



Intelligent Vehicle Global Path Planning Based on Improved Particle Swarm Optimization

Qiong Gao

College of Automotive Engineering, Shanghai University of Engineering Science, Shanghai, China

Email: sugo1993@163.com

How to cite this paper: Gao, Q. (2018) Intelligent Vehicle Global Path Planning Based on Improved Particle Swarm Optimization. *Open Access Library Journal*, 5: e4587.

<https://doi.org/10.4236/oalib.1104587>

Received: April 9, 2018

Accepted: April 24, 2018

Published: April 27, 2018

Copyright © 2018 by author and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Intelligent Transportation System (ITS) is able to reduce traffic jams and the incidence of accidents, and it is the current research hotspot. Intelligent vehicle is the key element of ITS, and path planning is one of the key technologies of intelligent vehicle. Particle Swarm Optimization (PSO) is a swarm-based intelligence algorithm. Using PSO for path planning can achieve good results. In order to improve PSO to get a better path planning result, a nonlinear inertia weight is proposed. The improved PSO and traditional PSO are compared in a global complex environment and the simulation results show that the improved PSO has a shorter path and better real-time performance.

Subject Areas

Mechanical Engineering

Keywords

Intelligent Transportation System, Path Planning, Particle Swarm Optimization, Nonlinear Inertia Weight

1. Introduction

Due to the complexity, randomness, multi-restraint, and multi-objective characteristics of intelligent vehicle path planning, the planning effect of traditional path planning methods such as Grid and Visibility Graph is not ideal. In recent years, with the continuous development of intelligent methods such as Particle Swarm Optimization (PSO), many researchers have adopted intelligent methods to study the path planning of intelligent vehicles and achieved good results.

Compared with other algorithms, PSO is widely used in path planning due to its fast convergence speed, low setting parameters, and simple implementation

[1]. Qin Yuanqing *et al.* [2] applied the PSO to the path planning of mobile robots for the first time. This method adopts the linkage graph method to carry on the environment modeling, uses the Dijkstra algorithm to obtain the shortest path of the link graph, then applies the PSO to carry on the second optimization to the path obtained by the link graph method. However, the usage of PSO for secondary optimization limits the global search ability of PSO. Sun Bo *et al.* [3] proposed a global path planning method for mobile robot based on PSO. This method first uses the coordinate transformation method to model the environment, encodes the path between the starting point and the ending point as particles, and then searches for the global optimal path based on the particle swarm algorithm. However, since the PSO uses linearly decreasing inertia weights, it cannot solve the premature phenomenon of the algorithm. Tang Biwei *et al.* [4] proposed a hybrid algorithm combining PSO with differential evolution algorithm for traditional PSO that cannot balance global and localization (exploration and exploitation) and easily trapped into stagnation. Nie Zhibin *et al.* [5] proposed the nonlinear inertial weight PSO to solve the problem of imbalance between exploration and exploitation, and combined simulated annealing algorithm to solve the problem of algorithm easily falling into the local minimum.

PSO has many shortcomings. How to optimize the parameters of the PSO to maximize the search efficiency is worth further investigation. A nonlinear inertia weight is proposed to improve the PSO in the following three ways: increase the planning success rate, decrease the path length and decrease the consumed time.

2. Environment Modeling

The methods of environmental modeling include the link graph method [6] [7], the visibility method [8] [9], the grid method [10] [11], etc. The most commonly used method is the grid method. However, a sharp increase in the number of grids in a complex environment will cause the problem of combinatorial explosion, and the choice of grid granularity will affect the accuracy of the algorithm's search. Sun Bo adopted a coordinate transformation environment modeling method, which has the advantages of simple modeling, clear physical meaning, and high path accuracy, and is suitable for path planning problems. This paper adopts the coordinate transformation method for environment modeling, in which the environment map simulates the structural environment of the urban community, and obstacles are all represented by rectangles.

As is shown in **Figure 1**, in the global coordinate system $O-X_1Y_1$, Start is the start point of the path, and End is the end point of the path. The black blocks in the figure represent obstacles. Let line of Start and End be X axis, and set the counterclockwise 90 degree direction as Y axis, thus established the local coordinate system Start-XY.

