iterate by replacing the current solution with a newly generated one, repeating the optimization process until optimal solutions are obtained or a maximum number of iterations is reached. In a study Cheng et al. (2020), a genetic algorithm (GA) accomplished path planning for a reconfigurable hinged-Tetromino robot, addressing multi-objective global optimization. However, the GA optimization is susceptible to local optima and exhibits slow convergence. To overcome this, an enhanced ant colony optimization (ACO) algorithm incorporating a time taboo grid strategy demonstrated success in dynamic environments (Xiong et al., 2021). However, ACO has a lengthy calculation cycle. Another study applied particle swarm optimization (PSO) with two objective functions—distance and risk—to determine the safest and shortest path by predicting random obstacle changes (Al Hilli et al., 2021). However, PSO is prone to premature convergence in complex problems. An improved cuckoo optimization algorithm (COA) achieved constrained path planning in simulations and real environments through an enhanced objective function

(Mohanty, 2020). However, COA left room for improved global convergence efficiency. In a related study Dereli (2022), the whale optimization algorithm (WOA) enhanced convergence by strictly following the leader to prey via Euclidean distance. However, WOA faces frequent local optima challenges. While swarm intelligence algorithms have proven effective in simulations and experiments, opportunities remain to address premature convergence, dynamically balance exploitation/exploration, and overcome localization. Focusing on such areas could further optimize performance. Therefore, improved meta-heuristic path planning for autonomous mobile robots remains an active research direction.

The performance of basic swarm intelligence algorithms generally improved additional components or combining algorithms (Mac et al., 2016). For instance, an enhanced diversity PSO algorithm in the diversity PSO algorithm to ensure diverse peaks and prevent iteration stagnation (Fernandes et al., 2022). Another study combined the bat optimization algorithm with CAO to select optimal qualities, reducing path calculation time (Saraswathi et al., 2018). A new WOA variant incorporated an artificial potential field to improve dynamic obstacle avoidance (Dai et al., 2023). Another approach integrated the Bacterial Foraging Optimization (BFO) algorithm with the Lévy flight to reduce iterations and accelerate convergence (Pang et al., 2019). Additionally, the gray wolf optimization algorithm (GWO) combined with PSO in the PSO_GWO introduced PSO to calculate gray wolf positions, effectively addressing local optimization (Teng et al., 2019). Moreover, meta-heuristic optimization processes are inherently stochastic—striking the right balance between exploration and exploitation proves crucial in optimizing algorithms (Mirjalili et al., 2014).

The slime mold algorithm (SMA) is a meta-heuristic inspired by slime mold behavior in nature (Li et al., 2020). It exhibits fewer parameters, robustness, and exploration ability, making it promising for global optimization. In benchmark function sets like IEEE CEC 2014, SMA outperformed algorithms such as the WOA, MFO, and GWO. However, SMA also demonstrates weaknesses, such as slow convergence or inability to reach global optima in certain states or functions. Like all algorithms, SMA has limitations in specific scenarios or problem types. Regarding path planning, SMA surpasses particle swarm optimization (PSO) and artificial bee colony (ABC) algorithms in static environments (Agarwal and Bharti, 2021). Further research is needed to address dynamic environment challenges. Improving convergence efficiency and achieving effective dynamic obstacle avoidance are critical for mobile robot path planning. Dedicated efforts developing innovative techniques and algorithms that can adapt swiftly to dynamic changes are needed.

To improve the efficiency and effectiveness of autonomous mobile robot path planning in dynamic environments and achieve the best paths, this paper introduces an enhanced version of the Slime Mold Algorithm (ESMA). The proposed planning process involves three stages: (1) Processing the navigation environment to create a movement position map for the robot, (2) Planning the trajectory between the start and endpoint of the desired movement, and (3) Post-processing the determined path to ensure the shortest route while considering the inherent limitations of the autonomous mobile robot. The final stage will not be discussed in this article. The main contributions of this study can be summarized as follows:

Abbreviations: ABC, Artificial bee colony; AGV, Automated guided vehicle; ACO, Ant colony optimization; AGDE, Adaptive guided differential evolution algorithm; BFO, Bacterial foraging optimization; COA, Cuckoo optimization algorithm; ESMA, Enhanced slime mold algorithm; GA, Genetic algorithm; GWO, Gray wolf optimization; LRSMA, Lévy flight-rotation slime mold algorithm; MBO, Monarch butterfly optimization; PSO, Particle swarm optimization; PSO_GWO, Gray wolf optimization algorithm combined with particle swarm optimization; SMA, Slime mold algorithm; SMA-AGDE, Slime mold algorithm combined with adaptive guided differential evolution algorithm; WOA, Whale optimization algorithm.

- a) Adaptive technology using a linear decreasing strategy to increase random search and enhance the global capability.
- b) Improved potential field factor incorporating attraction and repulsion for dynamic obstacle avoidance.

The structure of this paper is as follows: Section 2 provides an overview of the enhanced Slime Mold Algorithm (ESMA). Section 3 presents the underlying theory of the SMA and its improvement. Section 4 conducts experiments demonstrating ESMA's dynamic path planning benefits for mobile robots. Finally, Section 5 summarizes key conclusions regarding the significance of the findings.

2. Related work

Like other meta-heuristic algorithms, the SMA involves three key aspects when dealing with optimization: exploration, exploitation, and the transaction between the two stages (Cai et al., 2020). Exploration dedicates to identifying potential areas within the entire search space that may contain the optimal solution, aiming to progressively narrow this region. Conversely, exploitation focuses on finding the best solution within the defined feasible area (Alyasseri et al., 2022). The smooth transition between the exploration and exploitation stages plays a pivotal role in achieving a well-balanced search strategy for the algorithm (Rodriguez-Molina et al., 2020). Optimizing exploration and exploitation strategies provide an important approach for swarm intelligence algorithm optimization (Lin and Gen, 2009).

A series of proposed improved SMAs have addressed the issues of optimization and slow convergence while also considering obstacle avoidance in the path planning. Yu et al. (2021) proposed a strategy combining exploration and exploitation using quantum rotation gates and a water cycle approach to enhance convergence speed. To mitigate premature convergence (Rizk-Allah et al., 2022), implemented a strategy incorporating chaotic search and cross-reverse enhancement, effectively expanding the search space and improving non-linear convergence accuracy. Researchers introduced reverse learning to improve global exploration ability. Liu and Liu (2022) combined quasi-reverse learning and quasi-reflective learning techniques to enlarge population searching range. Additionally, they utilized the unscented transformation sigma point to improve stability and alleviate stagnation. Nguyen et al. (2020) proposed an enhanced SMA incorporating opposition-based learning and adjusting individual position update weight coefficients to enhance performance. Houssein et al. (2022) integrated modified opposition-based and orthogonal learning techniques to improve accuracy. These studies demonstrated advantages in balancing exploration and exploitation stages. However, configuring the reverse learning search space poses a challenge directly impacting effectiveness.

The crossover and mutation operators in the adaptive guided differential evolution algorithm (AGDE) possess strong local optimization capabilities, making them effective for enhancing the local search capability of SMA. Houssein et al. (2021) utilized AGDE to promote population diversity and overcome premature convergence, resulting in SMA-AGDE included here for comparison. The Lévy flight strategy is a well-known approach

for improving global search ability in intelligent optimization algorithms (Liu J. X. et al., 2022). Building on the Lévy search (Zheng et al., 2023), introduced rotation perturbation with local optimization ability, yielding the Lévy flight-rotation SMA (LRSMA) with higher convergence accuracy. LRSMA proves particularly effective for static environment path planning problems. Consequently, LRSMA is also included as one of the comparative algorithms in this study.

While these improved SMAs can enhance the algorithm's convergence speed, further research on obstacle avoidance in dynamic environments is needed. Hence, this study proposes an ESMA considering the environment in two dimensions and efficiently plan autonomous mobile robot paths dynamically. To verify the efficiency and utility of the ESMA, experimental comparative studies evaluated it against SMA, SMA-AGDE, and LRSMA.

3. Mathematical model of algorithm

3.1. SMA and its advanced approach

3.1.1. SMA

The SMA abstracts and simulates the foraging behavior and morphological changes observed in slime molds through a mathematical model. During foraging, the SMA creates a venous network that connects various food sources. The higher the quality and density of the food source, the thicker the venous network. As slime molds locate food, the SMA dynamically adjusts the cytoplasm flow rate within veins and modifies network thickness using oscillations from a biological oscillator. When food concentration is higher, the wave generated by the biological oscillator becomes stronger, resulting in faster cytoplasm flow and thicker veins.

The SMA maintains a balance between exploration and exploitation throughout the search process. Even after finding food sources, there remains a probability for slime molds to explore unknown areas, ensuring the algorithm continues searching for potentially better solutions. The search mechanism of the SMA can be summarized as follows (Li et al., 2020):

$$X(t+1) = \begin{cases} rand \times (UB - LB) + LB, & rand < z \\ X_{best}(t) + vb \\ \times (W \times X_{r1}(t) - X_{r2}(t)), & rand \geq z \text{ and } rand_1 < p \\ vc \times X(t), & rand \geq z \text{ and } rand_1 \geq p \end{cases} \quad (1)$$

$$W(SIndex(i)) = \begin{cases} 1 + r_2 \times \lg\left(\frac{bF - S(i)}{bF - wF} + 1\right), & \text{condition} \\ 1 - r_2 \times \lg\left(\frac{bF - S(i)}{bF - wF} + 1\right), & \text{others} \end{cases} \quad (2)$$

$$SIndex(i) = \text{sort}(N) \quad (3)$$

In the SMA, the following equations and parameters are utilized:

t : Current iterative index.

$X_{r1}(t)$ and $X_{r2}(t)$: Positions of two randomly selected individuals from the slime molds.

$X(t+1)$: Updated position of the individual after the current iteration.

$X_best(t)$: Position of the individual with the highest food concentration in the t th iteration (optimal position).

vc : Control parameter measuring the utilization of historical records by individual slime molds, decreasing linearly in the range of $[1, 0]$.

UB and LB : Upper and lower bounds of the search space, respectively.

$rand$: Random number between 0 and 1.

