

Research Article

An Improved Simulated Annealing Particle Swarm Optimization Algorithm for Path Planning of Mobile Robots Using Mutation Particles

Jianzhang Lu  and Zhihao Zhang 

Chongqing University-University of Cincinnati Joint Co-op Institute, Chongqing University, Chongqing 400000, China

Correspondence should be addressed to Zhihao Zhang; 20186101@cqu.edu.cn

Received 30 September 2021; Revised 3 November 2021; Accepted 6 November 2021; Published 6 December 2021

Academic Editor: Deepak Gupta

Copyright © 2021 Jianzhang Lu and Zhihao Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Artificial intelligence technology has brought tremendous changes to human life and production methods. Mobile robots, UAVs, and autonomous driving technology have gradually entered people's daily life. As a typical issue for a mobile robot, the planning of an optimal mobile path is very important, especially in the military and emergency rescue. In order to ensure the efficiency of operation and the accuracy of the path, it is crucial for the robot to find the optimal path quickly and accurately. This paper discusses a new method and MP-SAPSO algorithm for addressing the issue of path planning based on the PSO algorithm by combining particle swarm optimization (PSO) algorithm with the simulated annealing (SA) algorithm and mutation particle and adjusting the parameters. The MP-SAPSO algorithm improves the accuracy of path planning and the efficiency of robot operation. The experiment also demonstrates that the MP-SAPSO algorithm can be used to effectively address path planning issue of mobile robots.

1. Introduction

1.1. The Purpose and Significance of the Research. As modular and intelligent thinking gradually penetrates into business and traditional manufacturing, the application of cutting-edge technologies has become much more efficient, and a large number of digital and intelligent high-end technology products have entered human daily life with lower cost and higher quality. The mobile robot, the object of this thesis, is a microcosm of this change. Path planning is the key to the autonomous control module of a robot. The path planning problem is an NP-hard problem, which allows the robot to automatically calculate the best path from the start point to the end point in minimum time [1]. To solve the path planning problem, two issues need to be tackled. One is to build an environment model that can simulate the robot's working scenario, and the other is to establish the rules for the robot to search the path in the environment model and find the optimal path for the robot in the envi-

ronment model. In this way, the path planning problem for robots is transformed into a combinatorial optimization problem. The common path planning methods include swarm intelligence algorithm, graph search algorithm, RRT algorithm, Artificial Potential Field algorithm, BUG obstacle avoidance algorithm, and incremental heuristic algorithm. Swarm intelligence algorithm is a popular category in current research, and the more mature algorithms include particle swarm optimization algorithm and ant colony algorithm, although the particle swarm optimization algorithm is relatively simple in implementation. However, compared to other algorithms, there are problems such as premature convergence and more likely to fall into local optimal solutions. To address these shortcomings [2], this paper combines particle swarm optimization algorithm and simulated annealing algorithm, introduces the concept of mutation factor, and proposes a new robot path planning scheme for path planning under a known environment model. Compared with the traditional classical particle swarm

optimization algorithm, this algorithm can effectively avoid premature convergence, improve the efficiency of the algorithm, and provide a new idea for robot path planning.

1.2. Mobile Robot Path Planning Problem. According to the different perception degrees of robot to its own environment model, common path planning methods are divided into global path planning with a known environment model and local path planning methods using sensor, SLAM radar, and other technologies. This paper mainly studies the global path planning algorithm under the known environment model. Similar to the research of Katrakazas et al. [3] on path planning for mobile robot, the global path planning for robots should include the following problems:

- (1) How to implement path planning in actual robot tasks

If the path planning strategy violates the hardware structure and dynamics of the robot, the path planning strategy is inaccurate. This can be achieved through personalized modeling of the environment model and personalized adjustment of path planning strategy for different robots.

- (2) How to obtain the optimal solution of path planning

There are many paths from the start point to the end point, but due to the efficiency and power of the robot, it is necessary to find a feasible optimal path.

- (3) How to avoid collision between the robot and obstacles on the planned path

The robot may deal with a variety of situations when performing tasks. Collision with obstacles in the environment will have a certain impact on the robot and surrounding obstacles. Collision will also adversely affect the execution path of the robot, thereby reducing work efficiency.

1.3. Development Trend of Robot Path Planning Algorithm. Some works are focused on the path planning to solve the above problems. Figure 1 shows some of the algorithms used in path planning.

The key to building an environment model is to make the state of robot in the environment model consistent with the real environment; the path planning algorithm focuses on optimizing the robot path based on the known environment model. For the second point, robot path planning algorithm has attracted the attention of many experts and scholars. There are also many solutions and achievements in this field. Roberge et al. combine genetic algorithm and particle swarm optimization algorithm to solve the path planning problem of UAV in a complex three-dimensional environment [4]. Mo and Xu proposed a global path planning method, which combines BBO, PSO, and approximate Voronoi boundary network (AVBN) in a static environment [5]. Yu and Rus proposed a centralized algorithm framework to solve the multirobot path planning problem in a general two-dimensional continuous environment [6]. Li et al. developed an effective Improved Artificial Potential Field-

based Simultaneous Forward Search (Improved APF-based SIFORS) method in a known environment [7]. Jiao et al. performed an improved quantum particle swarm optimization algorithm to solve the path planning model [8]. Tharwat et al. proposed a new chaos particle swarm optimization algorithm to optimize the control points of Bezier curve [9]. Li et al. proposed a new adaptive learning mechanism to select the most suitable search strategy adaptively in different stages of the optimization process, which improves the searchability of particle swarm optimization algorithm [7]. Das et al. performed an improved particle swarm optimization (IPSO) algorithm combined with differential disturbance velocity (DV) algorithm to determine the optimal path of multirobot in a clutter environment [10]. Zeng et al. aimed to propose a particle swarm optimization (PSO) algorithm based on nonhomogeneous Markov chain and differential evolution for path planning of intelligent robot when encountering obstacles in the environment [11].

We can see particle swarm optimization has not only strong local searchability but also global exploration ability. Moreover, the proportion of local search and global search can be controlled by adjusting the parameters. Therefore, it is suitable to be used as a path optimization algorithm for mobile robot.