The angle between the $O-X_1$ axis and the Start-X axis is α . The obstacle information in the global coordinate system $O-X_1Y_1$ needs to be transformed into the local coordinate system Start-XY through the coordinate transformation. The

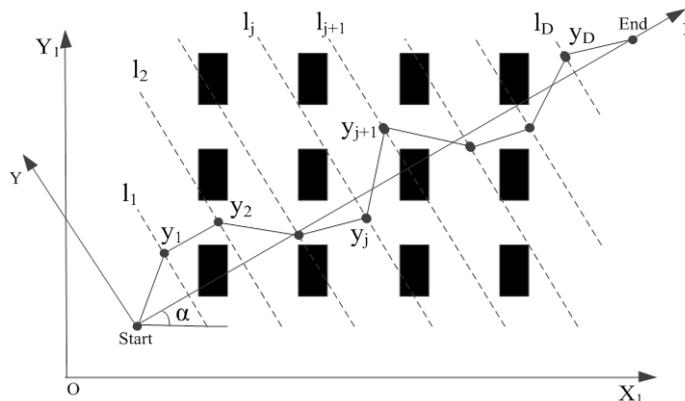


Figure 1. Environment modeling and path generation.

coordinate transformation formula is shown below.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} x_{1start} \\ y_{1start} \end{bmatrix} \quad (1)$$

In the local coordinate system Start-XY, the line Start-End is equally divided into $D+1$ segments. D is the predetermined parameter, which is related to the complexity of the environment. Draw vertical lines at each of these points, and get parallel line cluster $(l_1, l_2, \dots, l_j, l_{j+1}, \dots, l_D)$. The intersection of this cluster and the path Line (i) is the target points sequence $(y_{i,1}, y_{i,2}, \dots, y_{i,j}, y_{i,j+1}, \dots, y_{i,D})$.

After the modeling is finished, the path planning problem is transformed into a constrained optimization problem. The objective function is to minimize the length of the path and the constraint is that any adjacent two points in the point set Line(i) are line connection and they do not intersect with obstacles. The point set Line(i) is represented as $\{\text{Start}, p_{i,1}, \dots, p_{i,j}, \dots, p_{i,D}, \text{End}\}$, Where $p_{i,j} = (x_{i,j}, y_{i,j})$.

3. Global Path Planning Based on PSO

3.1. Particle Swarm Optimization

Kennedy and Eberhart proposed PSO in 1995 [12]. In PSO, each particle adjusts its path according to its own flight experience and its companions' flight experience. The best position that each particle has experienced is personal best value (Pbest). The best position that the whole group has experienced is global best value (Gbest). Each particle keeps updating itself through these two best values. The merit of the particle is evaluated by Fitness Value which is determined by the optimization problem. The PSO adopts the speed-position ($v-x$) search model, and each particle has a speed to determine the direction and the size of the update. If there are N particles in the population and each particle has n dimensions, its velocity and location are updated in Equation (2) and (3) as follow. Where $v_{i,j}^{k+1}$ is the speed of particle i in j^{th} dimension in $(k+1)^{\text{th}}$ iteration and $x_{i,j}^{k+1}$ is the location of particle i in j^{th} dimension in $(k+1)^{\text{th}}$ iteration. And r_1

and r_2 are the random number within the interval of [0,1]. ω is inertial weight, c_1 and c_2 are the learning factors.

$$v_{i,j}^{k+1} = \omega * v_{i,j}^k + c_1 * r_1 * (Pbest_{i,j}^k - x_{i,j}^k) + c_2 * r_2 * (Gbest_j^k - x_{i,j}^k) \quad (2)$$

$$x_{i,j}^{k+1} = x_{i,j}^k + v_{i,j}^{k+1} \quad (3)$$

3.2. Improved Particle Swarm Optimization

Shi Y first introduced the inertia weight into the particle swarm algorithm [13], and analyzed that a large inertia weight is favorable for global search, while a smaller inertia weight is more favorable for local search. In order to better balance the algorithm's global search and local search capabilities, Shi. Y proposed Linear Decreasing Inertia Weight (LDIW), as shown in Equation (4).

$$\omega(k) = \omega_{start} - (\omega_{start} - \omega_{end}) \left(\frac{k}{T_{max}} \right) \quad (4)$$

where, ω_{start} is the initial value of the inertia weight; ω_{end} is the final value of the inertia weight; k is the current number of iterations; T_{max} is the maximum number of iterations. In general, the inertia weight $\omega_{start} = 0.9$, $\omega_{end} = 0.4$ can reach the best performance of the algorithm. In this way, as the iteration progresses, the inertia weight decreases linearly from 0.9 to 0.4. The larger inertia weight at the beginning of the iteration keeps the algorithm's global search ability strong, and the smaller inertia weight at the later iterations helps the algorithm to be more accurate.

