Solving Path Planning Problem Based on Particle Swarm Optimization Algorithm with Improved Inertia Weights

Yi-Xuan Lu, Jie-Sheng Wang*, and Sha-Sha Guo

Abstract—**The path planning problem refers to find the shortest path to reach the predetermined target position in a certain complex environment. Particle swarm optimization (PSO) algorithm is derived from the imitation of the population cooperation of the flock and the predatory behavior of the competition. The sharing of information by the individuals in the swarm makes the movement of the whole swarm in the problem solution space from disorder process to order process. In this paper, the improved PSO algorithm based on improved inertia weights is adopted to solve the path planning problems. For the three constructed different maps, the improved PSO algorithm based on five different inertia weight adjustment strategies is used to solve the path planning problems. The simulation results are used to verify the effectiveness of the proposed algorithm and inertia weight adjustment strategies.**

***Index Terms*—path planning problem, particle swarm optimization algorithm, inertia weight**

I. INTRODUCTION

THE path planning problem is one of the main research contents of motion planning. The motion planning problem consists of path planning and trajectory planning. The sequence points or curves connecting the starting point and the ending position are called paths. The strategy of forming the path is called path planning [1]. The planning problem with the topological of point and line networks can be basically solved by the path planning methods. Therefore, the path planning has been widely applied in many application fields, such as the robot autonomous collision-free operation, the drone obstacle avoidance flight, the cruise missile avoiding the radar searching, the GPS navigation, the GIS-based road planning, the vehicle routing problem (VRP) in the city road network planning and navigation, and the routing problem in communication technology [2-18].

Manuscript received February 28, 2019; revised April 21, 2019. This work was supported by the the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. 2017FWDF10), and the Project by Liaoning Provincial Natural Science Foundation of China (Grant No. 20180550700).

Yi-Xuan Lu is an undergraduate student in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail: 1095545671@qq.com).

Jie-Sheng Wang is with the School of International Finance and Banking, University of Science and Technology Liaoning, Anshan, 114051, PR China; National Financial Security and System Equipment Engineering Research Center, University of Science and Technology Liaoning. (Corresponding author, phone: 86-0412-2538246; fax: 86-0412-2538244; e-mail: wang_jiesheng@126.com).

Sha-Sha Guo is a postgraduate student in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail: 1909133855@qq.com).

According to the basic principles of the path planning solving algorithms, they can be roughly divided into the classical algorithms, the graphic methods and intelligent hybrid algorithms. A* algorithm, Dijkstra algorithm, C-space algorithm, Floyd algorithm, Voronoi diagram, particle swarm optimization algorithm, the mixed integer linear programming method, genetic algorithm, heuristic search method and ant colony algorithm have been applied on solving the path planning problem.

Particle swarm optimization (PSO) algorithm is a new evolutionary algorithm (EA) developed by J. Kennedy and R. C. Eberhart [19]. The PSO algorithm also starts from the random solutions and finds the optimal solution through iteration. It also evaluates the quality of the solutions by fitness, and finds the global optimum by following the current searched optimal solution. This kind of algorithm has attracted the attention of the academic scholars because of its easy implementation, high precision and fast convergence. It has demonstrated its superiority in solving many practical problems, such as the prediction of seismic slope stability, carton heterogeneous vehicle routing problem. The optimal power management of plug-in hybrid electric vehicles, the optimal operation of micro grid, the electrochemical model parameter identification of a lithium-ion battery, the mixed-model two-sided assembly line balancing, the multidimensional Knapsack problem and the trajectory optimization of manipulator motion planning problem [20-28]. Inertia weight is an important parameter in PSO algorithm. In this paper, the PSO algorithm based on different improved inertia weights is used to solve the path planning problem. The simulation experiments are carried out to evaluate the performance of different inertia weights on the PSO algorithm to solve the typical path planning problems.

II. PARTICLE SWARM OPTIMIZATION ALGORITHM AND IMPROVED INERTIA WEIGHTS

A. Basic Principles of Particle Swarm Optimization Algorithm

The particle swarm optimization (PSO) algorithm was proposed by Eberhart and Kennedy in 1995. It originated from the imitation of population cooperative and competing predation behavior of birds. The swarm behavioral measures, such as competition and collaboration, guide the entire population moving toward food. Based on the observation of the activity behavior of animal swarms, the PSO algorithm adopts the individual's sharing of information in the swarm to make the movement of the whole swarm in the problem solving space from the disordered to the orderly evolution process so as to obtain the optimal solution. The general mathematical description of the PSO algorithm is described as follows.

Assume that there are m particles in a swarm in an N-dimensional searching space, the velocity and position of the d-dimensional component of the i-th particle are updated by the following equations.

$$v_{id}^{k+1} = \omega \times v_{id}^{k} + c_1 r_1 \times \left(p_{id}^{k} - x_{id}^{k}\right) + c_2 r_2 \times \left(g_{id}^{k} - x_{id}^{k}\right) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \quad (2)$$

where, v_{id}^{k+1} is the d-dimensional component of the velocity vector of particle i at the $k+1$ iteration, x_{id}^{k+1} is the d-dimensional component of the position vector of particle i at the $k+1$ iteration, p_{id}^{k} is the d-dimensional component of the position vector of particle i at the k iteration, and g_{id}^{k} is the d-dimensional component of the position vector of the global optimum at the k iteration. c_1 and c_2 are the acceleration factors, which are usually non-negative constants and have the ability to make particles learn from the best individuals in the population, and then approach the historical optimum and the historical optimum within the population. r_1 and r_2 are random numbers located in the scope [0,1]. ω is the inertia weight, which usually affects the searching ability of the algorithm in some part of the space. The proper adjustment of ω can overcome the problem that the particles fall into the local optimal values.

B. Algorithm Flowchart

The flowchart of the PSO algorithm is shown in Fig. 1. The specific algorithm procedure is described as follows.

Step 1: Initialize the relevant parameters of the PSO algorithm, the velocity and the position of each particle in the population.

Step 2: Calculate the fitness value of each particle.

Step 3: For all particles, compare their fitness values with the individual extreme value. If the fitness value is better than the individual extreme value, replace the individual extreme value with the fitness value.

Step 4: For each particle, compare its fitness value with the global extreme. If the fitness value is better than the global extreme, replace the global extreme with the fitness value.

Step 5: Update the velocity and position of the particles based on Eq. (1) and Eq. (2).

Step 6: If the termination condition is satisfied, output the results, otherwise return to Step 2.

C. Improved Inertia Weights

It can be seen from the Eq. (1) that the inertia weight in the PSO algorithm is an important factor in regulating the local search and global search ability. The larger the inertia weight, the stronger the ability to search globally, but the local search ability will be weakened. Conversely, the smaller the inertia weight, the stronger the ability to search locally and the weaker the global search ability. The former may cause the algorithm to ignore an optimal solution, but the latter may make the PSO algorithm fall into the local optimum.