$rand_1$: Random number between 0 and 1, representing the switching probability between exploration and exploitation modes.

z : The proportion of slime molds randomly distributed in all populations.

W : Weight coefficient, simulating the change in the biological oscillator's frequency with the quality and density of the food during foraging.

r_2 : Random number, either 0 or 1.

$SIndex(i)$: Index of slime mold individuals after sorting.

bF and wF : Optimal and worst fitness values in the current iteration, respectively.

lg : Value used to slow down the rate of change.

$condition$: Slime mold individuals whose fitness values are in the top half of the population.

$others$: Remaining slime mold individuals.

The exploration stage is active when $rand_1 < p$, while the exploitation stage occurs when $rand_1 \geq p$.

P and vb are control parameters, and the formula is as follows:

$$p = \tanh |S(i) - DF|, i \in \{1, 2, \dots, N\} \quad (4)$$

$$a = \text{arctanh} \left(-\frac{t}{IT_{\max}} + 1 \right) \quad (5)$$

i : Index of slime molds.

$S(i)$: Fitness value of the i th slime mold in the current iteration.

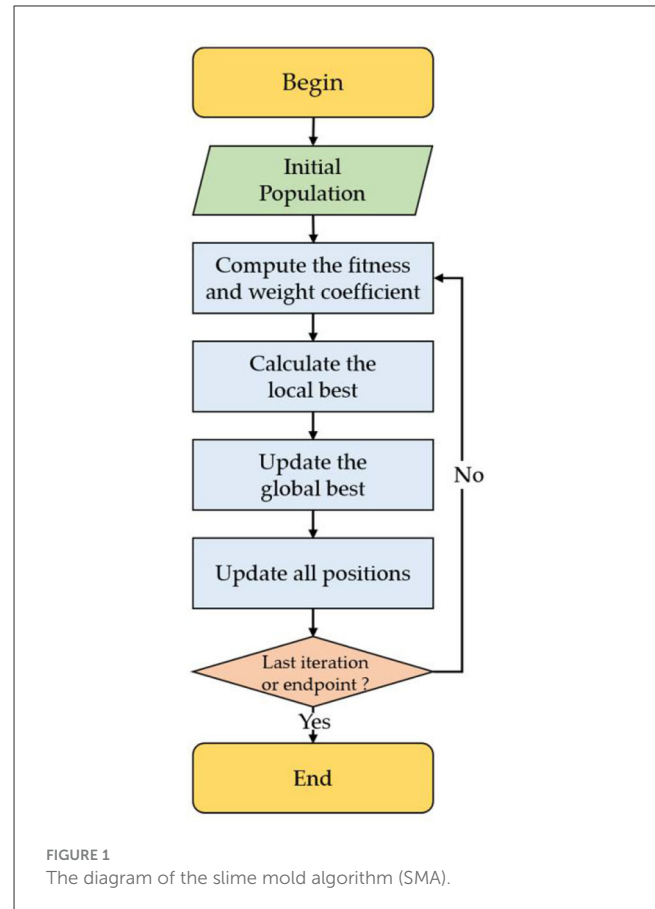
DF : Optimal fitness value among all iterations.

vb : Random value in $[-a, a]$, with the range decreasing as the value of a decreases.

IT_{\max} : Maximum number of iterations.

Figure 1 shows the SMA detailed steps, which involves the following processes:

- (1) Initialization.
- (2) Obtain each individual's position through random search, calculating corresponding fitness values, and local best solution.
- (3) Renew the global best consequence.
- (4) Determine whether the maximum search iteration limit is reached. If so, continue updating individuals based on value and Equation (1) until the best solution is obtained.



3.1.2. SMA-AGDE

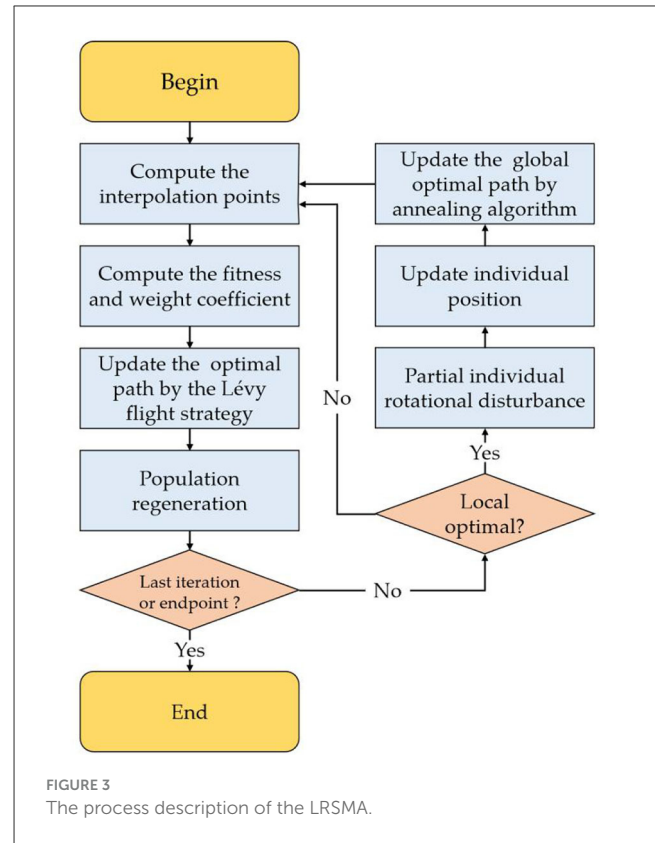
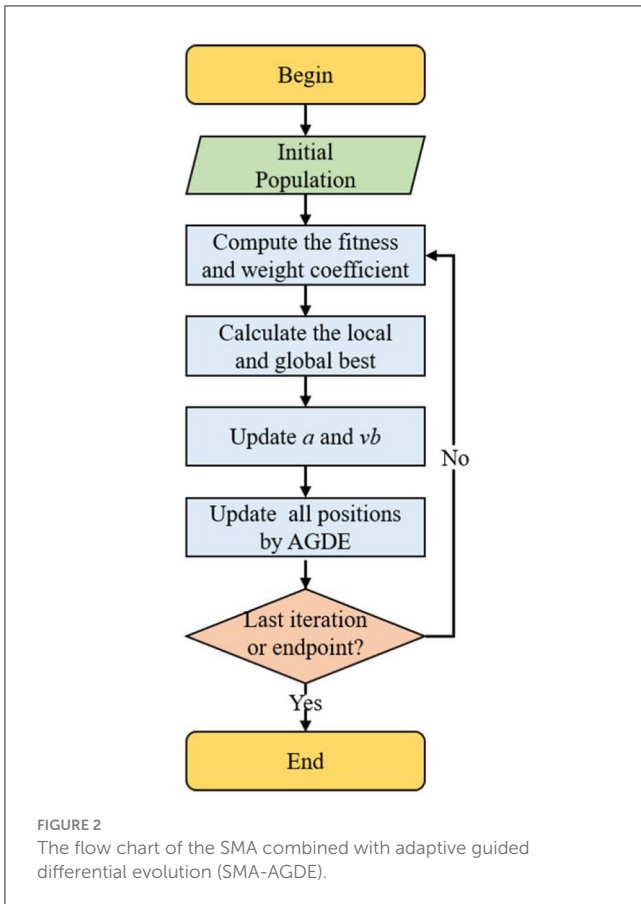
AGDE demonstrates robust convergence ability through mutation, crossover and selection (Mohamed and Mohamed, 2019). In the SMA-AGDE, the initial population forms via random sampling, hybridizing SMA and AGDE. AGDE's mutation and crossover process serves as SMA's individual update method. This integration aims to increase diversity, enhance local search capability, and prevent premature convergence. By combining SMA and AGDE strengths, the SMA-AGDE achieves improved performance in exploration, exploitation, and overall optimization.

Figure 2 illustrates the SMA-AGDE search update mechanism. The main steps are (Houssein et al., 2021):

- (1) Initialization.
- (2) Obtain each individual's position through random search, calculate fitness value, and find local best solution.
- (3) Update parameters a and vb .
- (4) Update the slime mold individuals according to AGDE.
- (5) Repeat steps (2)–(4) until the best solution occurs when iterations reach the maximum limit.

3.1.3. LRSMA

The LRSMA updates slime mold positions by introducing a variable neighborhood Lévy flight strategy to improve the global optimization ability of the SMA. The LRSMA applies a rotation disturbance mutation approach considering local optimization



tolerance to disturb each slime mold, expand population local search range, and enhance algorithm development ability.

Figure 3 shows the diagram of the LRSMA (Zheng et al., 2023), whose steps are as follows:

- (1) Initialize.
- (2) Calculate the global best solution. Calculate and sort fitness values to obtain the best and worst fitness value. Update each individual's weight coefficient W (*SIndex*) and position.
- (3) Re-update global best solution. Update optimal position according to variable neighborhood Lévy flight strategy.
- (4) Reconstruct convergence population. Judge the algorithm convergence. If converged, rotate some individuals. Recalculate the global best position by simulating annealing approach.
- (5) Regenerate the population.
- (6) Repeat steps (2)–(5) until the iteration number reaches the maximum limit and output the global best solution.

3.2. The proposed ESMA

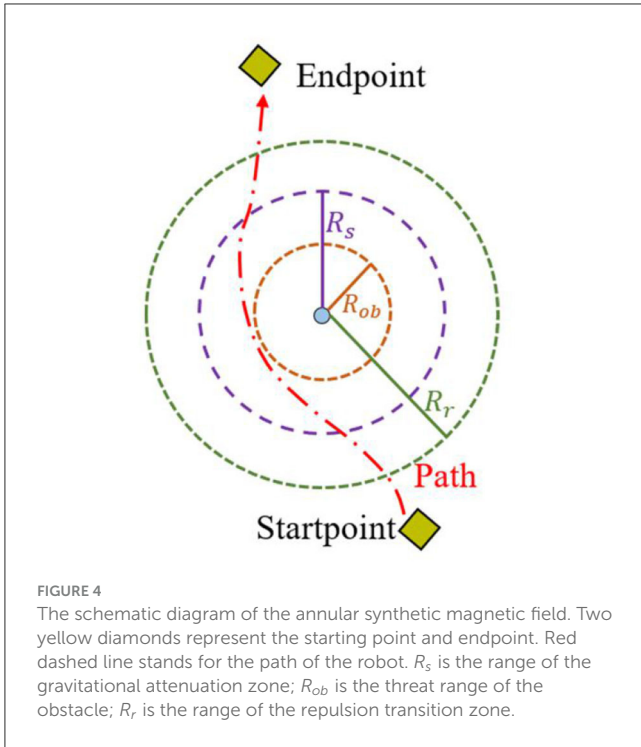
The ESMA proposes a new adaptive technique that utilizes the same weight coefficient processing as SMA. The ESMA uses the same fitness calculation as LRSMA in path planning. However, the ESMA's data update method after local optimization differs from LRSMA.