In order to better balance the global searching ability of the PSO in the initial iterations and the local optimization ability in the later iterations, the Nonlinear Decreasing Inertial Weight (NDIW) is proposed in Equation (5). **Figure 2** shows the inertia weight curves of the two algorithms.

$$\omega(k) = \omega_{end} + (\omega_{start} - \omega_{end}) e^{-\left(\frac{ck}{T_{max}}\right)^2} \quad (5)$$

where $c = 2.3$.

3.3. Simulation

In order to validate the effectiveness of the proposed algorithm, the path planning simulation is carried out in a complex environment. Initializing 30 particles, the initialization path generated is shown in **Figure 3(a)**. Set the maximum iteration times T as 100, the final path after iterations is shown in **Figure 3(b)**. **Figure 3(c)** is the fitness value evolution curve. It can be seen that the algorithm has been completely convergent, and 30 particles have found the global optimal value. When the maximum number of iterations is reached, the consumed time of the algorithm is 0.71 s, and the optimal path length is 1429.71 m. Particles generally start to converge around 50 generations.

In order to illustrate the superiority of the improved algorithm, the comparison simulation was conducted between the improved algorithm (in order to differentiate, the improved PSO algorithm which has NDIW is referred to as IPSO)

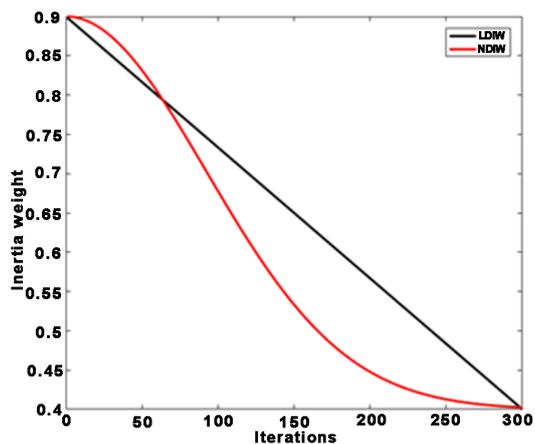
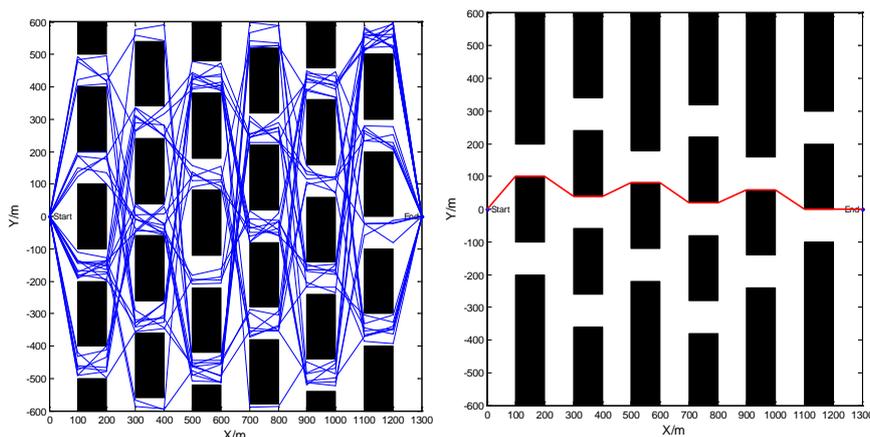
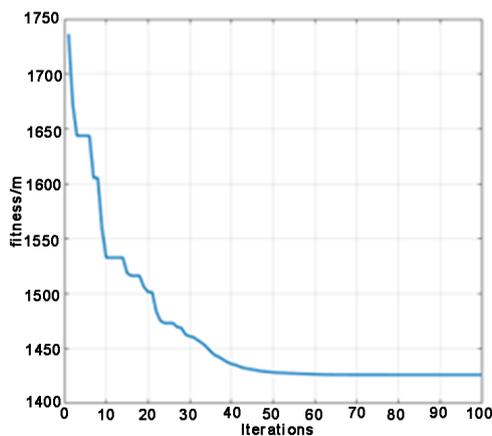


Figure 2. Changing curves of inertia weight for two algorithms.