2. Establishment and Optimization of the Model

2.1. Establishment of Environmental Model. The application problem discussed in this paper needs to simulate the environment with obstacles and give a visual expression of the improved algorithm, so a direct and specific environment model is needed. For the path planning algorithm, the solution is only limited to the two-dimensional plane space of the robot movement, so a two-dimensional space model is established.

2.1.1. Polygon Obstacle Environment Model. When the environment model is a two-dimensional space, a polygonal obstacle environment is the clearest and most direct representation of obstacles [12]. The image displayed by this environment model can be regarded as the top view of the two-dimensional obstacle environment in a three-dimensional space (as shown in Figure 2). It can not only clearly display the state of the obstacle environment and facilitate the adjustment of the obstacle environment but also prepare for the optimization path of the visualization algorithm.

2.1.2. Improvement of Polygon Obstacle Environment Model. In the actual process of robot movement, due to the limitation of its own mechanical structure, the robot cannot fully adapt to the movement of the edges and corners of obstacles. Therefore, considering the protection of the robot itself and the needs of practical application, the polygon obstacles were regarded as a noncontact space containing multiangle obstacles in the environment model, and it is stated that as long as the mobile body does not enter the space, it is regarded as effective movement and can be realized in the path; The noncontact space is built following the principle of



FIGURE 1: The development of path planning algorithm.

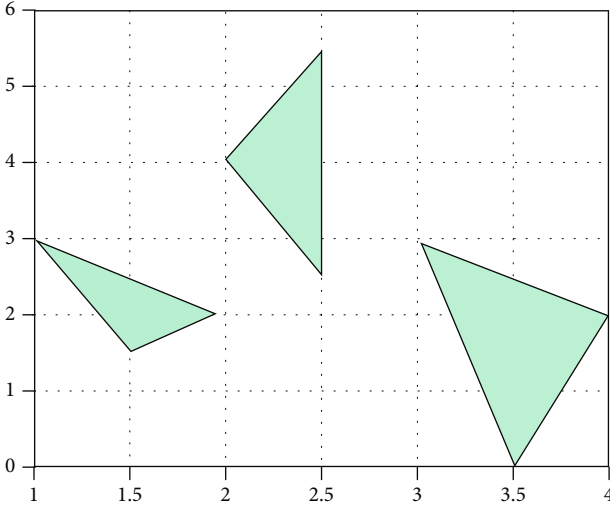


FIGURE 2: Polygon obstacle environment.

minimum enclosing circle, and the space of the obstacle is defined as the minimum circular space that can completely cover the obstacle. To find the minimum enclosing circle, we need to get the vertices of the obstacle and form the point set P . By iterating in P , we can find three vertices, so as to determine a minimum circle that completely covers all the vertices of the obstacle. Let these three points be $p_1(x_1, y_1)$, $p_2(x_2, y_2)$, and $p_3(x_3, y_3)$. Then, the midpoint of the line l_1 with $p_1(x_1, y_1), p_2(x_2, y_2)$ is

$$(x_{\text{mid}}, y_{\text{mid}}) = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right). \quad (1)$$

The equation of the vertical line l_1 is $A_1x + B_1y = C_1$, where

$$\begin{aligned} A_1 &= -B = x_2 - x_1, \\ B_1 &= A = y_2 - y_1, \end{aligned} \quad (2)$$

$$C_1 = -B * x_{\text{mid}} + A * y_{\text{mid}} = \frac{(x_2^2 - x_1^2) + (y_2^2 - y_1^2)}{2}.$$

Two midperpendicular lines are obtained using the above-mentioned method, and their intersection point is the center of the minimum enclosing circle, and the noncontact space that completely covers the obstacle can be obtained, as shown in Figure 3.

If the result of the robot entering the obstacle space occurs in the loop iteration of the algorithm, the violation is output and marked with “*” as shown in Figure 4.

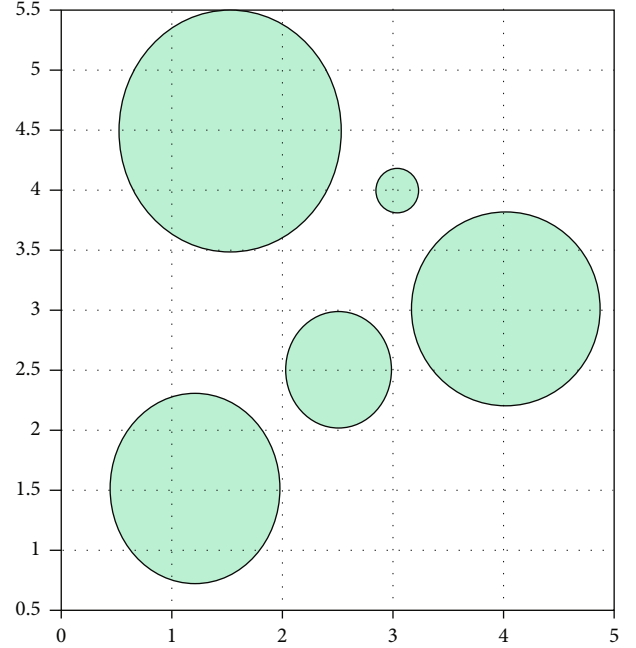


FIGURE 3: Improvement of multilateral obstacle model.

2.2. Basic Particle Swarm Optimization Algorithm

2.2.1. Overview of Particle Swarm Optimization Algorithm. Particle swarm optimization (PSO, hereinafter referred to as “PSO”) is a kind of evolutionary computing technology, which affects the simple behavior of individuals through the sharing of information among individuals in the group. It simulates the evolution process of the group from disorder to order in the space via iteration and obtains the optimal solution. By controlling the inertia parameter, acceleration parameter, and fitness algorithm, the group behavior can be influenced and converged to the desired direction. On the premise that the computing platform supports large-scale parallel computing, particle swarm optimization has higher efficiency and faster computing speed. Compared with other evolutionary algorithms, particle swarm optimization has the advantages of high accuracy, fewer iterations, and rapid convergence. It is a mature and feasible solution to apply particle swarm optimization algorithm to the path planning of robots.