The free space method models the autonomous mobile robot's driving environment, where achievable space is blank and unachievable space contains obstacle area. To simplify calculation, a point-like object moving on a two-dimensional platform simulates the robot. Obstacles enlarge to half robot width on mapping to the environment, represented by circles of radius determining obstacle space (Na et al., 2022).

This paper aims to search the drivable area in the free space through the algorithm to obtain an effective start-end trajectory plan. The trajectory consists of discontinuous points forming a continuous shortest path on the map. This process solves three problems: obstacle avoidance, global path obtaining, and local path planning (Dai et al., 2023).

In the SMA, z represents the proportion of randomly distributed slime mold individuals among the total population. As iterations increases, z remains constant. Early on, the SMA quickly approaches the best position of the current population. However, when falling into local optimization, it may jump from local optima through random search and explore unknown regions with certain probability. Since z directly determines random search probability, the fixed parameter z cannot fully meet the search requirements. To address this, an adaptive technique linearly increases z within a constrained interval as iterations increase. The adaptive update process of z is as follows:

$$z = z_0^* \left(1 + \frac{t}{IT_{max}} \right) \tag{6}$$



where, z_0 is a fixed parameter, t is the current iteration, and IT_{max} is the maximum number of iterations. The value of z is the smallest in the early stage of the search, and the value of z becomes larger in the later stage of the search.

In the later stage of the SMA, the population tends to converge to the best individual, which reduces diversity and increasing susceptibility to local optimization. When SMA falls into local optimization, adjusting the parameter z alone may be insufficient for escape. In the ESMA, gravitational and repulsive potential fields are established. A synthetic potential field model is constructed based on these, on these, combining the strengths. The gravitational field improves local planning efficiency while the repulsive field enables obstacle avoidance in local robot path planning. The introduced potential integrated field serves as another ESMA optimization variable.

Traditional artificial potential fields suffer from issues such as unreachable targets and local minimum traps. Therefore, this paper employs a circular synthesis-enhanced potential field method to construct a comprehensive potential field model that neglects long-range repulsion and weakens the attraction from obstacle points. As shown in Figure 4, the potential synthetic field in the annular region can provide a non-oscillatory trajectory (Liu et al., 2020). In the robot local path planning process, the repulsive potential field is weakened when the robot is far from obstacles to prevent premature deviation from the trajectory. Simultaneously, the attractive potential field is weakened when approaching obstacles to prevent local oscillations. The synthetic potential in the annular region is given by Liu et al. (2020).

$$U(X) = \varepsilon_a * U_a(X) + \varepsilon_r * U_r(X) \tag{7}$$

$$U_a(X) = \begin{cases} -\eta_a * \left(1 - \frac{d(X, X_g)}{d(X_{ob}, X_g)}\right)^{q_a} & \\ 0, & d(X, X_{ob}) < R_{ob} \\ \frac{1}{2} \left(\sin\left(\frac{d(X, X_{ob}) - R_{ob}}{R_r - R_{ob}} \cdot \pi - \frac{\pi}{2}\right) + 1\right), & R_{ob} \leq d(X, X_{ob}) < R_s \\ 1, & otherwise \end{cases} \tag{8}$$

$$U_r(X) = \begin{cases} -\eta_r * \left(\frac{R_{ob}}{d(X, X_{ob})}\right)^{q_r} & \\ 1, & d(X, X_{ob}) < R_s \\ \frac{1}{2} \left(\cos\left(\frac{d(X, X_{ob}) - R_s}{R_r - R_s} \cdot \pi\right) + 1\right), & R_s \leq d(X, X_{ob}) < R_r \\ 0, & d(X, X_{ob}) \geq R_r \end{cases} \tag{9}$$

where $U(X)$ is the potential synthetic field of the robot at position x ; ε_a is the gravitational potential field coefficient; $U_a(X)$ is the potential gravitational field; $U_r(X)$ is the potential repulsive field; ε_r is the coefficient of the repulsive potential field; η_a is the gravitational field coefficient; X_{ob} is the obstacle position; X_g is the end position; q_a is a positive integer; R_{ob} is the threat range of the obstacle; R_s is the range of the gravitational attenuation zone; $d()$ is the Euclidean distance of two points; η_r is the repulsive field coefficient; q_r is a positive integer; R_r is the range of the repulsion transition zone.

To smoothly navigate around obstacles and avoid local oscillations, improvements have been made to the attractive potential field function by introducing a weakened attractive potential field region near obstacles. To prevent premature deviation from the desired path, the repulsive potential field has been enhanced through a potential field smoothing transition strategy. The potential gravitational function $F_a(X)$ and the potential repulsive function $F_r(X)$ can be obtained by deriving the gravitational and repulsive potential functions, respectively. The mathematical models of gravity and repulsion are as follows:

$$F_a(X) = \begin{cases} 0, & d(X, X_{ob}) < R_{ob}, \\ \frac{1}{2} \left(\sin\left(\frac{d(X, X_{ob}) - R_{ob}}{R_r - R_{ob}} \cdot \pi - \frac{\pi}{2}\right) + 1\right) * \frac{q_a}{d(X_{ob}, X_g)} * \left(1 - \frac{d(X, X_g)}{d(X_{ob}, X_g)}\right)^{q_a - 1}, & R_{ob} \leq d(X, X_{ob}) < R_s, \\ \frac{q_a}{d(X_{ob}, X_g)} * \left(1 - \frac{d(X, X_g)}{d(X_{ob}, X_g)}\right)^{q_a - 1}, & otherwise, \end{cases} \tag{10}$$

$$F_r(X) = \begin{cases} \frac{q_r * R_{ob}^{q_r}}{d^{q_r + 1}(X, X_{ob})}, & d(X, X_{ob}) < R_s, \\ \frac{1}{2} \left(\cos\left(\frac{d(X, X_{ob}) - R_s}{R_r - R_s} \cdot \pi\right) + 1\right) * \frac{q_r * R_{ob}^{q_r}}{d^{q_r + 1}(X, X_{ob})}, & R_s \leq d(X, X_{ob}) < R_r, \\ 0, & d(X, X_{ob}) \geq R_r, \end{cases} \tag{11}$$

where $F_a(X)$ represents gravity, and $F_r(X)$ represents repulsion.

The directions of gravity and repulsion correspond to the fastest decreasing directions of potential gravitational and repulsive

energies, respectively, represented by the negative gradients of the potential gravitational field and potential repulsion field. The resultant force acting on the robot can be obtained via vector superposition of the gravity and repulsion acting on it.

In this paper, the robot's position changes detect local optimization presence in the mobile robot. When the optimal solution of the robot oscillates near a certain point within a small range, it is considered to have fallen into the local optimization. Hence, it is necessary to guide the robot with potential synthetic field local planning until escaping local optimization.

Algorithm 1 shows the ESMA pseudocode.

```

1: Set the obstacle avoidance threshold
    $D_{min}$  and initialize the population
2: for  $it =: IT_{max}$  do
3:   for each individual do
4:     Calculate the fitness value of each
       individual
5:     Update the local best solution
6:   end for
7:   Recalculation the global best solution
8:   if the global best solution reaches the
       endpoint then
9:     return the global best solution
10:  end if
11:  if  $d(X_{best}(t), X_{best}(t-1)) < D_{min}$  then
12:    repeat add potential field and
       update the global best solution
13:    until  $d(X_{best}(t), X_{best}(t-1)) \geq D_{min}$ 
14:  end if
15:  Check if any individual goes out of
       search space and modify it
16:  Calculate the fitness of each
       individual
17:  Update the population
18: end for
19: return the global best solution

```

Algorithm 1. The pseudocode of the ESMA.

The steps of the ESMA are as follows:

- (1) Initialization. Randomly generate the initial myxobacteria population and update the fitness value of each smile mold individual.
- (2) Update the local and global best solutions according to Equations (1)–(6).
- (3) If the algorithm falls into the local optimization, re-update the global best solution according to Equations (7)–(11) and proceed to step (4). Otherwise, proceed to step (4).
- (4) Update the population.
- (5) Continue iterating through steps (2)–(4) until the best solution is obtained, as long as the number of searches is less than the specified number of iterations.

4. Experiment results and analysis

4.1. Scenarios and parameter setting

The experimental scenario simplified the obstacles as circles with different radii. To represent the position of the robot, the coordinates of its center point were utilized, irrespective of its size. To verify the algorithm's robustness, two scenarios were designed, as shown in Figure 5, with different obstacle distributions. In Figure 5, the obstacles are punctate in scenario 1, while in scenario 2, they have a banded distribution. The experimental area is 800*800 cm, and the white grid represents the accessible area.

The obstacle area is depicted by a blue circle, where the size of the circle corresponds to the extent or coverage of the obstacle. Obstacles with larger radii indicate larger obstacles. The radii are 75 and 50 cm. The starting point and endpoint coordinates are $[-800, -800]$ and $[800, 800]$, respectively, represented by a yellow diamond. The unit of measurement is cm.