(a)



(c)

Figure 3. Simulation result of the proposed algorithm. (a) Initialization path; (b) Path after iterations; (c) Fitness value evolution curve.

and traditional PSO algorithm (abbreviation for TPSO which has LDIW). **Figure 4(a)** is the final path comparison curve of the two algorithms. **Figure 4(b)**

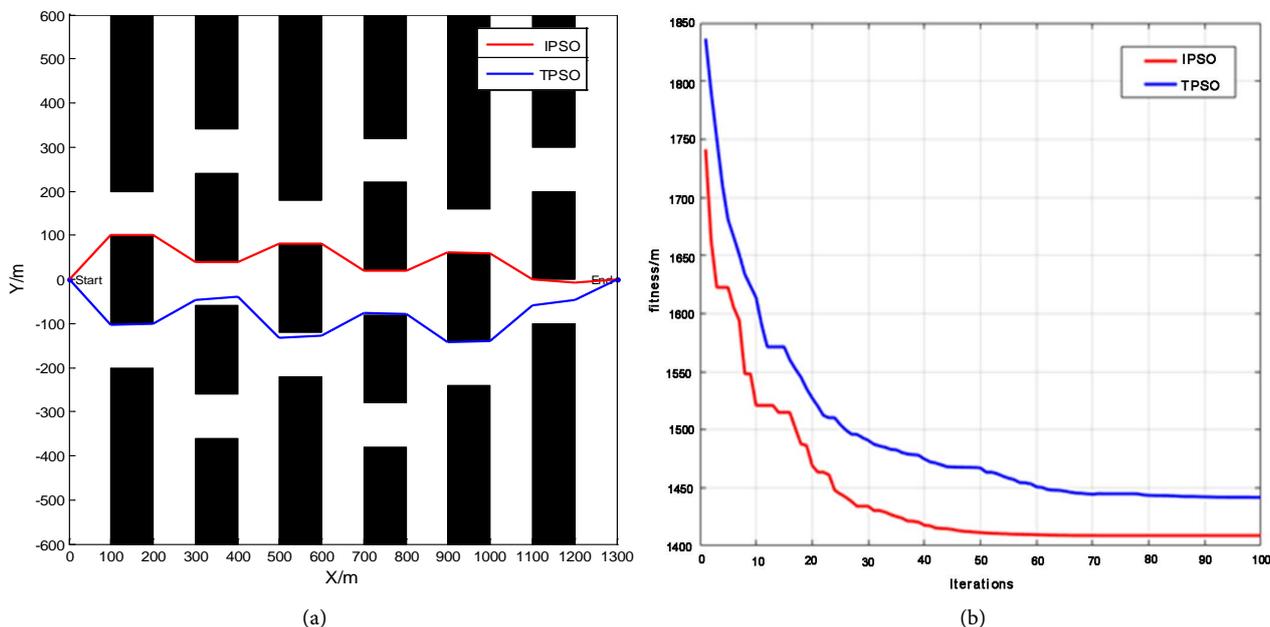


Figure 4. Simulation result of comparison experiment. (a) Final path comparison curve; (b) Fitness value comparison curve.

Table 1. The comparison of simulation result.

Algorithm	Path Length	Consumed Time	Iterations to Reach the Optimal Value	Planning Success Rate
IPSO	1409.13	0.71	48	96%
TPSO	1446.75	0.89	42	84%

shows the fitness value comparison curve of the two algorithms.

The two algorithms are simulated in the same complex environment 50 times respectively, and the average value is calculated and shown in **Table 1**.

Figure 4 and **Table 1** show that the length of path generated from the proposed algorithm is reduced about 37.6 m, and planning success rate is increased by 12%. The consumed time of the algorithm decreases by 0.18 s. The real-time performance of the algorithm is improved.