2.2.2. Equations of the Basic Particle Swarm Optimization Algorithm. The core of particle swarm optimization algorithm lies in two equations velocity update equation and position update equation. In each iteration, each particle updates its position and velocity according to the velocity update equation and the position update equation and

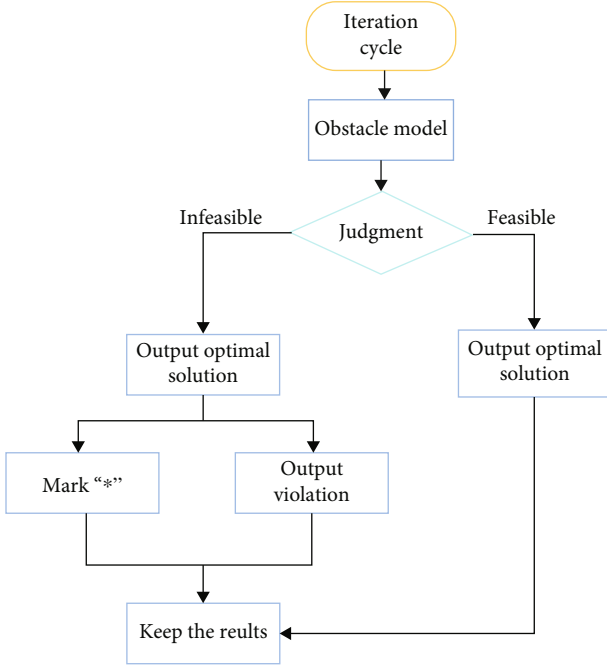


FIGURE 4: Flow chart of space violation judgment.

updates the individual optimal solution and the global optimal solution. Through several iterations, the particle swarm in the solution space converges to the optimal solution and finally converges to the optimal solution.

Velocity update equation

$$v_i = \omega \times v_i + c_1 \times \text{rand}() \times (\text{pbest}_i - x_i) + c_2 \times \text{rand}() \times (\text{gbest}_i - x_i). \quad (3)$$

Location update equation

$$x_i = x_i + v_i. \quad (4)$$

In the equations, v_i is the velocity of the particles, ω is the weighted inertia, and x_i is the location for the particles. pbest_i is the local optimal solution. gbest_i is the global optimal solution. $\text{rand}()$ is a random number between 0 and 1, and c_1 and c_2 are learning factors. Usually, $c_1 = c_2 = 2$.

2.2.3. Defects of Basic Particle Swarm Optimization Algorithm. The basic particle swarm optimization algorithm has the advantages of fast convergence speed, few adjustment parameters, and simple implementation. The speed displacement model and the small amount of calculation have attracted the attention of many researchers. However, with more and more research, researchers have gradually found the defects of particle swarm optimization algorithm. The defects of particle swarm optimization are mainly manifested in two aspects: (1) the accuracy of particle swarm optimization is not high. Since the PSO only uses the global optimal solution and local optimal solution in each iteration, it does not use all the information of each iteration. Therefore, even if the convergence speed of PSO is fast, it is diffi-

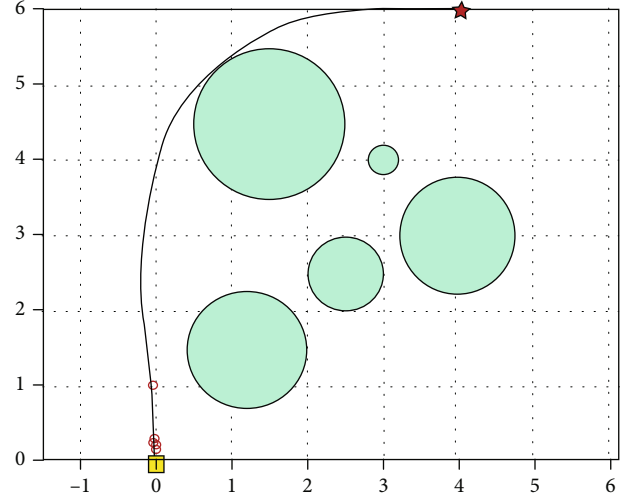


FIGURE 5: PSO path planning result.

cult to guarantee the accuracy of the algorithm. (2) Local optimization, premature convergence, and the inability to find the global optimal solution are likely to occur in the search process of particle swarm optimization. This may be because the particle swarm optimization algorithm cannot get a complete proof of convergence in theory [13]. For more complex multimodel function optimization, the global optimal solution may not be obtained; in addition, if the parameter selection is wrong, particles quickly lose diversity, and the global optimal solution cannot be obtained.

In order to solve these problems, Zhou et al. proposed an improved adaptive PSO algorithm, where the Bezier curve was incorporated and larger control parameters were used to improve the smoothness and efficiency of path planning [14]. The adaptive particle swarm optimization algorithm based on simulated annealing (SAPSO) proposed by Yan et al. has also achieved good results [15].

However, the current PSO algorithm still has the problems of slow convergence and is easy to fall into local optimal solution, so its application in robot path planning is not ideal.

For example, the particle swarm algorithm is applied to the optimized environment model, and the following results are obtained (Figure 5):

It can be seen that the path planning result is not ideal, obviously not the optimal path, and even if the number of iterations is increased, the result will not be significantly improved.

In order to solve the above problems, this paper puts forward the following improvements in the algorithm and its practical application: firstly, by incorporating the concept of mutation factor in simulated annealing algorithm and genetic algorithm, on the improvement of parameters, the paper tries to deal with the linear differential decreasing of weighted inertia, optimize the convergence factor and acceleration factor, and make a comparison. A Simulated Annealing Particle Swarm Optimization Algorithm with Mutation Particles (MP-SAPSO) was proposed in the paper, and Figure 6 illustrates the basic flow of the MP-SAPSO.