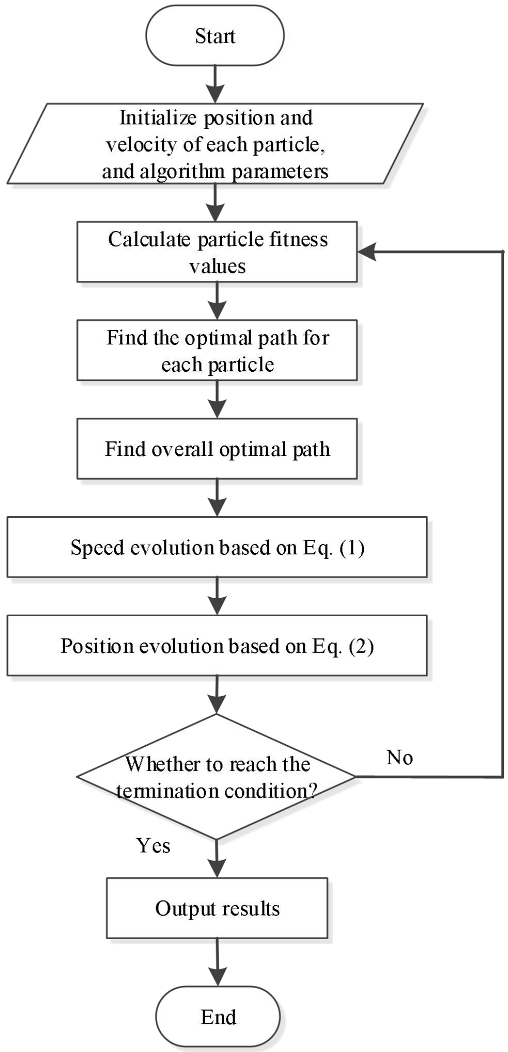


Fig. 1 Flowchart of particle swarm optimization algorithm.

(1) Decremented Inertia Weight

This inertia weight is gradually reduced as the number of iterations increases [29]. ω is a function of the iterations number, which is linearly decreasing from 0.9 to 0.4 along the straight line. The relationship between ω and the iteration number is described as follows.

$$\omega(k) = -0.5 \times \frac{k}{MaxNumber} + 0.9 \quad (3)$$

where, k is the iteration number, and $MaxNumber$ is the maximum iteration number.

(2) Incremental Inertia Weight

This inertia weight is gradually increased as the number of iterations increases [30]. The inertia weight ω is a function of the iteration number, which is linearly increasing from 0.4 to 0.9 along the straight line. The relationship between ω and the iteration number is shown as follows.

$$w(k) = 0.5 \times \frac{k}{MaxNumber} + 0.4 \quad (4)$$

where, k is the iteration number, and $MaxNumber$ is the maximum iteration number.

(3) Incremental-Decremented Inertia Weight

In this method, with the increment of the iteration number, the inertia weight is linearly increased firstly, and then linearly decremented [31]. It has a good global searching ability in the early stage of the algorithm, and the convergence is better in the later stage, that is to say that the inertia weight ω slowly and linearly increases from 0.4 to 0.9 along the fold line, then linearly decreases to 0.4. The incremental-decremented inertia weight ω is adjusted by Eq. (5).

$$\omega(k)=\begin{cases}1\times\dfrac{k}{MaxNumber}+0.4 & 0\le\dfrac{k}{MaxNumber}\le 0.5\\ -1\times\dfrac{k}{MaxNumber}+1.4 & 0.5\le\dfrac{k}{MaxNumber}\le 1\end{cases}\tag{5}$$

where, k is the iteration number, and $MaxNumber$ is the maximum iteration number.

(4) Stochastic Dynamic Inertia Weight

The stochastic dynamic inertia weight is shown in Eq. (6) [32]. It has a dynamic inertia factor, which is a random number between 0 and 1.

$$\omega=0.5+\frac{rand(\ )}{2}\tag{6}$$

where, $rand()$ is a random number between 0 and 1.

(5) Sinusoidal function inertia weight

The inertia weight with the sinusoidal function [33] is described as follows.

$$\omega(k)=0.4+0.5\sin\left(\pi k/k_{\max}\right)\tag{7}$$

where, k is the iteration number and $k_{\max}$ is the maximum iteration number. It can be seen from Eq. (7) that ω is roughly sinusoidal. In the algorithm, the particles do a local search in the vicinity of itself, and then carry out the global search to find the optimal value.

III. Simulation Experiments and Results Analysis

A. Different Inertia Weights

The simulation environment in this paper adopts the *Windows*7 operating system, the *Intel* 2.40GHz processor with 8G memory, and the Matlab 2014a software. In the simulation experiments, five different inertia weight adjusting strategies ($\omega_1(k)$, $\omega_2(k)$, $\omega_3(k)$, $\omega_4(k)$ and $\omega_5(k)$) are selected to carry out the simulation experiments, whose expressions and parameters are listed in Table 1.

B. Map Construction

The Matlab simulation software is use to construct three different maps with the related obstacles. The specifications of the map are 6 units in the horizontal direction and 8 units in the vertical direction. They are placed in a plane rectangular coordinate system. The solution to be found by the improved PSO algorithm is the accessibility shortest path from the starting point (0,0) to the ending point (6,8). The constructed three maps are shown in Fig. 2-4.

C. Simulation Results and Analysis

In order to verify the effectiveness of the adopted strategies and demonstrate the fairness of the simulation experiments, the population with the same initialized species and algorithm parameters are adopted. In this simulation experiments, the population size is 200, the number of iterations is 100, the maximum weight inertia is 0.98, and the learning factors are set as 2. Simulation experiments are carried out based on PSO algorithm with five different inertia weight adjustment strategies to solve constructed three path planning maps. The simulation results by using PSO algorithm with five different inertia weight adjustment strategies to solve the path planning problem (map 1#) are shown in Fig. 5 and Fig. 6. The simulation results by using PSO algorithm with five different inertia weight adjustment strategies to solve the path planning problem (map 2#) are shown in Fig. 7 and Fig. 8. The simulation results by using PSO algorithm with five different inertia weight adjustment strategies to solve the path planning problem (map 3#) are shown in Fig. 9 and Fig. 10. The comparison results of simulation experiments are shown in Tab. 2.