To provide a comprehensive evaluation of the algorithm's performance, this study conducted comparisons between the ESMA and other algorithms, namely SMA (Agarwal and Bharti, 2021), SMA-AGDE (Houssein et al., 2021), and LRSMA (Zheng et al., 2023). The parameter settings for all algorithms were determined based on the reference papers. In SMA, the parameter is $z = 0.3$. The parameters of SMA-AGDE are set as $z = 0.3$, $CR1 \in [0, 0.05]$, $CR2 \in [0.9, 1.0]$. In LRSMA, the parameters are $z = 0.3$, $F_{min} = 1$, $p_r = 0.5$. The parameters of ESMA are set as $z \in [0.3, 0.6]$, $F_{min} = 1$, $D_{min} = 10$, $q_a = 1$, $q_r = 1$, $R_{ob} = 200$, $R_s = 250$, $R_r = 400$. The evaluation criteria in this study included path length, planning time, and planning success rate. The minimum value, average value, and standard deviation of these metrics were calculated and compared to assess the algorithm's performance across these indices.

4.2. Simulation experiment and analysis

4.2.1. Experimental environment

The running environment of the simulation experiment platform was as follows: the CPU was an Intel Core i7-6500U, the frequency was 2.50 GHz, the memory was 8 GB, the programming language was Python 3.9, the operating system was Windows10 64 bit, and the compilation software was Visual Studio Code 1.6.1.

4.2.2. Simulation experiments and analysis

The experiments conducted in static environments involved the application of the SMA, SMA-AGDE, LRSMA, and ESMA algorithms to two environments as shown in Figure 5. The algorithm's universality was confirmed through 100 simulation experiments. Record the data for 100 successful runs (excluding local optima), including the optimal path length and processing time. Also, calculate the success rate in a 100-run algorithm. The path length serves as a reflection of the algorithm's ability to find the optimal path, with shorter lengths indicating better solutions. Optimization time refers to the time taken by the algorithm to find the optimal path. Analyzing the data from 100 experimental runs allows for the calculation of metrics such as the minimum path

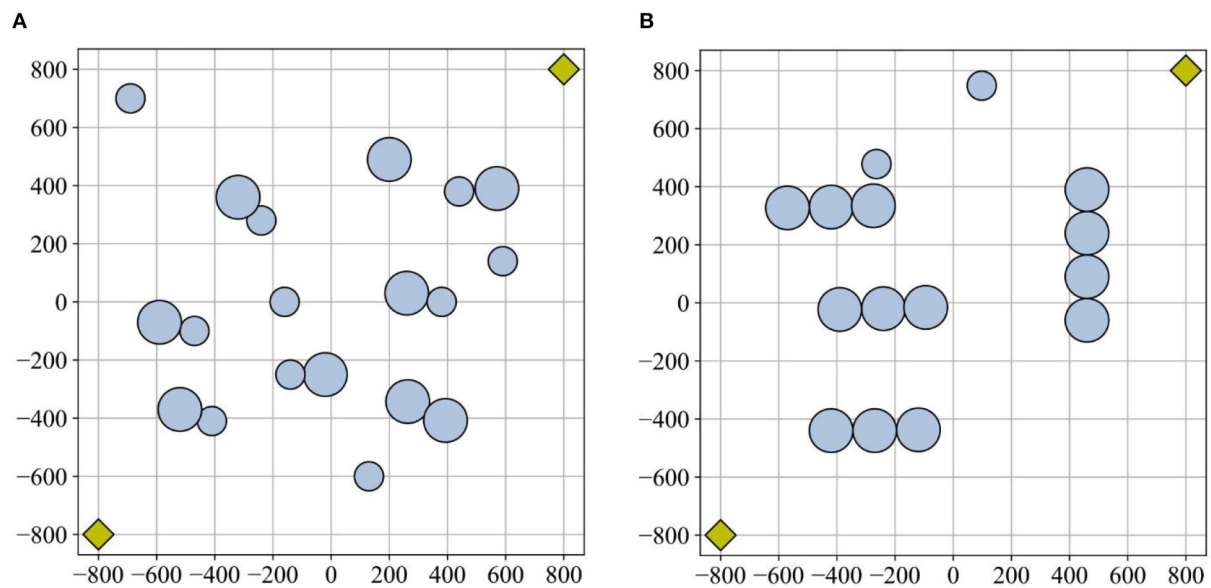


FIGURE 5

Experimental scenarios: (A) scenario 1 and (B) scenario 2. The obstacle area is depicted by a blue circle. The starting point and endpoint coordinates are represented by a yellow diamond.

length for the best solution, the average path length, the standard deviation of the path length, the time required to process the best solution, and the average processing time.

Table 1 provides a performance analysis of these four algorithms. Figure 6 shows detailed results of the four algorithms in 100 experiments conducted in scenario 1. In terms of the path length, from Figure 6A, it can be observed that the ESMA algorithm yields the shortest path, measuring 2,323 cm, with the least overall fluctuation. The SMA-AGDE algorithm, on the other hand, results in the longest optimal path, measuring 2,423 cm. LRSMA exhibits the most significant fluctuations in path length. Table 1 reveals that the standard deviation of path lengths for ESMA is 194, while for LRSMA, it is 276. The ESMA outperformed the SMA, SMA-AGDE, and LRSMA. The evaluation indicators, i.e., minimum path length and average path length, of the ESMA decreased by (4.26, 4.30, and 1.59%) and (0.71, 0.22, and 1.38%), respectively. During the search process, the APF method is employed to guide the algorithm in breaking free from local optimization and directing it toward the desired endpoint. In situations where the SMA becomes trapped in local optimization, the ESMA algorithm addresses this issue by increasing the proportion of the random search population and incorporating the APF technique. From Figure 6B, it can be observed that the ESMA algorithm requires the least amount of time, which is 0.093 in terms of time. From Table 1, it can be observed that the ESMA showed a significant improvement over the SMA, SMA-AGDE, and LRSMA, with the two evaluation indexes of the ESMA (the time to process the best solution and the average time) decreasing by (33.33, 67.74, and 16.13%) and (3.42, 36.75, and 0.00%), respectively, indicating that the ESMA has the fastest convergence speed. The results suggest that ESMA is capable of generating faster, more stable, collision-free, and shorter paths.

Figure 7 provides a detailed breakdown of the results obtained from the 100 experiments conducted in scenario 2, showcasing the performance of the four algorithms. When comparing the path lengths, from Figure 7A, it can be observed that the ESMA algorithm yields the shortest path, measuring 2,465 cm, with relatively minor overall fluctuations compared to LRSMA. The SMA algorithm, on the other hand, results in the longest optimal path, measuring 2,493 cm. SMA-AGDE exhibits the most significant fluctuations in path length. From Table 1, the ESMA algorithm demonstrated significant improvements over the SMA, SMA-AGDE, and LRSMA algorithms. The minimum and average path lengths achieved by ESMA were reduced by approximately (42.7, 18.2, and 2.1%) and (5.3, 2.0, and 1.5%), respectively, compared to the other algorithms. The standard deviation of the path length of the ESMA was larger than that of the LRSMA and smaller than that of the SMA and SMA-AGDE. This phenomenon is due to the fact that the adaptive parameters of the ESMA increase the randomness of the algorithm, while the APF strategy in the ESMA algorithm enhances its global search ability. When comparing the data from Figure 7B, it can be observed that the processing time for obtaining the best solution was comparable between ESMA, SMA, and LRSMA, while SMA-AGDE required a longer processing time. From Table 1, it can be seen that the minimum required time for ESMA, SMA, and LRSMA algorithms is all 0.077, while SMA-AGDE requires a time of 0.172. The average time was slightly better than that of the LRSMA and SMA, and less than that of the SMA-AGDE. However, when the obstacles are distributed in a banded pattern, it becomes necessary to increase the amount of continuous APF guidance, thus reducing the ESMA's time advantage. Overall, the ESMA demonstrated the best comprehensive performance in terms of the

TABLE 1 Simulation performance of the four algorithms.

		Path length (cm)			Planning time (s)		Satisfaction
		Minimum	Mean	Standard deviation	Processing the best solution	Mean	Rate
Scenario 1	SMA	2,422	2,695	221	0.124	0.121	45%
	SMA-AGDE	2,423	2,682	240	0.156	0.160	50%
	LRSMA	2,360	2,713	276	0.108	0.117	65%
	ESMA	2,323	2,676	194	0.093	0.117	98%
Scenario 2	SMA	2,493	2,726	153	0.077	0.118	40%
	SMA-AGDE	2,485	2,943	299	0.172	0.180	56%
	LRSMA	2,492	2,723	139	0.077	0.115	60%
	ESMA	2,465	2,717	142	0.077	0.114	85%

Best results are shown in bold.

minimum and average path lengths, planning time, and stability of the planning results, compared with the SMA, SMA-AGDE, and LRSMA.

Table 1 shows the comparison consequences of the ESMA planning satisfaction rate with that of the other methods. Best results are shown in bold. The satisfaction rate of the other three algorithms did not exceed 70 %, while the ESMA's satisfaction rate in both scenarios was >80%. The satisfaction rate of the ESMA in scenario 1 was higher than that in scenario 2. This shows that the satisfaction rate of the ESMA in scenarios with a punctate obstacle distribution is better than that in scenarios with a banded obstacle distribution. This phenomenon may be due to the higher success rate of the SMA in scenario 1 than in scenario 2. Additionally, the optimization performance of the ESMA in scenario 2 decreased as it is an improvement of the SMA.

The algorithm's sensitivity in a dynamic environment can be assessed by adjusting the obstacle's position and observing the path length before and after the adjustment (Dai et al., 2023). To test the sensitivity of the ESMA, some obstacles in Figure 8 were changed to dynamic obstacles, numbering three in scenario 1 and scenario 2. Therefore, the experiment involved resetting dynamic obstacles along the original dynamic change path. There were two cases: one where the original planned path was not obstructed, and another where it was. As shown in Figure 8, the black route represents the path planned by the ESMA in a static environment, orange circles symbolized the dynamic obstacles in the visualization, and each orange circle is numbered to indicate the sequence of the dynamic obstacles. Table 2 displays all the path length results obtained in all cases. The change rate refers to the relative change in the average length compared to the original path. The average length in scenario 1 is 2,676 cm, while in scenario 2, it is 2,717 cm.

