A Hybrid Path Planning Algorithm Based on Simulated Annealing Particle Swarm for The Self-driving Car

Jinkai Yin, Weiping Fu*
Mechanical and Precision Instrument Engineering
Xi'an University of Technology
Xi'an, China

Jinkai Yin
College of Applied Engineering
Henan University of Science and Technology
Sanmenxia, Chin
yjk516@163.com

Abstract—**In order to improve the safety of the self-driving cars for the purpose of using the particle swarm algorithm alone in path planning of the self-driving cars, a hybrid path planning algorithm based on simulated annealing algorithm and particle swarm optimization is proposed. The hybrid optimization algorithm keeps the PSO algorithm simple and easy to implement, improves the global optimization ability of the algorithm, and improves the convergence speed and calculation accuracy of the algorithm. Simulation experimental results show that the algorithm has better global optimization ability and can provide guarantee for solving the global path planning problem of the self-driving cars.**

Keywords—Hybrid algorithm, Path planning, Simulated annealing, Particle swarm

I. Introduction

With the advancement of technology, the self-driving cars have gradually entered the field of vision. Path planning is one of the key technologies of the self-driving cars. Its core technology is Path Planning Algorithm (PPA) [1]. At present, ant colony algorithm, genetic algorithm, artificial potential field method, etc. are commonly used methods of path planning. These methods have their own advantages, but there are also some limitations, such as high computational complexity, poor convergence, easy to fall into local. The shortcomings of optimality and so on make path planning limited to a certain degree [2].

Particle Swarm Optimization (PSO) is a swarm intelligence optimization algorithm proposed by Eberhart and Kennedy in 1995. The algorithm retains the global search strategy of the population. It can achieve dynamic tracking and real-time adjustment of the search through the speed-displacement model and its own memory characteristics. Strategies and other functions have the advantages of simple algorithm design, easy implementation, fewer parameters to be adjusted, fast convergence speed, and strong global search ability. Once they were proposed, they attracted the attention and research of scholars. PSO has been applied to many fields because of its simple implementation and effective solution to problems. However, in some practical applications, some aspects are still unsatisfactory. The main problem is that it is prone to premature convergence, and it is easy to fall into local optimums. In response to these problems, scholars have proposed many improved PSO algorithms. Fan Ming et al. [4] established a linkage relationship between the inertia weight coefficients, the cognitive coefficient and the contraction factor, and introduced the crossover and mutation operations of the genetic algorithm. The particle swarm algorithm is improved, and the improved algorithm is used in the intelligent access system to solve the path optimization problem of the access system. Wang Bo et al. [5] improved the particle swarm algorithm by introducing crossover and mutation operations of the genetic algorithm. The improved particle swarm algorithm was used in cloud computing task scheduling. The feasibility and effectiveness of the algorithm were verified by simulation tests. Qiao Junfei et al. [6] proposed an improved dynamic adaptive particle swarm optimization algorithm to solve the water supply network optimization problem. Simulation results show that the improved algorithm has strong global search capability and fast convergence speed.

For the global path planning of the self-driving cars, this paper uses the simulated annealing algorithm [7] to update the position and velocity of the particles, calculate the particle fitness before and after the update, and the particle's fitness according to the Metropolis criteria to accept the optimization solution. The probabilistic acceptance of the worsening solution enables the PSO algorithm to jump out of the local extreme region and converge to the global optimal solution. With the characteristics of high convergence and easily out of local optima, the simulated annealing particle swarm algorithm (SAPSO) can prevent premature population loss and falling into the local optimal solution, effectively solving the global path planning of the self-driving cars.

II. Environmental Modeling and Problem Description

In order to describe the problem of the self-driving car global path planning, this paper first carries out environment map modeling and sets the motion environment of the self-driving cars. The settings are as follows: 1) The motion environment of the self-driving car is a two-dimensional space; 2) Some static obstacles are distributed in the motion space of the self-driving car, and the position and shape of the obstacle are all known; 3) To prevent the generation of the self-driving cars and obstacle edges in collisions, the obstacles are inflated and the self-driving cars are regarded as particles.

A. Behavioral Dynamics

The dynamic model of the car is a highly complex nonlinear structure. In order to facilitate the understanding of the basic characteristics of car handling and stability, in actual research, the mathematical model is generally simplified according to the actual situation. The driver controls the heading angle of the car by inputting the steering wheel angle and analyzes the response characteristics of the heading angle of the car output at the input of the steering wheel angle. Through certain assumptions on the car model, the car model is supported by two elastic tires on the ground. A two-degree-of-freedom car calculation model with lateral and yaw motions. Figure 1 shows a two-degree-of-freedom car model.