TABLE 1. SIMULATION TESTING FUNCTIONS

Inertia weight	Expression	Range
Decremented inertia weight	$\omega_1(k)=-0.5\times\frac{k}{MaxNumber}+0.9$	[0.4，0.9]
Incremental inertia weight	$\omega_2(k)=0.5\times\frac{k}{MaxNumber}+0.4$	[0.4，0.9]
Incremental-Decremented inertia weight	$\omega_3(k)=\begin{cases}1\times\frac{k}{MaxNumber}+0.4 & 0\le\frac{k}{MaxNumber}\le 0.5\\ -1\times\frac{k}{MaxNumber}+1.4 & 0.5\le\frac{k}{MaxNumber}\le 1\end{cases}$	[0.4，0.9]
Stochastic dynamic inertia weight	$\omega_4=0.5+\frac{rand(\ )}{2}$	[0，1]
Sinusoidal function inertia weight	$\omega_5(k)=0.4+0.5\sin(\pi k/k_{\max})$	[0.4，0.9]

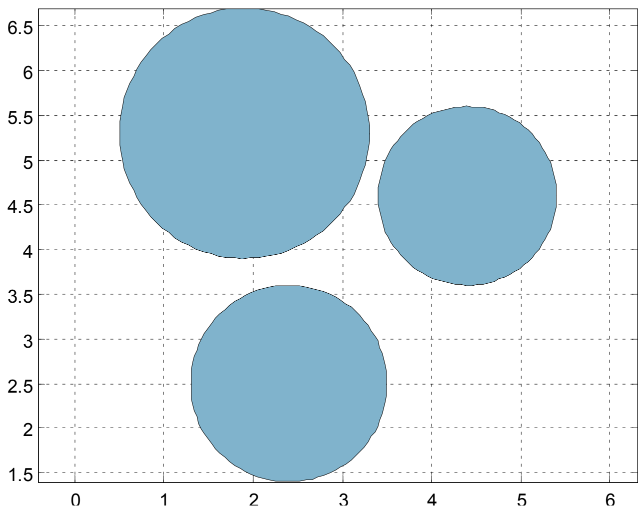
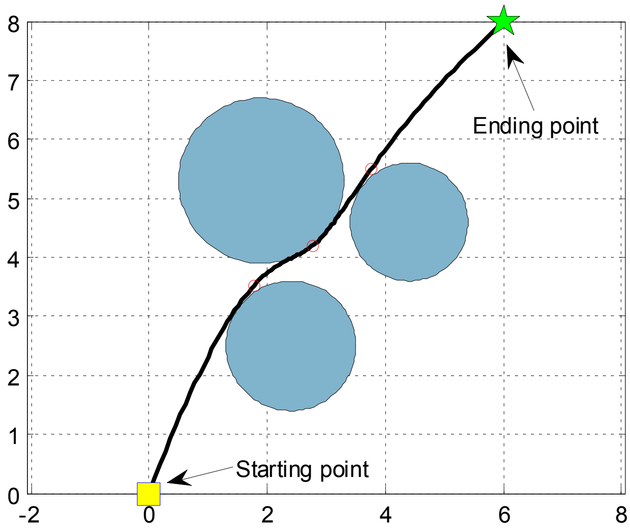


Fig. 2 Constructed map 1#.



(a) Inertia weight adjustment strategy 1#

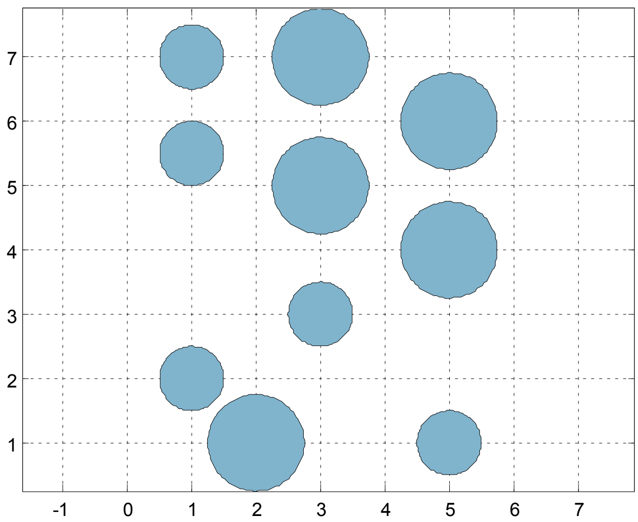
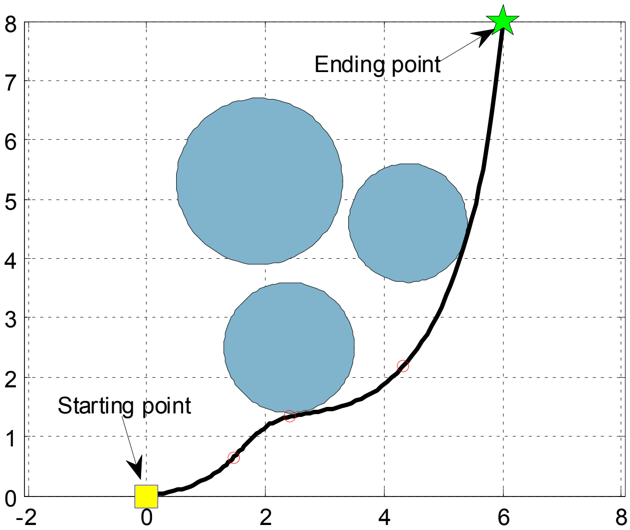


Fig. 3 Constructed map 2#.