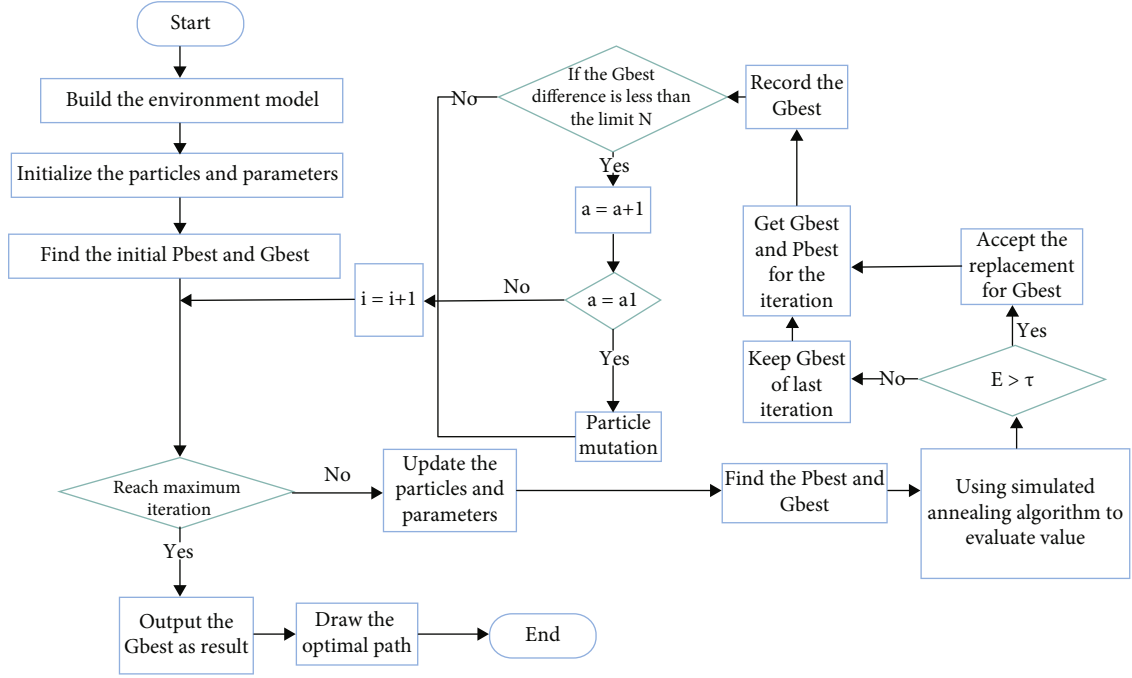


FIGURE 6: MP-SAPSO flow chart.

2.3. *Improvement and Optimization of PSO Algorithm (Simulated Annealing Particle Swarm Optimization Algorithm with Mutation Particles)*. Aiming at the shortcomings of premature convergence of traditional particle swarm optimization, this paper incorporates the concept of population mutation factor in simulated annealing algorithm (SA) and genetic algorithm [16], assembles the optimization for adaptive annealing particle swarm optimization algorithm, and adjusts the original parameters and application parameters in order to apply to the actual path planning application. The following will be divided into two parts, algorithm and parameters, to introduce the kernel logic.

2.3.1. For the Optimization of the Algorithm

(1) *The Combination of Simulated Annealing (SA) and Particle Swarm Optimization (PSO)*. Simulated annealing algorithm (SA) is one of the best algorithms for solving the local optimal solution in deep learning [17]. The algorithm is derived from the crystal cooling process. If the temperature drops too fast, an amorphous with high energy will be produced, and if the temperature drops slowly, regular crystals with high density and low energy will be formed. Combined with the actual results of the algorithm, the noncrystal corresponds to the local optimal solution, and the regular crystal corresponds to the global optimal solution.

In simulated annealing, the process of applying energy to the amorphous to form crystal can be understood as jumping out of the local optimal solution and then continuing to search. The Metropolis algorithm, which is used to accept the new state by probability instead of using the standard rules, is a great choice for the local optimal solution in particle swarm optimization [18].

In each iteration of PSO, the value of each particle is evaluated, and the best choice α of each particle in the group is recorded, which will be compared with the global optimal particle β recorded at this time. If it is better than β , it is replaced and the next iteration is carried out. For this optimization method, we choose to add probability. For each particle I in the n th iteration, it has the optimal solution substitution expectation E and the substitution tolerance τ , where

$$E = e^{-(\alpha(i) - \beta(i)) / \alpha(i)},$$

$$\tau = \text{rand}() \times \frac{n_{\max} - n}{n_{\max}}, \quad (5)$$

random rational numbers with $\text{rand}()$ being (0,1).

If $E > \tau$, the replacement of the particle optimal solution is accepted, otherwise it is rejected.

The convergence trend after optimization is shown in Figures 7 and 8.

It can be seen from the curve that the introduction of “annealing” probability can effectively reduce the number and time of searching for the local optimal solution, and at the same time, it can quickly jump out of the local optimal solution.

(2) *The Embodiment of Population Variation in PSO*. Although SA algorithm can reduce the time and times of falling into the local optimal solution, it cannot change the trend of large-scale particles gathering in the local optimal solution. Therefore, the premature convergence characteristic of the particle swarm algorithm cannot be fundamentally changed solely from the probability.

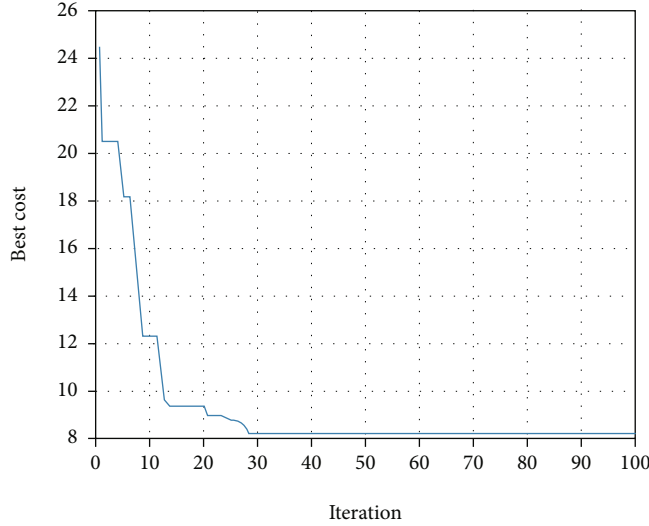


FIGURE 7: Convergence curve of traditional particle swarm optimization.