Table 2 presents the sensitivity analysis of the ESMA algorithm to the position of obstacles. In cases where the obstacle did not obstruct the original planned path, the path length remained unchanged. However, when the obstacle hindered the original planned path, the path length was altered. This analysis highlights the dependence of the path length on the positioning of obstacles and demonstrates the impact of obstacle placement on the effectiveness of the ESMA algorithm.

4.3. Robot experiment and analysis

4.3.1. Experimental environment

To further verify the optimization performance of the ESMA, real machine testing was conducted using a robot system. The robot had a length and width of 1.53 and 0.82 m, respectively, and the driving speed was 6–8 km/h, with an Ackerman wire control driving mode. ROS 18.04 was used as the robot's operating system, and the pure pursuit path-tracking method was used to control the robot (Shan et al., 2015). To simplify the calculation, the robot drove at a constant speed of 6 km/h. The test environment used in this study was designed to resemble the simulation environment. It consisted of a blank area where cardboard obstacles were strategically placed.

4.3.2. Robot experiment and analysis

The SMA, SMA-AGDE, LRSMA, and ESMA were run on the robot system to test the 10-run satisfaction. Best results are shown in bold. Table 3 presents various performance metrics for the environment depicted in Figure 5. It includes the minimum path length, average path length, standard deviation of the path length, time required to process the best solution, and average time of path planning. The results indicate that all algorithms were capable of planning collision-free paths in the given environment.

Figure 9A shows the robot tracking the best solutions in scenario 1. Figure 10 shows the detailed path planning lengths and times. From Figure 10A, it can be observed that, in terms of path length, the bars corresponding to the ESMA algorithm have the shortest height, indicating that it finds the shortest path, measuring 2,335 cm. The bar corresponding to the SMA-AGDE algorithm is the tallest, signifying the longest path, measuring 2,489 cm. The bars for ESMA are generally shorter than those for SMA, SMA-AGDE, and LRSMA, implying that only some of ESMA's results are better than those of SMA, SMA-AGDE, and LRSMA. Due to the randomness of the swarm intelligence random algorithm, the planning results show some randomness. Despite the findings mentioned earlier, it is important to note that these specific instances where ESMA did not outperform the other algorithms in terms of path length do not diminish the overall advantages of

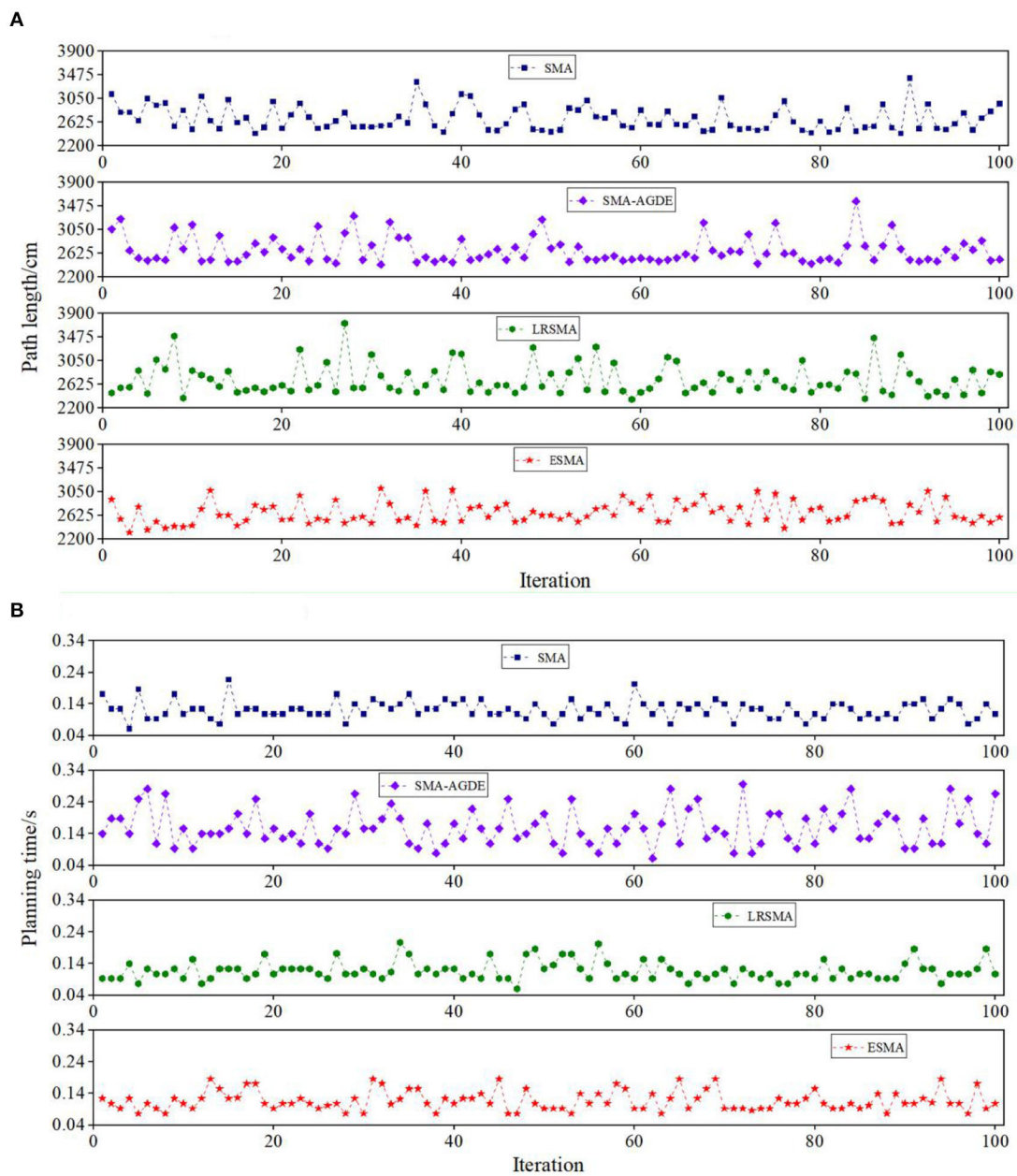


FIGURE 6 Detail simulation performance of the four algorithms in scenario 1: (A) path length and (B) planning time. The Y coordinate of the shape is the path length in (A) and the planning time in (B). The result is the record of 100 runs in each picture.

ESMA in robot path planning. In Figure 10B, it can be seen that the bars corresponding to the ESMA algorithm have the smallest height. As shown in Table 3, compared to the SMA, SMA-AGDE, and LRSMA, the ESMA reduced the minimum path length, average path length, time to process the best solution, and average time by (5.14, 6.60, and 2.57%), (6.69, 1.46, and 3.14%), (16.07, 49.64, and 9.35%), and (0.04, 43.48, and 4.18%), respectively. These results are consistent with the simulation outcomes.

Figure 9B shows the robot tracking the best solutions in scenario 2. Figure 11 shows the detailed path planning lengths and times. From Figure 11A, it can be observed that, in terms of path length, the bar corresponding to the minimum value achieved by

the ESMA algorithm has the smallest height, indicating the shortest path, measuring 2,465 cm. It is relatively close to the minimum value achieved by LRSMA. The bar corresponding to the SMA algorithm is the tallest, signifying the longest path, measuring 2,535 cm. From Figure 11B, it can be seen that ESMA's processing of the best solution is close to LRSMA but outperforms SMA and SMA-AGDE. As depicted in Table 3, compared to the SMA, SMA-AGDE, and LRSMA, the ESMA reduced the minimum path length, average path length, time to process the best solution, and average time by (2.84, 1.58, and 0.32%), (3.92, 8.93, and 2.73%), (73.08, 125.64, and 1.28%), and (27.18, 68.93, and 13.59%), respectively. The results obtained in the physical test environment surpass those