(b) Inertia weight adjustment strategy 2#

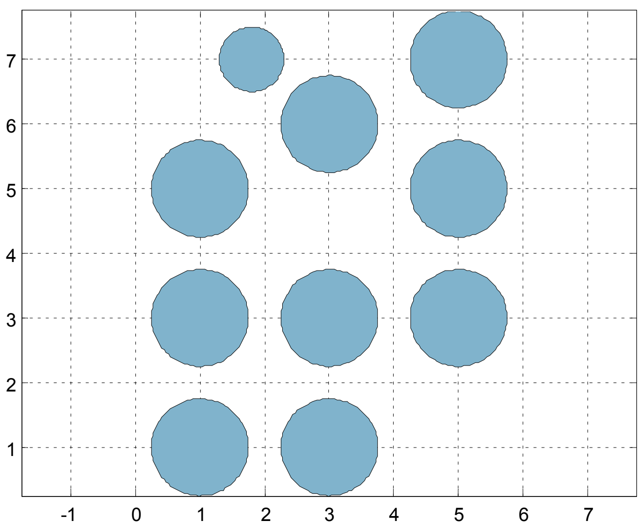
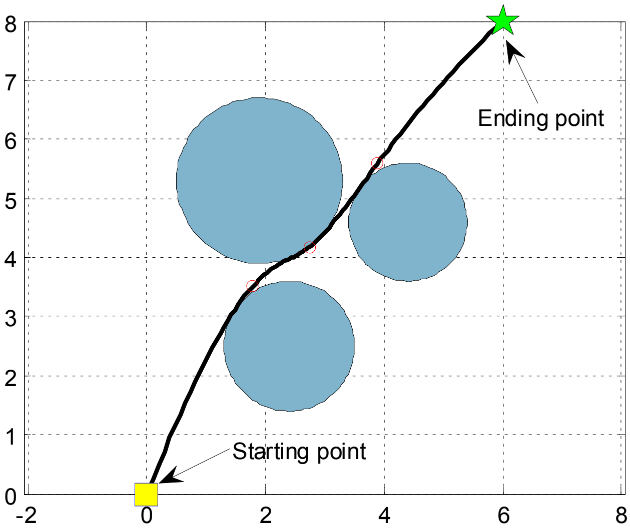
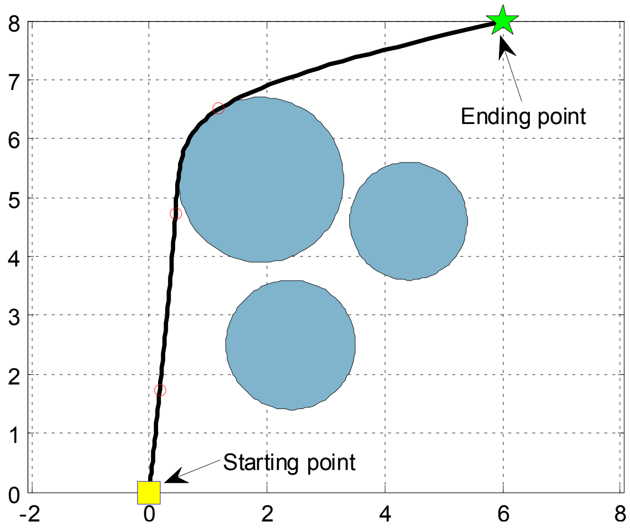


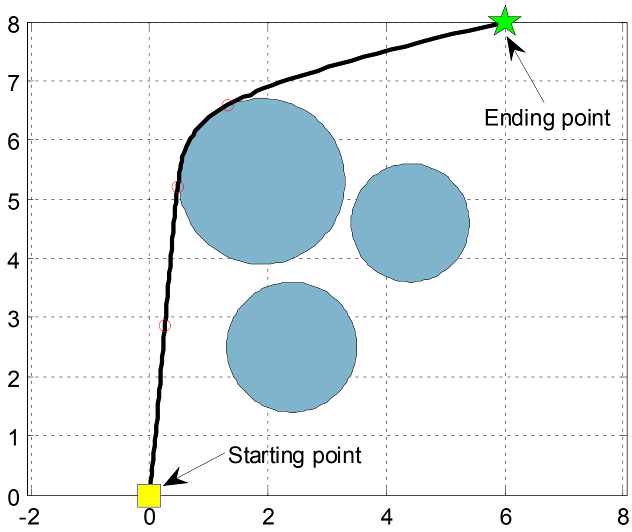
Fig. 4 Constructed maps 3#.



(c) Inertia weight adjustment strategy 3#



(d) Inertia weight adjustment strategy 4#



(e) Inertia weight adjustment strategy 5#

Fig. 5 Simulation results based on different inertia weight adjustment strategies (Map 1#).

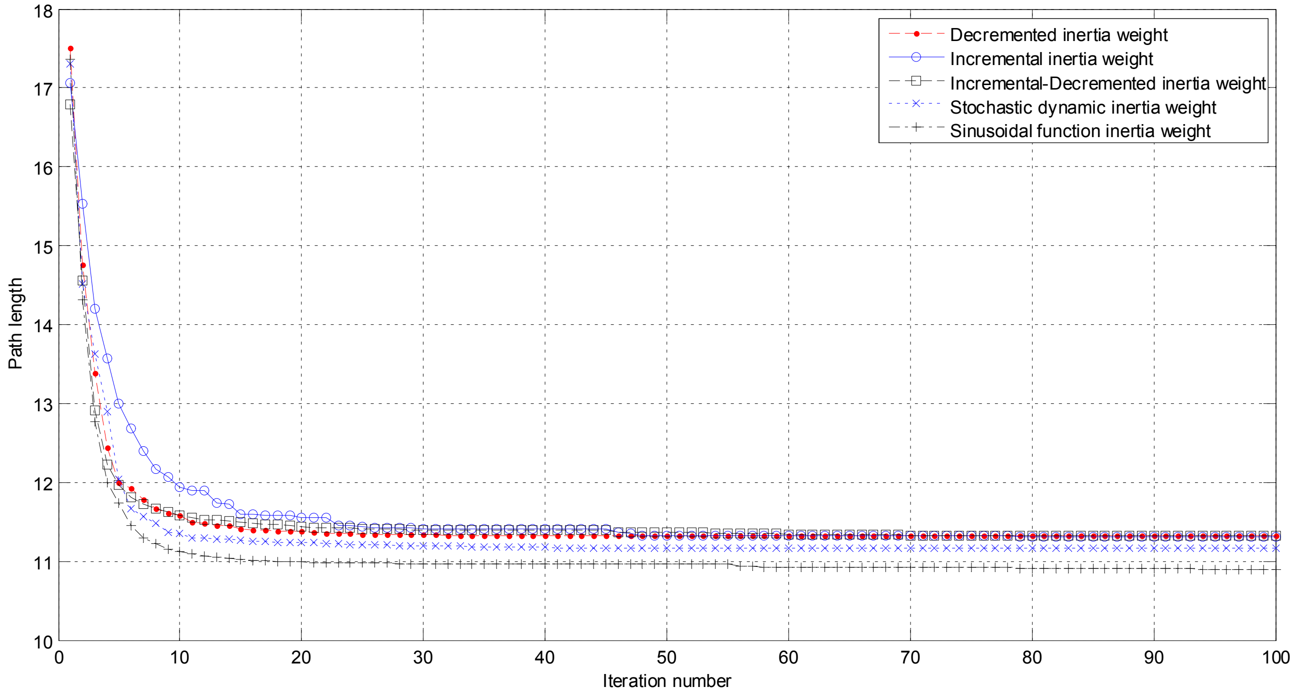
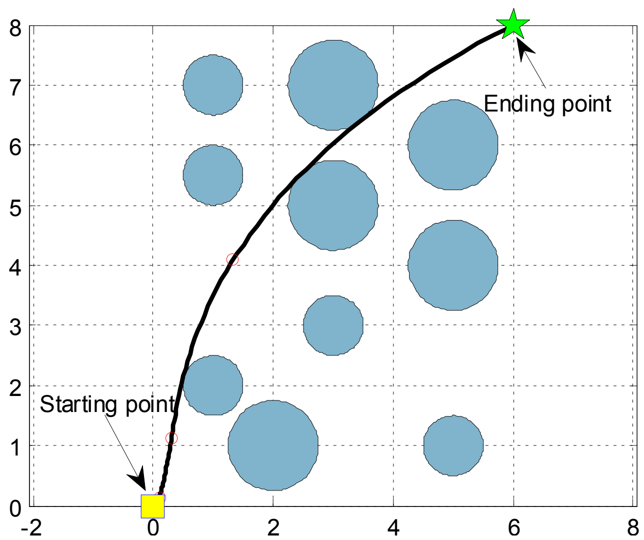
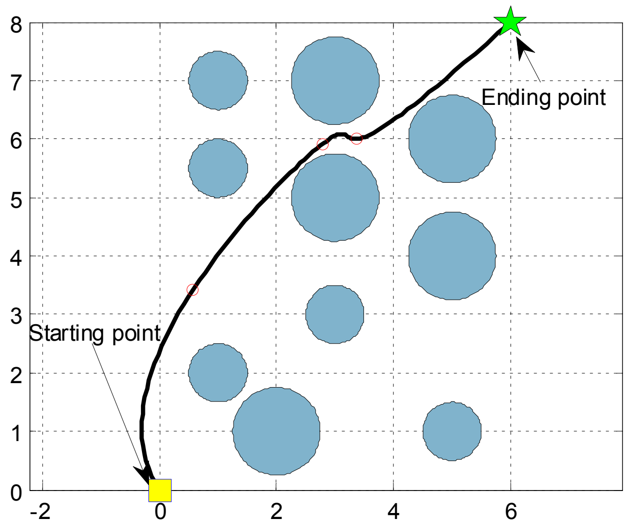