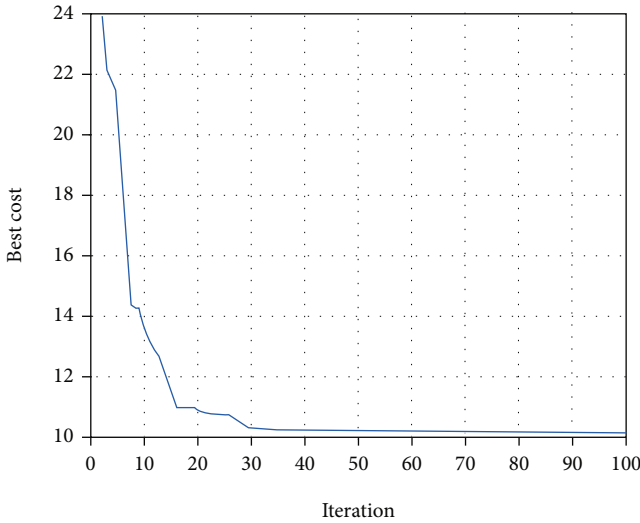


FIGURE 8: Convergence curve after SA optimization.

Considering that particle swarm optimization is also a branch of swarm intelligence, it can also be applied to the treatment of swarm intelligence. So we think of the classic genetic algorithm. The algorithm processing process of genetic algorithm is shown in Figure 9.

Since a single particle does not have many characteristics of chromosome, it consumes too much computational power for calculation, which deviates from the application. So we only use the mutation module to optimize the population [19]. When the state difference between the n th iteration and the $n - 1$ iteration is less than the limit n , record once. When the number of times reaches n_1 , the population variation of particle swarm is simulated. In the application of path planning, the inherent attributes of particles are position (x, y) and velocity (v) .

So we reinitialize the particles of population mutation and bring them into the next iteration to simulate the mutation, as shown in Figure 10.

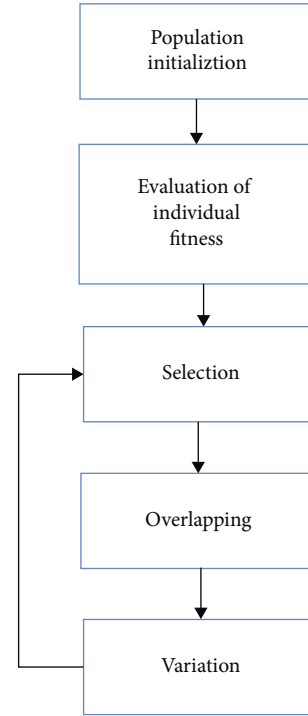


FIGURE 9: Genetic algorithm process.

Specifically, in the mathematical model, we set the randomly selected particles from the original particle swarm number L as mutation particle swarm L_1 and reinitialize the velocity and position of each particle i .

$$\begin{aligned}
 \text{Particle}(i).\text{position} &= \text{create random solution}(\text{model}), \\
 &\downarrow \\
 (x, y) &= (x_0, y_0), \\
 v &= v_0.
 \end{aligned}$$

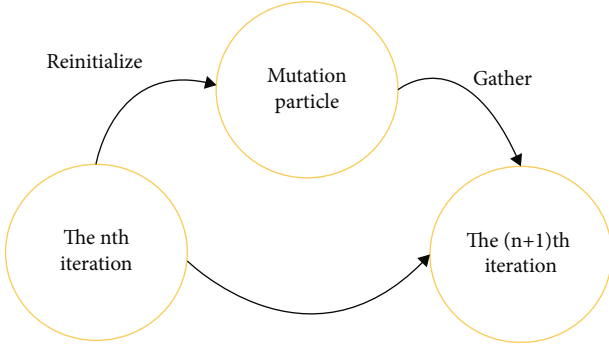


FIGURE 10: Simulation of mutation process.

Substituting the mutation particle swarm into the traditional particle swarm, the comparison is shown in Figures 11 and 12.

We can interpret from Figures 11 and 12 that the addition of mutation particles not only changes the state of being trapped in the local optimal solution for a long time but also successfully updates the global optimal solution even in the later iteration. At the same time, the global optimal solution obtained has a better evaluation value, which is more effective in the obstacle environment model.

2.3.2. Adjustment and Introduction of Parameters. In the particle swarm optimization, not only the algorithm needs to be improved, but also the adjustment of parameters is indispensable. The characteristics of particle swarm optimization requires discretion on the selection of parameters. Attention needs to be paid first to the global search and to the local search, so as to avoid premature local optimal solution and premature convergence. For the optimization of particle swarm optimization parameters, we try two optimization strategies.

(1) Linear Differential Decreasing: Optimization Strategy of Weighted Inertia w . Larger weight inertia is conducive to global search, and smaller weight is conducive to local search [20]. At the same time, due to the aggregation property of particles, the rate of reducing the search range of obstacle environment is proportional to the number of iterations. So for W , linear differential decreasing can satisfy this function relation.

$$\begin{aligned} \frac{dw}{dt} &= \frac{2 \cdot (w_{\max} - w_{\min})}{t_{\max}^2} \cdot t, \\ \int_{w(t)}^{w_m} dw &= \frac{2 \cdot (w_{\max} - w_{\min})}{t_{\max}^2} \int_0^t \pi \tau, \\ w &= w_{\max} - \frac{w_{\max} - w_{\min}}{t_{\max}^2} \cdot t^2, \end{aligned} \quad (7)$$

where t is the number of iterations.

(2) Convergence Factor: Acceleration Factor c_1 and c_2 Optimization Strategy. In order to ensure the convergence, Clerc

and Kennedy introduce a convergence factor K [21] and cancel the introduction of weighted inertia:

$$\begin{aligned} V_i &= K[V_i + \varphi_1 \times \text{rand}() \times (\text{pbest}_i - x_i) \\ &\quad + \varphi_2 \times \text{rand}() \times (\text{gbest}_i - x_i)], \\ K &= \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}, \\ \varphi &= \varphi_1 + \varphi_2, \quad \varphi > 4. \end{aligned} \quad (8)$$

For the comparison of the two strategies, we also substitute into the obstacle environment model to observe the convergence curve and the final result, as shown in Figures 13 and 14.

In terms of convergence speed, although the two strategies are faster than the traditional PSO, the application of convergence factor makes the convergence speed not limited by the number of iterations, resulting in the fact that mutation particles cannot directly reflect the role, and the curve presents the trend of premature convergence (lack of fluctuations about the approximate solution). In contrast, linear differential decrement has a better search process and is compatible with the optimization of mutation jump out of local optimal solution. So we finally adopt the linear differential decreasing scheme.