4. Conclusions

Research on path planning of intelligent vehicle based on improved PSO was conducted in this paper. Simulation results show that the improved algorithm can successfully seek a safe path from the start point to the endpoint. The main conclusions of this paper are as follows:

- 1) The planning success rate of the proposed algorithm is 96%, which increased a lot to the traditional PSO, so the improved algorithm can effectively improve the planning success rate.
- 2) The path length of the proposed algorithm is 1409.13 m, which is shorter than the traditional PSO, so the improved algorithm can get better path.
- 3) The consumed time of the proposed algorithm is 0.71 s, so the real-time performance of the improved algorithm is better.

Acknowledgements

The author expresses his sincere and heartfelt thanks to the editor and reviewers for their constructive suggestions to improve the quality of this paper. The author also expresses his sincere and heartfelt thanks to mentors who have provided help in the process of writing this paper. The author declares that there are no conflicts of interest regarding this work.

References

- [1] Roberge, V., Tarbouchi, M. and Labonte, G. (2012) Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. *IEEE Transactions on Industrial Informatics*, **9**, 132-141. <https://doi.org/10.1109/TII.2012.2198665>
- [2] Qin, Y.Q., Sun, D.B., Li, N., *et al.* (2004) Path Planning for Mobile Robot Based on Particle Swarm Optimization. *Robot*, **26**, 222-225.
- [3] Sun, B., Chen, W.D. and Xi, Y.G. (2005) Particle Swarm Optimization Based Global Path Planning for Mobile Robots. *Control and Decision*, **20**, 1052-1060.
- [4] Tang, B.W., Zhu, Z.X. and Luo, J.J. (2016) Hybridizing Particle Swarm Optimization and Differential Evolution for the Mobile Robot Global Path Planning. *International Journal of Advanced Robotic Systems*, **13**, 1-17. <https://doi.org/10.5772/63812>
- [5] Nie, Z.B., Yang, X.B., Gao, S.H., *et al.* (2016) Research on Autonomous Moving Robot Path Planning Based on Improved Particle Swarm Optimization. *Evolutionary Computation*, **2016**, 2532-2536.
- [6] Guo, J.C., Gao, Y. and Cui, G.Z. (2015) The Path Planning for Mobile Robot Based on Bat Algorithm. *International Journal of Automation and Control*, **9**, 50-60. <https://doi.org/10.1504/IJAAC.2015.068041>
- [7] Tan, G.Z., He, H. and Sloman, A. (2006) Global Optimal Path Planning for Mobile Robot Based on Improved Dijkstra Algorithm and Ant System Algorithm. *Journal of Central South University of Technology*, **13**, 80-86. <https://doi.org/10.1007/s11771-006-0111-8>
- [8] Xu, S.J. and Cao, Q.Y. (2011) A Visibility Graph Based Path Planning Algorithm for Mobile Robot. *Computer Applications and Software*, **28**, 220-222.
- [9] Chen, C., Tang, J., Jin, Z.G., *et al.* (2014) A Path Planning Algorithm for Seeing Eye Robots Based on V-Graph. *Mechanical Science and Technology for Aerospace Engineering*, **33**, 490-495.
- [10] Hsu, C.C., Chen, Y.J., Lu, M.C., *et al.* (2012) Optimal Path Planning Incorporating Global and Local Search for Mobile Robots. *Consumer Electronics*, Tokyo, 2-5 October 2012, 668-671. <https://doi.org/10.1109/GCCE.2012.6379947>
- [11] Yoon, S., Yoon, S.E., Lee, U., *et al.* (2015) Recursive Path Planning Using Reduced States for Car-Like Vehicles on Grid Maps. *IEEE Transactions on Intelligent Transportation Systems*, **16**, 2797-2813. <https://doi.org/10.1109/TITS.2015.2422991>
- [12] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. *Proceedings of the IEEE International Conference on Neural Networks*, **4**, 1942-1948. <https://doi.org/10.1109/ICNN.1995.488968>
- [13] Shi, Y.H. and Eberhart, R. (1998) A Modified Particle Swarm Optimizer. *Proceedings of the IEEE Conference on Evolutionary Computation*, Anchorage, 4-9 May 1998, 69-73. <https://doi.org/10.1109/ICEC.1998.699146>