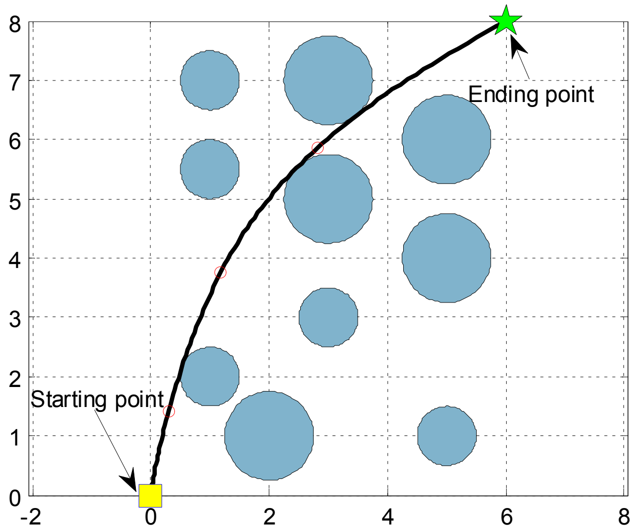
Fig. 6 Convergence curves based on five different inertia weights (Map 1#).



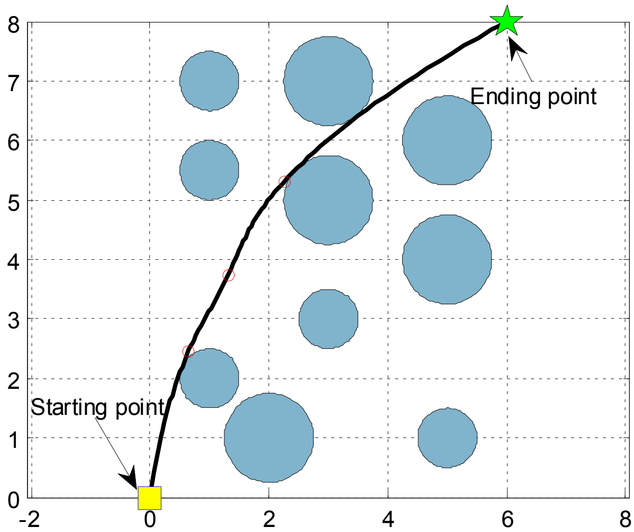
(a) Inertia weight adjustment strategy 1#



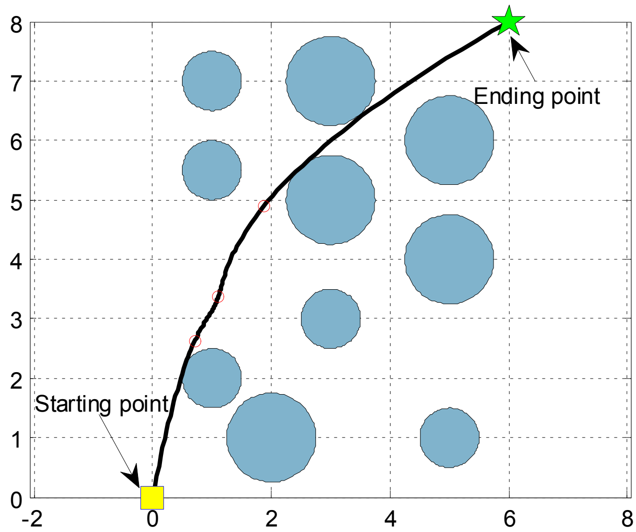
(b) Inertia weight adjustment strategy 2#



(c) Inertia weight adjustment strategy 3#



(d) Inertia weight adjustment strategy 4#



(e) Inertia weight adjustment strategy 5#

Fig. 7 Simulation results based on different inertia weight adjustment strategies (Map2#).