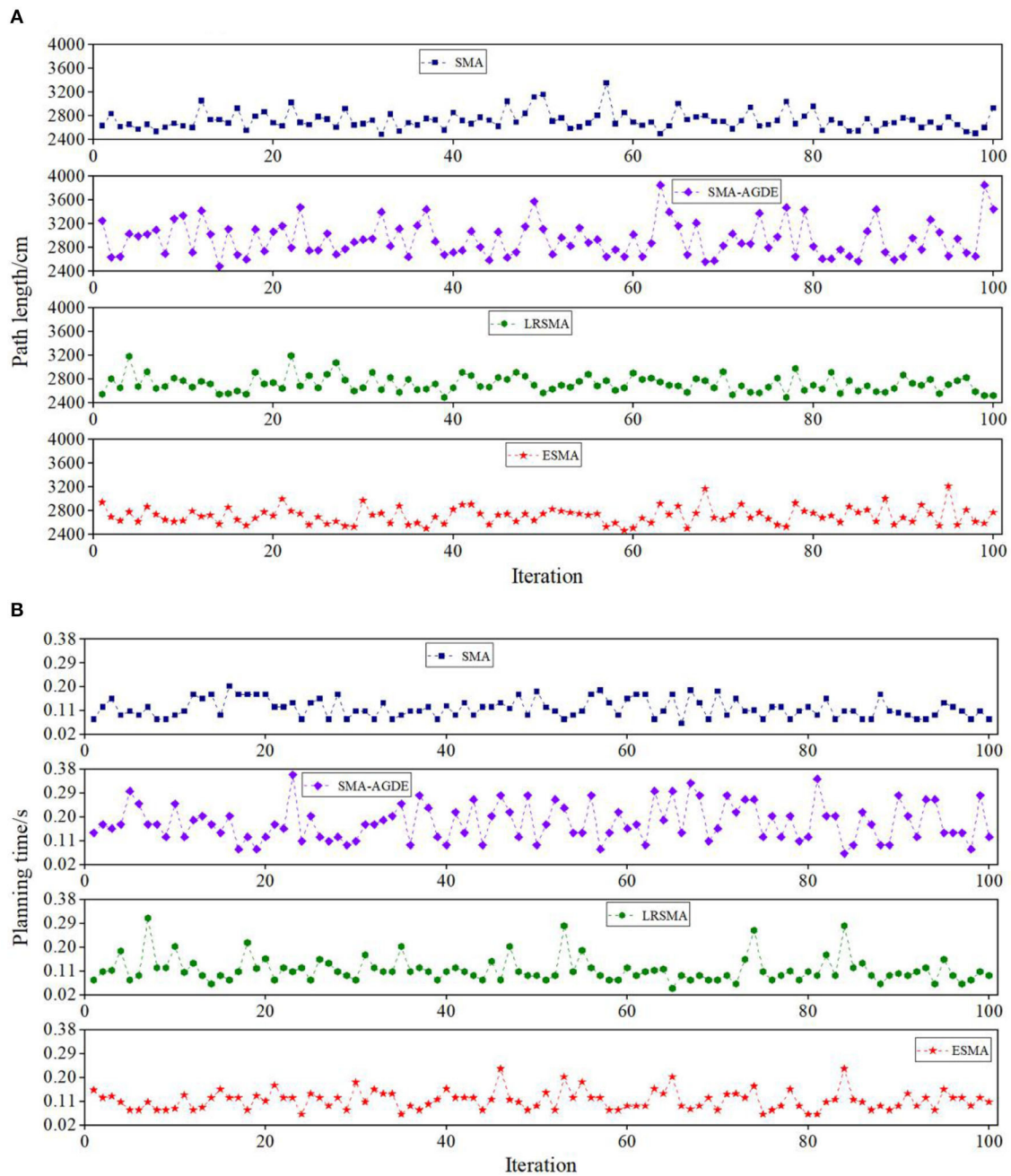


FIGURE 7
Detail simulation performance of the four algorithms in scenario 2: (A) path length and (B) planning time. The Y coordinate of the shape is the path length in (A) and the planning time in (B). The result is the record of 100 runs in each picture.

achieved in the simulation environment, and this discrepancy could be attributed to the limited number of experiments conducted.

Figure 12 compares the convergence speed of the four algorithms at the minimum path length. The fitness value is the sum of distances from the current position to the previous position, and from the current position to the endpoint. The ESMA shows slow convergence because the APF guides the global optimal point to the endpoint with a fixed step size. Fifty iterations in all scenarios are sufficient to achieve good performance.

5. Conclusions

This paper introduces the ESMA, a novel swarm intelligence algorithm addressing the path-planning problem in dynamic environments for autonomous mobile robots. The ESMA algorithm incorporates adaptive technology and an artificial potential field to improve the convergence speed while overcoming local optimization issues. In comparison to the SMA, SMA-AGDE, and LRSMA algorithms, ESMA achieved the smallest average minimum path length and minimum path values, along with the

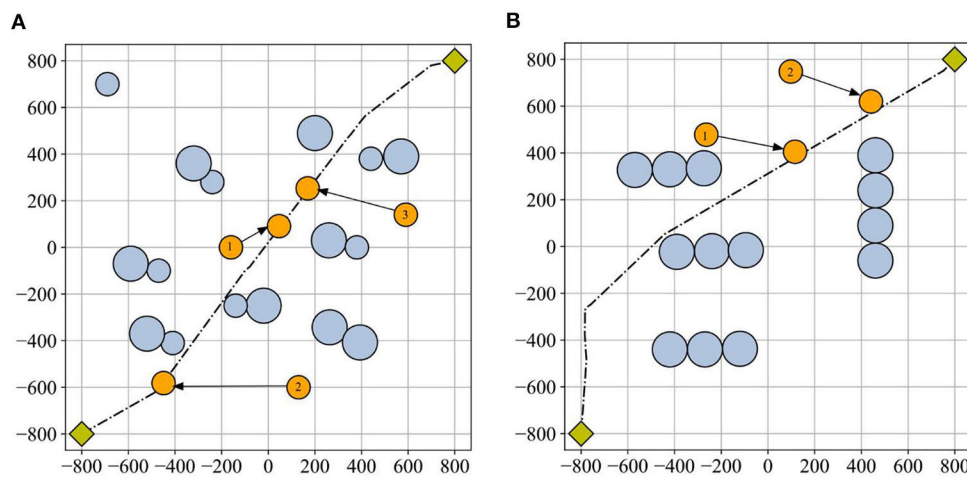


FIGURE 8 Dynamic obstacles in scenarios: (A) scenario 1 and (B) scenario 2. The black route represents the path planned by the ESMA in a static environment, orange circles symbolized the dynamic obstacles in the visualization, and each orange circle is numbered to indicate the sequence of the dynamic obstacles. The dynamic obstacles are reset along the original dynamic change path.

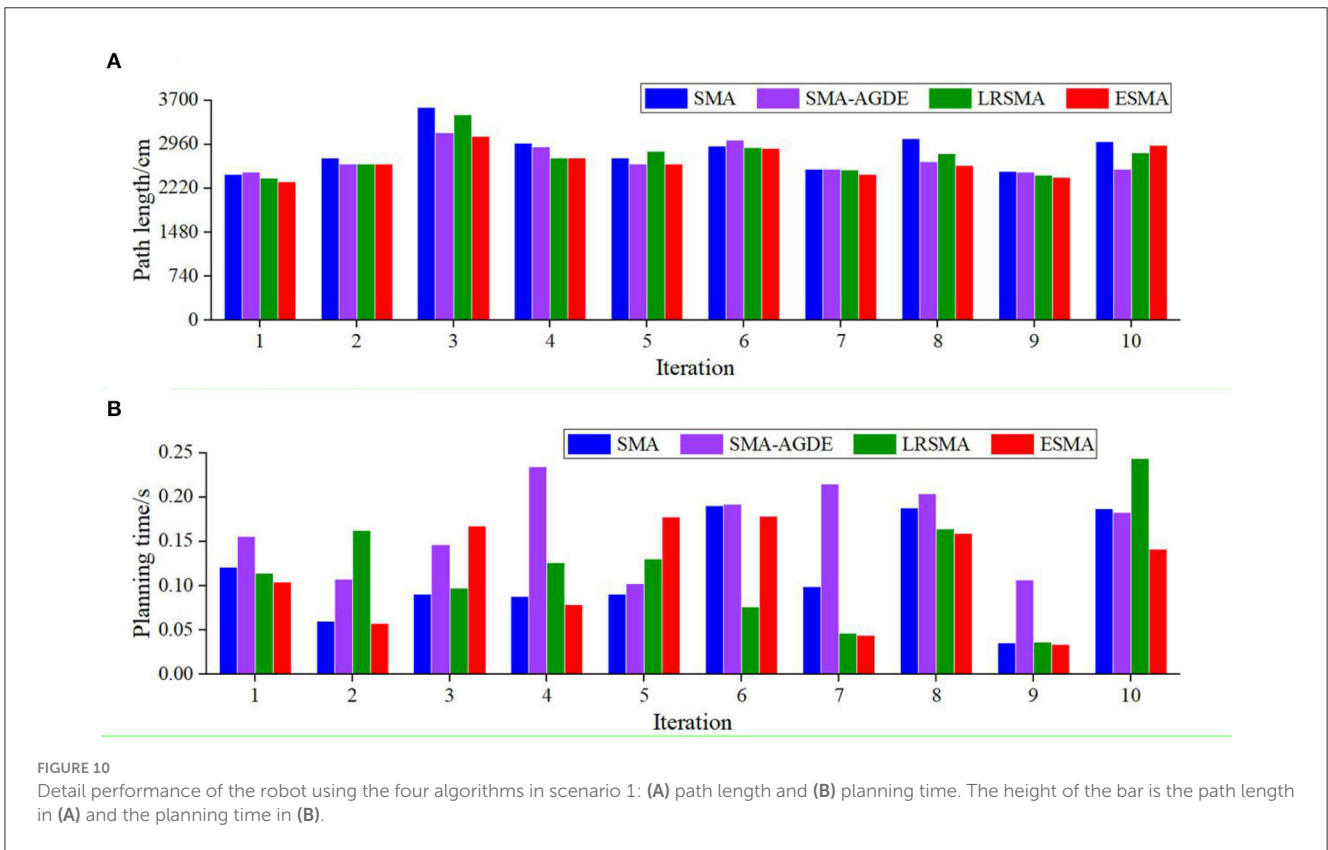
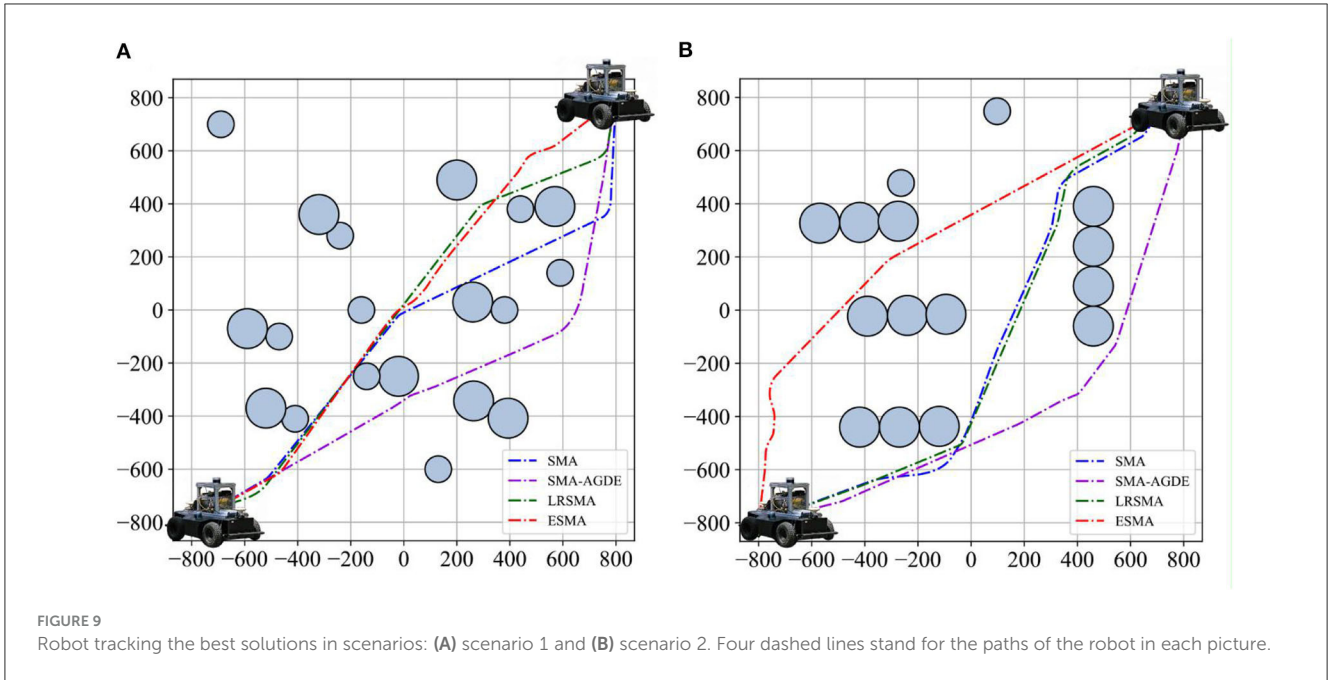
TABLE 2 Comparison results of planned path lengths.