3. Simulation Results of Path Planning

3.1. Algorithm Parameter Setting. In order to reflect the superiority of the MP-SAPSO algorithm over the traditional PSO algorithm, we apply them to the same environmental model for path planning and solution and analyze the results.

The parameter settings of the two algorithms are shown in Table 1.

In view of the uncomplicated environment model, the adjusted number of iterations is 100 times, and the population size is 150 times, which can reflect the convergence process and tend to be stable. At the same time, we know that when the inertia weight is within $[-4, 4]$ and the learning coefficient is within $[1.25, 2]$, the results can converge quickly and the global optimal solution can be obtained relatively easily [15]. In order to dynamically adjust the inertia weight, we introduced a damping coefficient of 0.8 to effectively make the inertia weight as superior as possible in the iteration [20]. To ensure convergence, we set the constant φ to 4.1 and then calculate the convergence factor K to 0.729 (Table 2) [21].

3.2. Results and Analysis

3.2.1. Analysis of Path Results. By substituting MP-SAPSO into the obstacle environment model and drawing the abscissa of the particle swarm as the global optimal solution, we get the simulation results of path planning for the specified starting point and end point, as shown in Figures 15 and 16.

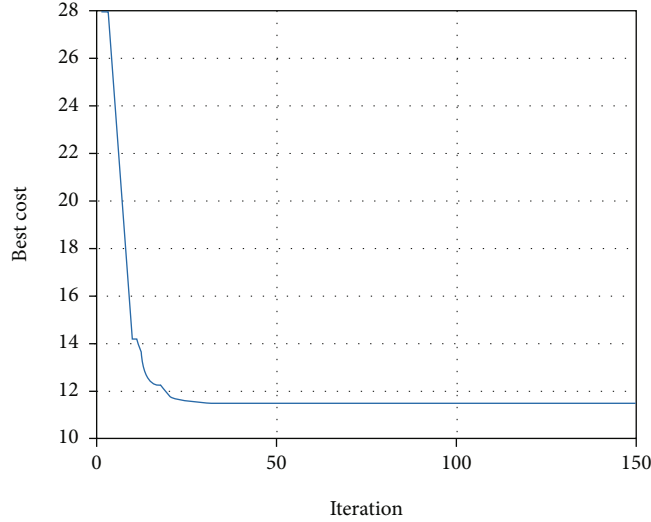


FIGURE 11: 150 times iteration curve of traditional particle swarm optimization.

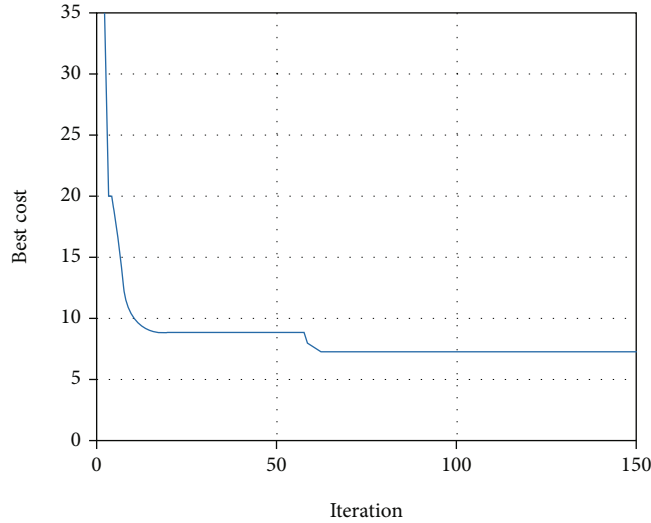


FIGURE 12: Variation particle swarm optimization curve.

As can be seen from Figure 15, the traditional particle swarm optimization fell into premature convergence, causing the calculation to stop in the early iteration. The convergence result shows a steep drop and several invalid iterations; in the 20th iteration, it was basically the same as the final result, and no optimization route planning was carried out. More importantly, the planned route does not achieve the result of crossing the obstacle model but finally shows the route out of the model.

It can be seen from Figure 16 that after MP-SAPSO is applied, the convergence of PSO is continuous. In the 30th iteration, PSO successfully jumps out of the local optimal solution, realizes the optimization function of mutation particle, and obtains the result that is more in line with the best route (through obstacles).

3.2.2. Algorithm Complexity Analysis. Since the core calculation process of the traditional PSO algorithm needs to

update the position of each individual in each iteration of the particle swarm, we can get

$$T(n) = O(I_{\max} \times P_{\text{size}}), S(n) = O(I_{\max} \times P_{\text{size}}). \quad (9)$$

Compared with MP-SAPSO, we found that the mutation factor needs to judge individual particles after a single iteration, and the substitution expectation E for this iteration needs to be calculated. Both have no effect on the number of iterations and the calculation of the population size. Therefore, the traditional PSO algorithm has the same complexity as the MP-SAPSO algorithm, and the introduction of parameters does not affect the solution time and the required calculation space.

3.2.3. Analysis of Algorithm Superiority. In order to fully prove the superiority of the MP-SAPSO algorithm from the perspective of statistical analysis, we use the method of

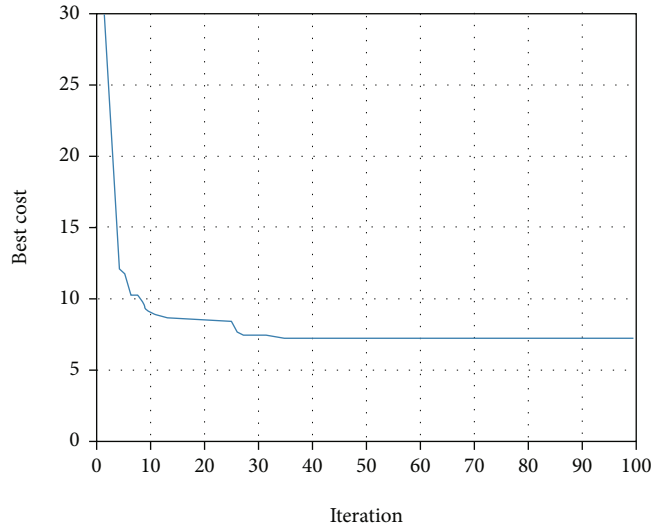


FIGURE 13: Linear differential decreasing w optimization.