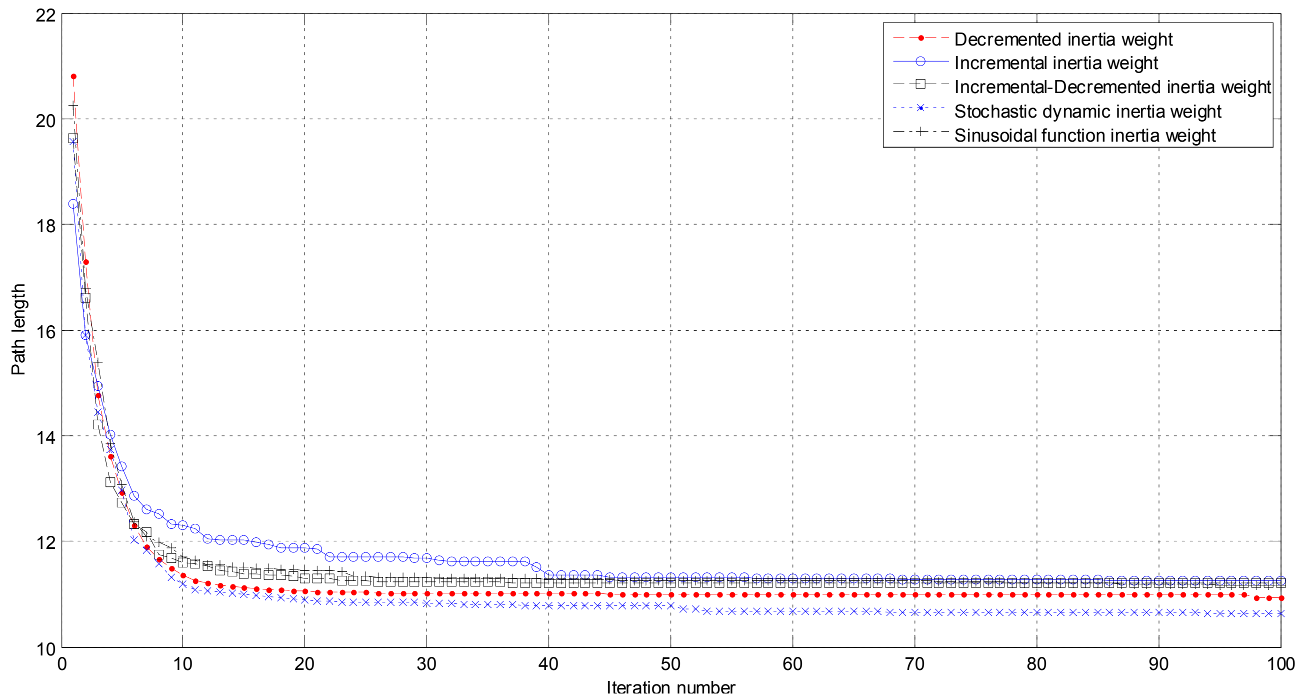
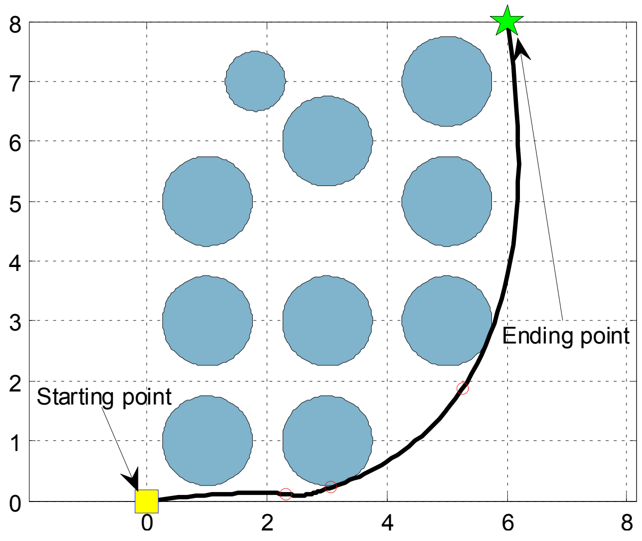
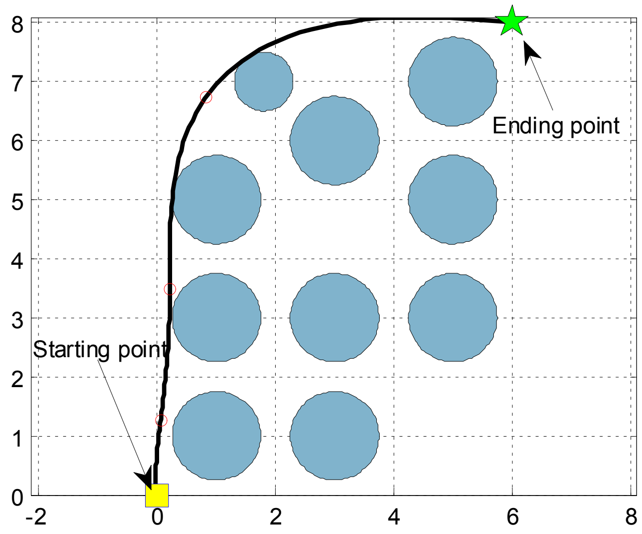


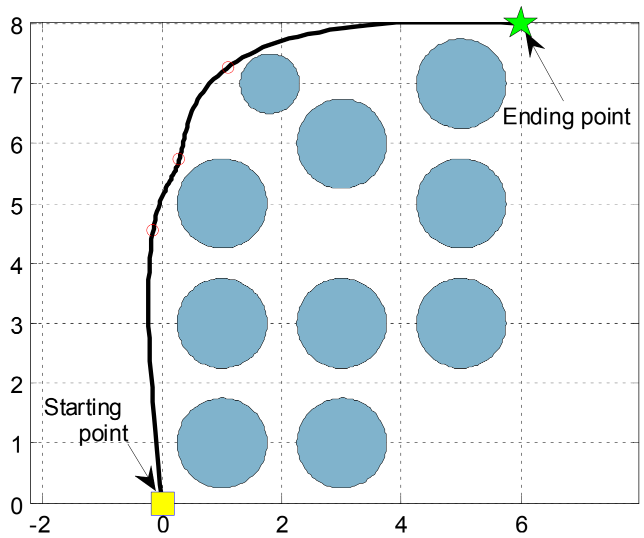
Fig. 8 Convergence curves based on five different inertia weights (Map 2#) .



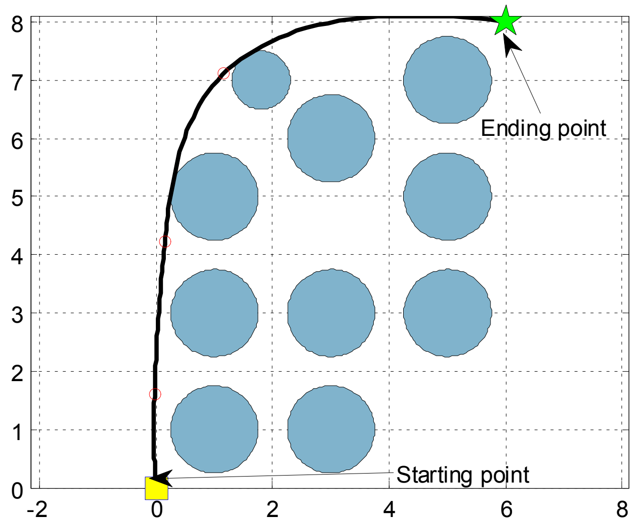
(a) Inertia weight adjustment strategy 1#



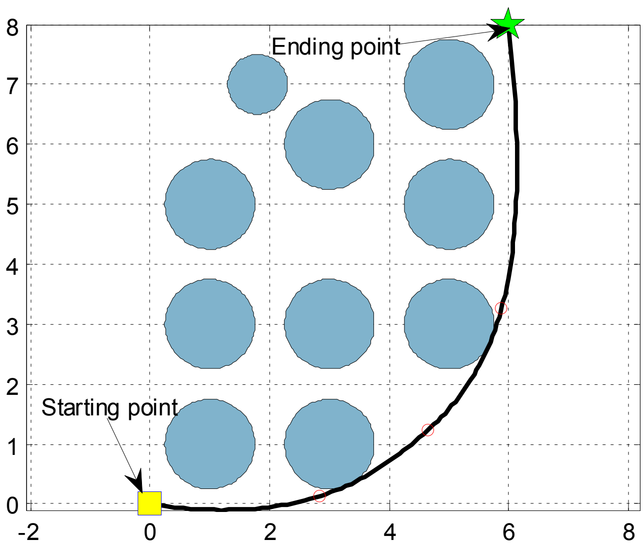
(d) Inertia weight adjustment strategy 4#



(b) Inertia weight adjustment strategy 2#



(e) Inertia weight adjustment strategy 5#



(c) Inertia weight adjustment strategy 3#

Fig. 9 Simulation results based on different inertia weight adjustment strategies (Map 3#).