	Obstacle number	Hindered original path (cm)	Rate of change	Unblocked original path (cm)	Rate of change
Scenario 1	1	2,623	2.00	2,669	0.28
	2	2,753	2.86	2,664	0.47
	3	2,728	1.93	2,661	0.61
Scenario 2	1	2,785	2.51	2,714	0.11
	2	2,788	2.62	2,711	0.21

TABLE 3 Performance of the robot using the SMA, SMA-AGDE, LRSMA, and ESMA.

		Path length (cm)			Planning time (s)	
		Minimum	Mean	Standard deviation	Processing the best solution	Mean
Scenario 1	SMA	2,455	2,853	336	0.121	0.115
	SMA-AGDE	2,489	2,713	239	0.156	0.165
	LRSMA	2,395	2,758	302	0.114	0.120
	ESMA	2,335	2,674	248	0.104	0.115
Scenario 2	SMA	2,535	2,782	241	0.135	0.131
	SMA-AGDE	2,504	2,916	298	0.176	0.174
	LRSMA	2,473	2,750	227	0.079	0.117
	ESMA	2,465	2,677	133	0.078	0.103

Best results are shown in bold.



shortest algorithm processing time, in experiments conducted in both simulation and real-world scenarios. Consequently, it demonstrated that ESMA is capable of generating shorter collision-free paths with greater accuracy and stability compared to the other solutions. However, on one hand, determining optimal parameters for the Artificial Potential Field (APF) is a challenge,

and automated parameter tuning is a topic for future research. On the other hand, enhancing the practical planning stability of ESMA remains an area for further improvement. Future work will address these issues and also involve a comparison of ESMA with other swarm intelligence algorithms in various dynamic environments.

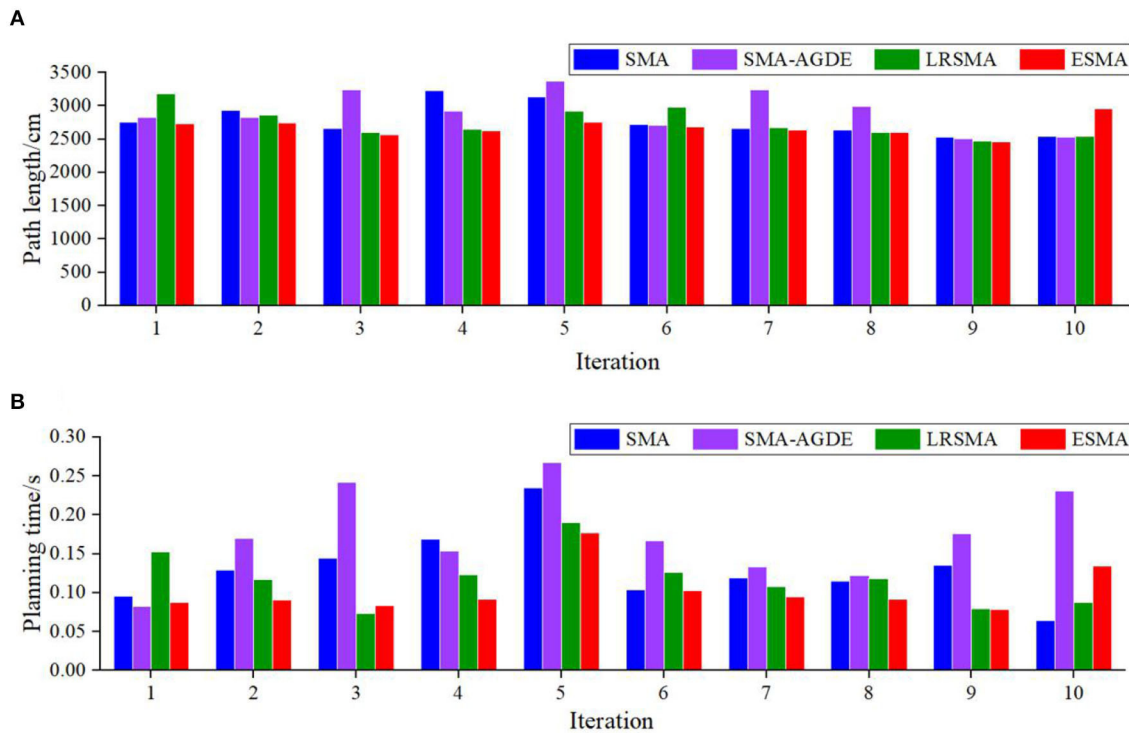


FIGURE 11 Detailed performance of the robot using the four algorithms in scenario 2: (A) path length and (B) planning time. The height of the bar is the path length in (A) and the planning time in (B).

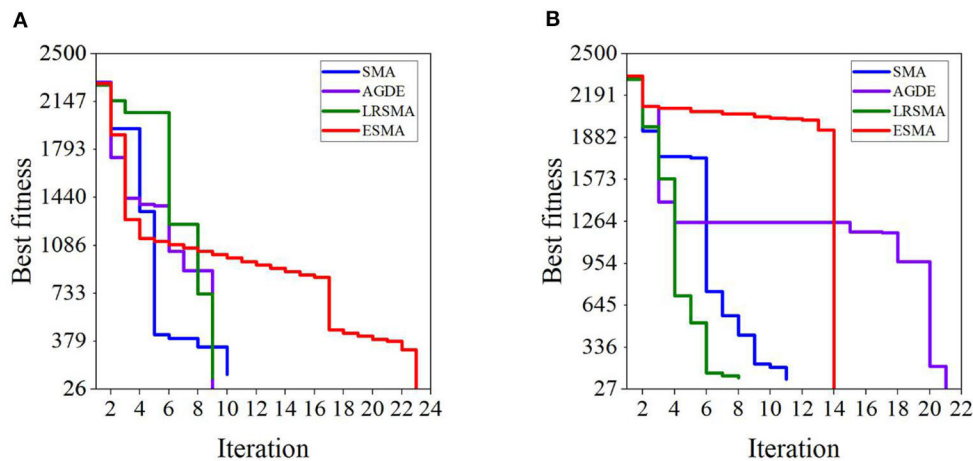


FIGURE 12 Iteration curves of the four algorithms in scenarios: (A) scenario 1 and (B) scenario 2. The results are the best fitness values of the four algorithms at the minimum path length. The fitness value is the sum of distances from the current position to the previous position, and from the current position to the endpoint.

Practically, for ground environments, the ESMA can aid automatic robots or slow automated guided vehicles (AGVs) path planning. For robots, it enables food delivery, express deliveries or automatic sales. For AGVs, it applies to port cargo handling, airport baggage transfer and other relevant applications. However,

ESMA may be unsuitable for fast-motion scenarios like relief supplies transportation. Additionally, one of the future research directions is to extend the algorithm’s practical application to other environments, such as path planning for unmanned boats, drones, and similar contexts.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

LZ: Conceptualization, Funding acquisition, Validation, Writing—original draft. CH: Writing—review and editing. HS: Validation, Writing—original draft. RC: Validation, Writing—original draft.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by China Postdoctoral Science Foundation 2021M691114, the Central Funds Guiding the Local Science