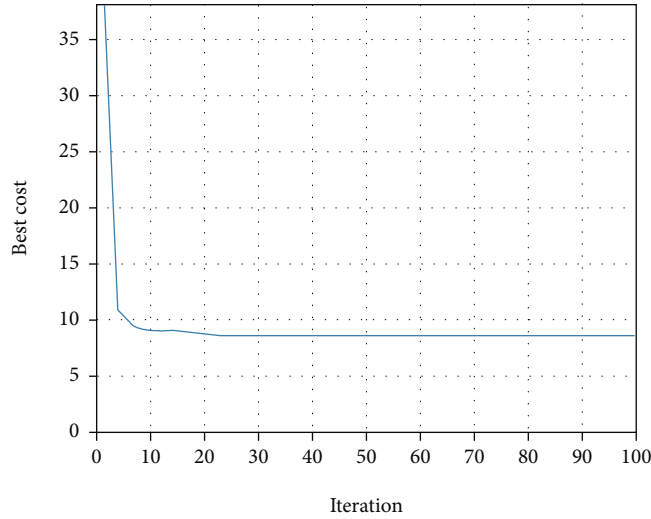


FIGURE 14: Convergence factor optimization.

TABLE 1: Parameter setting.

Parameter	Description	Traditional PSO	MP-SAPSO
I_{\max}	Maximum number of iteration	100	100
P_{size}	Population size	150	150
w	Inertia weight	1	1
w_{dr}	Inertia weight damping ratio	0.8	0.8
c_1	Personal learning coefficient	1.5	\
c_2	Global learning coefficient	1.5	\
φ	Constant	\	4.1
K	Convergence factor	\	0.729

TABLE 2: Algorithm complexity analysis.

Algorithm	Time complexity $T(n)$	Spatial complexity $S(n)$
Traditional PSO	$O(I_{\max} \times P_{\text{size}})$	$O(I_{\max} \times P_{\text{size}})$
MP-SAPSO	$O(I_{\max} \times P_{\text{size}})$	$O(I_{\max} \times P_{\text{size}})$

running the algorithm multiple times to extract data, use the two algorithms to solve the set environment model for multiple paths, and calculate the output cost function. After analysis, the preliminary results are shown in Table 3.

Since the final iteration will tend to stabilize and fluctuate and the standard deviations of the two algorithms in multiple experiments are known, the cost function results of the two algorithms can be sorted again by the approximate normal distribution, and the paired-sample T -test is used for statistical analysis (Table 4).

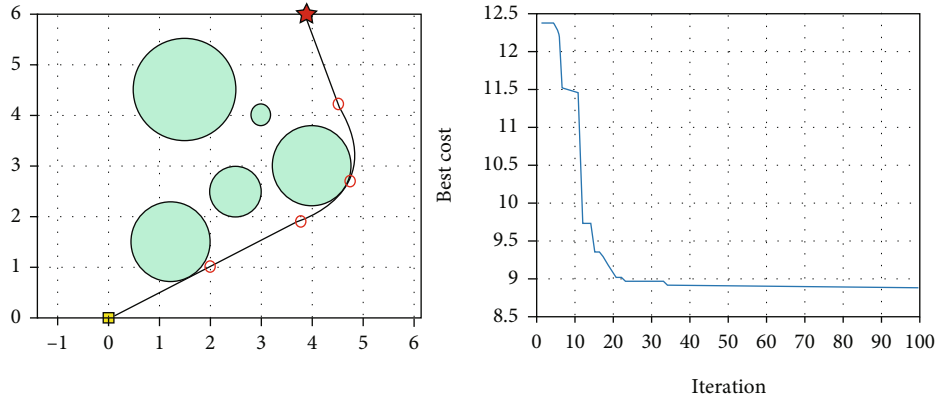


FIGURE 15: Simulation results of traditional particle swarm optimization.

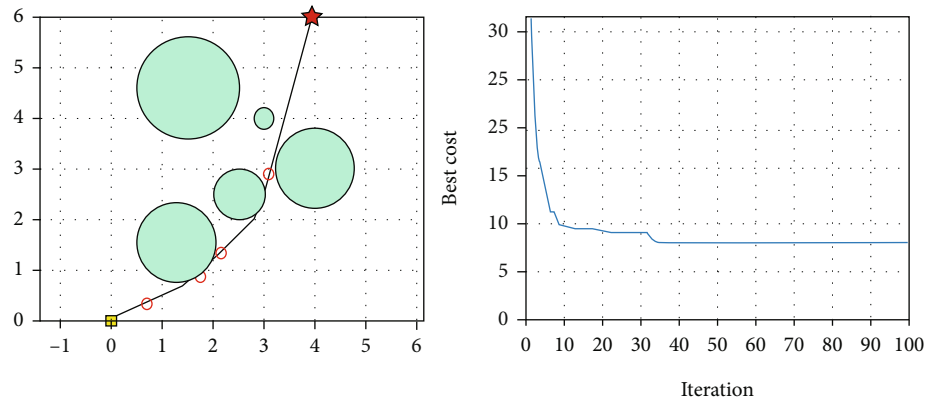


FIGURE 16: MP-SAPSO optimization results.

TABLE 3: Statistical analysis.

Algorithm	Best cost	Average cost	Standard deviation
Traditional PSO	8.6559	8.6790	0.0054
MP-SAPSO	7.6826	7.7608	0.0079

TABLE 4: Paired-sample T -test analysis.

Paired-sample T -test	Standard deviation	Mean value of standard error	P (significance)
PSO_ MP-SAPSO	0.0030	0.0004	$9.4306E-125$

It can be seen that $P < 0.001$, so the reduction in the output value of the cost function is very much related to the application of MP-SAPSO. It is proved that MP-SAPSO reduces the value of the cost function, which further reflects the advantage of MP-SAPSO to plan more accurate and better paths easily.