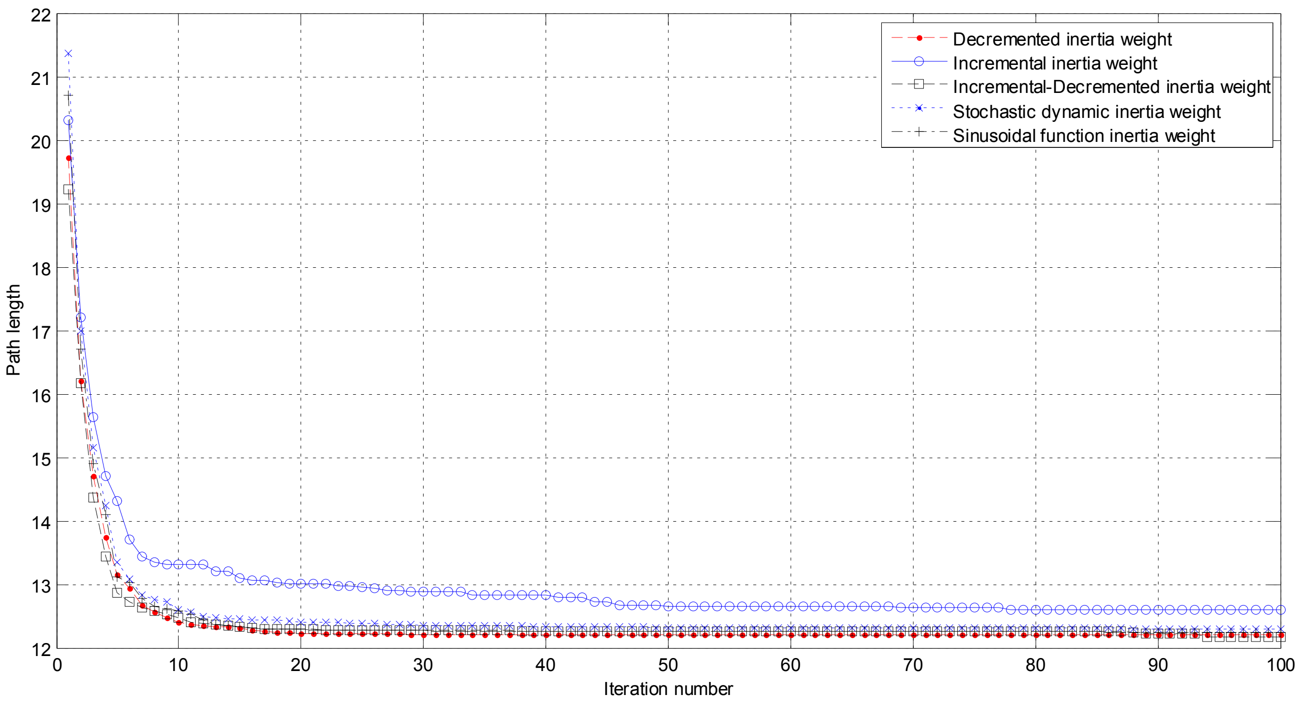


Fig. 10 Convergence curves based on five different inertia weights (Map 3#).

TABLE 2 COMPARISON OF SIMULATION PERFORMANCE

		PSO algorithm with decremented inertia weight	PSO algorithm with incremental inertia weight	PSO algorithm with incremental-decremented inertia weight	PSO algorithm with stochastic dynamic inertia weight	PSO algorithm with sinusoidal function inertia weight
Map 1#	Maximum iteration number	32	80	68	55	90
	Optimal path length	11.32	11.39	11.32	11.16	10.90
Map 2#	Maximum iteration number	36	85	70	70	89
	Optimal path length	11.0	11.25	11.20	10.65	11.17
Map 3#	Maximum iteration number	30	81	95	71	84
	Optimal path length	11.20	12.60	12.26	12.30	12.24

IV. CONCLUSION

In this paper, an improved particle swarm optimization algorithm based on improved inertia weights is adopted to solve the path planning problem. By setting up three different path planning maps, the particle swarm optimization algorithm based on five different inertia weight adjustment strategies is used to solve the typical path planning problem. The simulation results verify the effectiveness of the proposed algorithm and inertia weight adjustment strategies.

REFERENCES

[1] V. J. Lumelsky, and A. A. Stepanov, "Path-planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape," *Algorithmica*, vol. 2, no. 1-4, pp. 403-430, 1987.

[2] M. Liu, "Robotic Online Path Planning on Point Cloud," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1217-1228, 2015.

[3] Y. B. Chen, G. C. Luo, Y. S. Mei, J. Q. Yu, and X. L. Su, "UAV Path Planning Using Artificial Potential Field Method Updated by Optimal Control Theory," *International Journal of Systems Science*, vol. 47, no. 6, pp. 1407-1420, 2016.

[4] A. Alomari, F. Comeau, W. Phillips, and N. Aslam, "New Path Planning Model for Mobile Anchor-assisted Localization in Wireless Sensor Networks," *Wireless Networks*, vol. 24, no. 7, pp. 2589-2607, 2017.

[5] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control with Multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952-964, 2017.

[6] A. Abdullah, P. William, A. Nauman, and C. Frank, "Dynamic Fuzzy-logic Based Path Planning for Mobility-assisted Localization in Wireless Sensor Networks," *Sensors*, vol. 17, no. 8, pp. 1904, 2017.

[7] S. Jeong, O. Simeone, and J. Kang, "Mobile Edge Computing via a UAV-mounted Cloudlet: Optimization of Bit Allocation and Path Planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049-2063, 2018.

[8] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, and J. Ferruz, "Applying the MOVNS (Multi-objective Variable Neighborhood Search) Algorithm to Solve the Path Planning Problem in Mobile Robotics," *Expert Systems with Applications*, vol. 58, pp. 20-35, 2016.