References

- Agarwal, D., and Bharti, P. S. (2021). Implementing modified swarm intelligence algorithm based on Slime moulds for path planning and obstacle avoidance problem in mobile robots. *Appl. Soft Comput.* 107, 15. doi: 10.1016/j.asoc.2021.107372
- Al Hilli, A., Al-Ibadi, M., Alfidhel, A. M., Abdulshaheed, S. H., and Hadi, A. H. (2021). Optimal path finding in stochastic quasi-dynamic environments using particle swarm optimization. *Expert Syst. With Appl.* 186, 9. doi: 10.1016/j.eswa.2021.115706
- Alyasseri, Z. A. A., Alomari, O. A., Al-Betar, M. A., Makhadmeh, S. N., Doush, I. A., Awadallah, M. A., et al. (2022). Recent advances of bat-inspired algorithm, its versions and applications. *Neural Comp. Appl.* 34, 16387–16422. doi: 10.1007/s00521-022-07662-y
- Bao, S. J., Ebadi, A., Toughani, M., Dalle, J., Maselena, A., Baharuddin, et al. (2020). A new method for optimal parameters identification of a PEMFC using an improved version of Monarch Butterfly Optimization Algorithm. *Int. J. Hydr. Energy* 45, 17882–17892. doi: 10.1016/j.ijhydene.2020.04.256
- Cai, Z. Y., Xiong, Z. P., Wan, K. P., Xu, Y. Q., and Xu, F. (2020). A node selecting approach for traffic network based on artificial slime mold. *IEEE Access* 8, 8436–8448. doi: 10.1109/ACCESS.2020.2964002
- Cheng, K. P., Elara, M. R., Nhan, N. H. K., and Le, A. V. (2020). Multi-objective genetic algorithm-based autonomous path planning for hinged-tetro reconfigurable tiling robot. *IEEE Access* 8, 121267–121284. doi: 10.1109/ACCESS.2020.3006579
- Dai, Y. N., Yu, J. Y., Zhang, C., Zhan, B. W., and Zheng, X. T. (2023). A novel whale optimization algorithm of path planning strategy for mobile robots. *Appl. Intell.* 53, 10843–10857. doi: 10.1007/s10489-022-04030-0
- Das, M. S., Sanyal, S., and Mandal, S. (2022). Navigation of multiple robots in formative manner in an unknown environment using artificial potential field based path planning algorithm. *Ain Shams Eng. J.* 13, 15. doi: 10.1016/j.asej.2021.101675
- Dereli, S. (2022). A novel approach based on average swarm intelligence to improve the whale optimization algorithm. *Arab. J. Sci. Eng.* 47, 1763–1776. doi: 10.1007/s13369-021-06042-3
- Fernandes, P. B., Oliveira, R. C. L., and Neto, J. V. F. (2022). Trajectory planning of autonomous mobile robots applying a particle swarm optimization algorithm with peaks of diversity. *Appl. Soft Comp.* 116, 17. doi: 10.1016/j.asoc.2021.108108
- Fragapane, G., de Koster, R., Sgarbossa, F., and Strandhagen, J. O. (2021). Planning and control of autonomous mobile robots for intralogistics: literature review and research agenda. *Eur. J. Operat. Res.* 294, 405–426. doi: 10.1016/j.ejor.2021.01.019
- Hewawasam, H. S., Ibrahim, M. Y., and Appuhamillage, G. K. (2022). Past, present and future of path-planning algorithms for mobile robot navigation in dynamic environments. *IEEE Open J. Indust. Elect. Soc.* 3, 353–365. doi: 10.1109/OJIES.2022.3179617
- and Technology Development 2021Szvup045 and Fundamental Research Funds for the Central Universities 2042022kf0049.
- ## Conflict of interest
- HS was employed by the China Mobile Intelligent Mobility Network Technology Co., Ltd.
- The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
- ## Publisher's note
- All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.
- Houssein, E. H., Helmy, B. E.-d., Rezk, H., and Nassef, A. M. (2022). An efficient orthogonal opposition-based learning slime mould algorithm for maximum power point tracking. *Neural Comp. Appl.* 34, 3671–3695. doi: 10.1007/s00521-021-06634-y
- Houssein, E. H., Mahdy, M. A., Blondin, M. J., Shebl, D., and Mohamed, W. M. (2021). Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems. *Expert Syst. With Appl.* 174, 114689. doi: 10.1016/j.eswa.2021.114689
- Jin, Y., Yue, M., Li, W. J., and Shangguan, J. Y. (2022). An improved target-oriented path planning algorithm for wheeled mobile robots. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* 236, 11081–11093. doi: 10.1177/09544062221111695
- Kavraki, L. E., Svestka, P., Latombe, J. C., and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Rob. Autom.* 12, 566–580. doi: 10.1109/70.508439
- Li, S. M., Chen, H. L., Wang, M. J., Heidari, A. A., and Mirjalili, S. (2020). Slime mould algorithm: a new method for stochastic optimization. *Fut. Gen. Comp. Syst. Int. J. Esci.* 111, 300–323. doi: 10.1016/j.future.2020.03.055
- Liang, J. H., and Lee, C. H. (2015). Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm. *Adv. Eng. Software* 79, 47–56. doi: 10.1016/j.advengsoft.2014.09.006
- Lin, L., and Gen, M. (2009). Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation. *Soft Comp.* 13, 157–168. doi: 10.1007/s00500-008-0303-2
- Liu, B. Y., Ye, X. B., Wang, X. B., Jia, J., and Wang, T. (2020). Path avoidance algorithm for unmanned ground vehicles based on improved artificial potential field. *J. Chin. Inert. Technol.* 28, 769–777. doi: 10.13695/j.cnki.12-1222/o3.2020.06.011
- Liu, J. X., Shi, J. F., Hao, F., and Dai, M. (2022). A novel enhanced global exploration whale optimization algorithm based on Levy flights and judgment mechanism for global continuous optimization problems. *Eng. With Comput.* 29, 1–29. doi: 10.1007/s00366-022-01638-1
- Liu, Y., and Liu, S. (2022). Unscented sigma point guided quasi-opposite slime mould algorithm and its application in engineering problem. *Appl. Res. Comp.* 39, 2709–2716. doi: 10.19734/j.issn.1001-3695.2022.02.0055
- Liu, Z. X., Yuan, X. F., Huang, G. M., Tan, W. H., and Wang, Y. N. (2022). 3D gradient reconstruction-based path planning method for autonomous vehicle with enhanced roll stability. *IEEE Trans. Intell. Transp. Syst.* 23, 20563–20571. doi: 10.1109/ITITS.2022.3177608
- Mac, T. T., Copot, C., Tran, D. T., and De Keyser, R. (2016). Heuristic approaches in robot path planning: a survey. *Rob. Auton. Syst.* 86, 13–28. doi: 10.1016/j.robot.2016.08.001
- Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey wolf optimizer. *Adv. Eng. Soft.* 69, 46–61. doi: 10.1016/j.advengsoft.2013.12.007

- Mohamed, A. W., and Mohamed, A. K. (2019). Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. *Int. J. Mach. Learn. Cybern.* 10, 253–277. doi: 10.1007/s13042-017-0711-7
- Mohanty, P. K. (2020). An intelligent navigational strategy for mobile robots in uncertain environments using smart cuckoo search algorithm. *J. Amb. Intellig. Human. Comput.* 11, 6387–6402. doi: 10.1007/s12652-020-02535-5
- Na, X., Wang, J., Han, M., and Li, D. (2022). Gradient eigendecomposition invariance biogeography-based optimization for mobile robot path planning. *Soft Comput.* 26, 6131–6144. doi: 10.1007/s00500-022-07075-y
- Nguyen, T.-T., Wang, H.-J., Dao, T.-K., Pan, J.-S., Liu, J.-H., and Weng, S. (2020). An improved slime mold algorithm and its application for optimal operation of cascade hydropower stations. *IEEE Access* 8, 226754–226772. doi: 10.1109/ACCESS.2020.3045975
- Pang, B., Song, Y., Zhang, C. J., Wang, H. L., and Yang, R. T. (2019). Bacterial foraging optimization based on improved chemotaxis process and novel swarming strategy. *Appl. Intellig.* 49, 1283–1305. doi: 10.1007/s10489-018-1317-9
- Rafai, A. N. A., Adzhar, N., and Jaini, N. I. (2022). A review on path planning and obstacle avoidance algorithms for autonomous mobile robots. *J. Rob.* 2022, 14. doi: 10.1155/2022/2538220
- Rizk-Allah, R. M., Hassanien, A. E., and Song, D. (2022). Chaos-opposition-enhanced slime mould algorithm for minimizing the cost of energy for the wind turbines on high-altitude sites. *ISA Trans.* 121, 191–205. doi: 10.1016/j.isatra.2021.04.011
- Rodriguez-Molina, A., Mezura-Montes, E., Villarreal-Cervantes, M. G., and Aldape-Perez, M. (2020). Multi-objective meta-heuristic optimization in intelligent control: a survey on the controller tuning problem. *Appl. Soft Comput.* 93, 26. doi: 10.1016/j.asoc.2020.106342
- Sanchez-Ibanez, J. R., Perez-del-Pulgar, C. J., and Garcia-Cerezo, A. (2021). Path planning for autonomous mobile robots: a review. *Sensors* 21, 29. doi: 10.3390/s21237898
- Saraswathi, M.B.L., Murali, G.B., and Deepak, B. (2018). “Optimal path planning of mobile robot using hybrid cuckoo search-bat algorithm”, in *1st International Conference on Robotics and Smart Manufacturing (RoSma)*, eds S. Muthuswamy and M. Zoppi (Amsterdam: Elsevier Science Bv), 510–517.
- Shan, Y. X., Yang, W., Chen, C., Zhou, J., Zheng, L., and Li, B. J. (2015). CF-pursuit: a pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles. *Int. J. Adv. Rob. Syst.* 12, 13. doi: 10.5772/61391
- Teng, Z. J., Lv, J. L., and Guo, L. W. (2019). An improved hybrid grey wolf optimization algorithm. *Soft Comput.* 23, 6617–6631. doi: 10.1007/s00500-018-3310-y
- Versaci, M., Angiulli, G., di Barba, P., and Morabito, F. C. (2020). Joint use of eddy current imaging and fuzzy similarities to assess the integrity of steel plates. *Open Phys.* 18, 230–240. doi: 10.1515/phys-2020-0159
- Wang, B. F., Li, S., Guo, J., and Chen, Q. W. (2018). Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm. *Neurocomputing* 282, 42–51. doi: 10.1016/j.neucom.2017.12.015
- Xiong, N., Zhou, X. Z., Yang, X. Q., Xiang, Y., and Ma, J. Y. (2021). Mobile robot path planning based on time taboo ant colony optimization in dynamic environment. *Front. Neurobot.* 15, 11. doi: 10.3389/fnbot.2021.642733
- Xu, G. H., Zhang, T. W., Lai, Q., Pan, J., Fu, B., and Zhao, X. L. (2020). A new path planning method of mobile robot based on adaptive dynamic firefly algorithm. *Mod. Phys. Lett. B* 34, 17. doi: 10.1142/S0217984920503224
- Yu, C., Heidari, A. A., Xue, X., Zhang, L., Chen, H., and Chen, W. (2021). Boosting quantum rotation gate embedded slime mould algorithm. *Exp. Syst. With Appl.* 181, 115082. doi: 10.1016/j.eswa.2021.115082
- Zhang, Z. D., Qin, H., Yao, L. Q., Liu, Y. Q., Jiang, Z. Q., Feng, Z. K., et al. (2020). Improved multi-objective moth-flame optimization algorithm based on R-domination for cascade reservoirs operation. *J. Hydrol.* 581, 21. doi: 10.1016/j.jhydrol.2019.124431
- Zhang, Z. H., Qiao, B., Zhao, W. T., and Chen, X. (2021). A predictive path planning algorithm for mobile robot in dynamic environments based on rapidly exploring random tree. *Arab. J. Sci. Eng.* 46, 8223–8232. doi: 10.1007/s13369-021-05443-8
- Zheng, L., Tian, Y., Wang, H., Hong, C., and Li, B. (2023). Path planning of autonomous mobile robots based on an improved slime mould algorithm. *Drones* 7, 257. doi: 10.3390/drones7040257