3.3. Advantages and Further Research of the Algorithm. Compared with the traditional particle swarm optimization algorithm, the MP-SAPSO algorithm can obtain the valid

path in the obstacle environment more accurately. The simulated annealing algorithm (SA) is combined with the concept of population mutation factor in genetic algorithm to solve the premature convergence problem of traditional particle swarm algorithm. When there are multiple routes, the algorithm can jump out of invalid iterations and find more feasible routes. This helps the robot to improve the accuracy of path selection in obstacle environments. According to the complex calculation and statistical analysis, it can be concluded that after several rounds of experimental verification, the MP-SAPSO algorithm does not significantly increase the calculation time compared with the traditional algorithm, and the calculation space does not take up too much. The reduction of the cost function is proved to be significantly correlated with the application of the improved algorithm by the paired-sample t -test. This proves the superiority of the present algorithm to the traditional algorithm.

Although the establishment of the obstacle environment can avoid the influence of the edges and corners of the obstacle itself on the path selection of the robot, many specific environmental parameters (obstacle size, robot size) are not taken into consideration, including the discussion in a three-dimensional space. Therefore, further research and exploration are needed for more complex obstacle environments.

4. Conclusion

In this paper, an improved particle swarm optimization algorithm MP-SAPSO is proposed to solve the robot path planning problem. In building a mathematical model, considering the complexity of the obstacle model, and to facilitate the operation in practical application, we build the untouchable space as the obstacle environment model, which is easy to operate in practical application. At the same time, MP-SAPSO can make a better path planning strategy in the obstacle environment by screening the result of annealing probability and introducing mutation particles; by introducing linear differential decreasing inertia weight w , the convergence speed is not affected by the introduction of the new algorithm. Finally, without sacrificing the efficiency of the algorithm, the accuracy of path planning and the efficiency of robot operation are improved.

The simulation results show that compared with the existing methods, the application of this algorithm can make the robot find a better mobile path faster. It has a great application prospect in application scenarios such as UAV, rescue robot, cleaning robot, and military robot that need to ensure operational efficiency and accuracy. However, due to the simple design of the model, the complex situation in the actual application scenario is not covered in this paper. Some real parameters are not available, and the research of some specific application scenarios needs to be further deepened to strengthen the ability of this algorithm to deal with specific problems, which will be further explored in future research.

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] M. Nazarahari, E. Khanmirza, and S. Doostie, "Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm," *Expert Systems with Applications*, vol. 115, pp. 106–120, 2019.
- [2] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, pp. 39–43, Nagoya, Japan, 1995.
- [3] C. Katrakazas, M. Quddus, W. H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: state-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015.
- [4] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on industrial informatics*, vol. 9, no. 1, pp. 132–141, 2013.
- [5] H. Mo and L. Xu, "Research of biogeography particle swarm optimization for robot path planning," *Neurocomputing*, vol. 148, pp. 91–99, 2015.
- [6] J. Yu and D. Rus, *An effective algorithmic framework for near optimal multi-robot path planning* September 2021, <https://arc.cs.rutgers.edu/files/YuRus15ISRR.pdf>.
- [7] G. Li, S. Tong, G. Lv et al., "An improved artificial potential field-based simultaneous FORWARD search (improved APF-based SIFORS) method for robot path planning," in *2015 12th international conference on ubiquitous robots and ambient intelligence (URAI)*, pp. 330–335, Goyangi, Korea (South), 2015.
- [8] M. H. Jiao, H. X. Wei, B. W. Zhang, J. Q. Jin, Z. Q. Jia, and J. L. Yan, "Path planning of escort robot based on improved quantum particle swarm optimization," in *2019 Chinese control and decision conference*, Nanchang, China, 2019.
- [9] A. Tharwat, M. Elhoseny, A. E. Hassanien, T. Gabel, and A. Kumar, "Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm," *Cluster Computing*, vol. 22, no. S2, pp. 4745–4766, 2019.
- [10] P. K. Das, H. S. Behera, S. Das, H. K. Tripathy, B. K. Panigrahi, and S. K. Pradhan, "A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment," *Neurocomputing*, vol. 207, pp. 735–753, 2016.
- [11] N. Zeng, H. Zhang, Y. Chen, B. Chen, and Y. Liu, "Path planning for intelligent robot based on switching local evolutionary PSO algorithm," *Assembly Automation*, vol. 36, no. 2, pp. 120–126, 2016.
- [12] B. Ames, A. Beveridge, R. Carlson et al., "A leapfrog strategy for pursuit-evasion in a polygonal environment," *International Journal of Computational Geometry and Applications*, vol. 25, no. 2, pp. 77–100, 2015.
- [13] W. J. Yang, H. B. Wang, and J. H. Wang, "Research on path planning for mobile robot based on grid and hybrid of GA/SA," in *Advanced Materials Research*, vol. 479–481, pp. 1499–1503, 2012.
- [14] Z. Y. Zhou, P. A. Mu, and R. J. Zhang, "Application of PSO optimization algorithm in smooth path planning," *Software Guide*, vol. 20, no. 3, pp. 67–72, 2021.
- [15] Q. M. Yan, R. Q. Ma, Y. X. Ma et al., "An adaptive simulated annealing particle swarm optimization algorithm," *J Xi'an U Electron Sci Te*, vol. 48, no. 4, pp. 120–127, 2021.
- [16] F. Javidrad and M. Nazari, "A new hybrid particle swarm and simulated annealing stochastic optimization method," *Applied Soft Computing*, vol. 60, pp. 634–654, 2017.
- [17] R. Da Silva, E. V. Filho, and A. Alves, "A thorough study of the performance of simulated annealing in the traveling salesman problem under correlated and long tailed spatial scenarios," *Physica A*, vol. 577, article 126067, 2021.
- [18] S. Stefan and J. Wolfhard, "Accelerating polymer simulation by means of tree data-structures and a parsimonious Metropolis algorithm," *Computer Physics Communications*, vol. 256, article 107414, 2020.
- [19] A. Jamali, R. Mallipeddi, M. Salehpour, and A. Bagheri, "Multi-objective differential evolution algorithm with fuzzy inference-based adaptive mutation factor for Pareto optimum design of suspension system," *Swarm and Evolutionary Computation*, vol. 54, article 100666, 2020.

- [20] P. C. Cai and D. M. Zhang, "Improved gray wolf algorithm based on dynamic inertia weight and adaptive search," *Journal of Physics: Conference Series*, vol. 1802, no. 3, article 032122, 2021.
- [21] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.