[9] R. Kala, A. Shukla, and R. Tiwari, "Fusion of Probabilistic A* Algorithm and Fuzzy Inference System for Robotic Path Planning," *Artificial Intelligence Review*, vol. 33, no. 4, pp. 307-327, 2010.

[10] J. L. Solka, J. C. Perry, B. R. Poellinger, and G. W. Rogers, "Fast Computation of Optimal Paths Using a Parallel Dijkstra Algorithm with Embedded Constraints," *Neurocomputing*, vol. 8, no. 2, pp. 195-212, 1995.
[11] Z. Lin, J. Fu, H. Shen, and W. Gan, "Non-singular Tool Path Planning by Translating Tool Orientations in C-space," *International Journal of Advanced Manufacturing Technology*, vol. 71, no. 9-12, pp. 1835-1848, 2014.
[12] W. Shi, and W. Kai , "Floyd Algorithm for the Shortest Path Planning of Mobile Robot," *Chinese Journal of Scientific Instrument*, vol. 30, no. 10, pp. 2088-2092, 2009.
[13] E. Masehian, and M. R. Amin-Naseri, "A Voronoi Diagram-visibility Graph-potential Field Compound Algorithm for Robot Path Planning," *Journal of Field Robotics*, vol. 21, no. 6, pp. 275-300, 2004.
[14] 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.
[15] N. K. Yilmaz, C. Evangelinos, P. Lermusiaux, and N. M. Patrikalakis, "Path Planning of Autonomous Underwater Vehicles for Adaptive Sampling Using Mixed Integer Linear Programming," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 522-537, 2008.
[16] Z. Cai, and Z. Peng, "Cooperative Coevolutionary Adaptive Genetic Algorithm in Path Planning of Cooperative Multi-mobile Robot Systems," *Journal of Intelligent & Robotic Systems*, vol. 33, no. 1, pp. 61-71, 2002.
[17] J. W. Choi, R. E. Curry, and G. H. Elkaim, "Continuous Curvature Path Generation Based on Bézier Curves for Autonomous Vehicles," *IAENG International Journal of Applied Mathematics*, vol. 40, no. 2, pp. 91-101, 2010.
[18] K. K. Lim, Y. S. Ong, M. H. Lim, X. Chen, and A. Agarwal, "Hybrid Ant Colony Algorithms for Path Planning in Sparse Graphs," *Soft Computing*, vol. 12, no. 10, pp. 981-994, 2008.
[19] J. Kennedy, "Particle Swarm Optimization," *Springer U. S.*, 2011.
[20] B. Gordan, D. J. Armaghani, M. Hajihassani, and M. Monjezi, "Prediction of Seismic Slope Stability through Combination of Particle Swarm Optimization and Neural Network," *Engineering with Computers*, vol. 32, no. 1, pp. 85-97, 2016.
[21] B. Yao, B. Yu, P. Hu, J. Gao, and M. Zhang, "An Improved Particle Swarm Optimization for Carton Heterogeneous Vehicle Routing Problem with a Collection Depot," *Annals of Operations Research*, vol. 242, no. 2, pp. 303-320, 2016.
[22] R. P. Singh, V. Mukherjee, and S. P. Ghoshal, "Particle Swarm Optimization with an Aging Leader and Challengers Algorithm for the Solution of Optimal Power Flow Problem," *Applied Soft Computing*, vol. 40, pp. 161-177, 2016.
[23] Z. Chen, R. Xiong, and J. Cao, "Particle Swarm Optimization-based Optimal Power Management of Plug-in Hybrid Electric Vehicles Considering Uncertain Driving Conditions," *Energy*, vol. 96, pp. 197-208, 2016.
[24] P. Li, D. Xu, Z. Zhou, W. J. Lee, and B. Zhao, "Stochastic Optimal Operation of Microgrid Based on Chaotic Binary Particle Swarm Optimization," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 66-73, 2016.
[25] M. A. H. Akhand, S. Akter, M. A. Rashid, and S. B. Yaakob, "Velocity Tentative PSO: an Optimal Velocity Implementation Based Particle Swarm Optimization to Solve Traveling Salesman Problem," *IAENG International Journal of Computer Science*, vol. 42, no. 3, pp. 221-232, 2015.
[26] Y. Delice, E. K. Aydoğan, U. Özcan, and M. S. İlkay, "A Modified Particle Swarm Optimization Algorithm to Mixed-model Two-sided Assembly Line Balancing," *Journal of Intelligent Manufacturing*, vol. 28, no. 1, pp. 23-36, 2017.
[27] B. Haddar, M. Khemakhem, and C. Wilbaut, "A Hybrid Quantum Particle Swarm Optimization for the Multidimensional Knapsack Problem," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 1-13, 2016.
[28] J. J. Kim, and J. J. Lee, "Trajectory Optimization with Particle Swarm Optimization for Manipulator Motion Planning," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 620-631, 2015.
[29] Y. Shi, and R. C. Eberhart, "Empirical Study of Particle Swarm Optimization," *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1945-1950, July 1999.
[30] Y. L. Zheng, L. H. Ma, L. Y. Zhang, and J. X. Qian, "On the Convergence Analysis and Parameter Selection in Particle Swarm Optimization," *Proceedings of 2003 International Conference on Machine Learning and Cybernetics*, pp. 1802-1808, November 2003.
[31] M. Cui, and Q. B. Zhu, "Convergence Analysis and Parameter Selection in Particle Swarm Optimization," *Computer Engineering and Applications*, vol. 43, no. 23, pp. 89-91, 2007
[32] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing Hierarchical Particle Swarm Optimizer with Time-varying Acceleration Coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240-255, 2004.
[33] C. Y. Jiang, S. G. Zhao, S. G. Shen, and L. Z. Guo, "Particle Swarm Optimization Algorithm with Sinusoidal Changing Inertia Weight," *Computer Engineering and Applications*, vol. 48, no. 8, pp. 40-42, 2012.

Yi-Xuan Lu is an undergraduate student in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail: 1095545671@qq.com). Her main research interest is intelligent optimization algorithms.

Jie-Sheng Wang received his B. Sc. And M. Sc. degrees in control science from University of Science and Technology Liaoning, China in 1999 and 2002, respectively, and his Ph. D. degree in control science from Dalian University of Technology, China in 2006. He is currently a professor and Master's Supervisor in School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interest is modeling of complex industry process, intelligent control and Computer integrated manufacturing.

Sha-Sha Guo is received her B. Sc. degree from Liren college of yanshan University in 2018. She is currently a master student in School of Electronic and Information Engineering, University of Science and Technology Liaoning, China. Her main research interest is intelligent optimization algorithms.