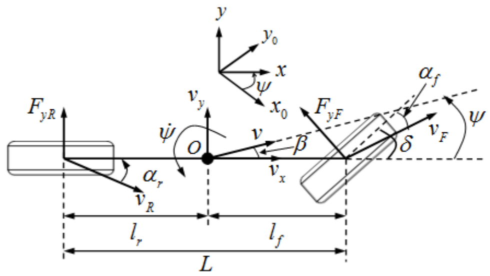

Figure 1 Two-degree-of-freedom car model

In Figure 1, O represents the center of mass of the car; v represents the speed, v_x , v_y represent the horizontal and vertical speed of the car respectively; ψ represents the heading angle of the car; l_f , l_r represent the distance between the front and rear wheels and the center of mass of the car respectively; δ represents the steering wheel angle; β represents the lateral declination angle of the car's centroid; α_f , α_r represent the tire front-rear tilt angle respectively; F_{yF} , F_{yR} represent the lateral force of the front and rear tires of the car respectively.

Taking the location of the center of mass of the car as the origin of the world coordinate system and combining the two-degree-of-freedom model of the car, the dynamic equation of the body movement is established.

$$mv(\dot{\beta}+\dot{\psi}) = C_f(\delta-\beta-\frac{l_f}{v}\dot{\psi}) + C_r(\frac{l_r}{v}\dot{\psi}-\beta) \tag{1}$$

$$I_z\ddot{\psi} = C_f(\delta-\beta-\frac{l_f}{v}\dot{\psi})l_f - C_r(\frac{l_r}{v}\dot{\psi}-\beta)l_r \tag{2}$$

In Equation (1) and (2), m is the quality of the car; I_z is the yaw inertia of the car; C_f , C_r are the equivalent cornering stiffness of the front and rear wheels of the car respectively; $\dot{\psi}$, $\ddot{\psi}$ are the heading angle (yaw angle) velocity and angular acceleration.

B. Environmental modeling

For the self-driving cars, path planning is the set of collisionless points that pass through the motion environment [8], which is shown in Figure 2.

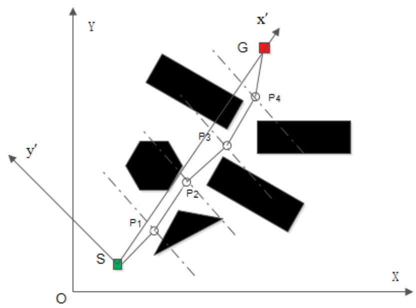

Figure 2 Motion Environment and Path Generation Process

First, build the global coordinate system XOY in which the sports environment is located. Black solid objects represent obstacles. $S(x_s, y_s)$ represents the starting point, $G(x_g, y_g)$ represents the target point, path planning is to find the set of the collisionless points in XOY $P=\{S,P_1,P_2,\cdots,P_i,G\}$, $i=1,2,\cdots,d$, and the path should meet the shortest length at the same time. Where, $(p_1,p_2,\cdots,p_m)$ is the planning goal and the requirement is a non-obstacle point, and the connection of adjacent p_i does not pass through obstacle. In order to facilitate the calculation, the global coordinate system needs to be transformed. The corresponding coordinate transformation relationship [9] is shown in Equation (3).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_s \\ y_s \end{bmatrix} \tag{3}$$

Where, any point (x, y) in the XOY coordinate system corresponds to point (x', y') in the $SX'Y'$ coordinate system, θ is the angle between the axis X and X' ,divide line SG segment $(m+1)$, the division point is the target point sequence $(p_1,p_2,\cdots,p_m)$, bisection length is $(L_1,L_2,\cdots,L_m)$. S is defined as p_0 , G is defined as p_{m+1} , The length LP of the path SG is represented by the (x', y') coordinate, shown in Equation (2).

$$L_P = \sum_{j=0}^{m}\sqrt{(x'p_j - x'p_{j+1})^2 + (y'p_j - y'p_{j+1})^2} \tag{4}$$

Intelligent car path planning under static environment should meet the shortest path length and can successfully avoid static obstacles. This paper is to solve the shortest path problem under the constraints of obstacles. If there is a path that does not collide with an obstacle, the shorter the path length, the better the path. The self-driving car optimal path planning is transformed to optimize the function of Equation (4), find the minimum L_P in the value space of $p_j(j=1,2,\cdots,m)$ in the $SX'Y'$ coordinate system. The smaller the individual fitness function value, the path satisfies the requirement.

C. Obstacle avoidance conditions

In order to meet the obstacle avoidance requirements of path planning, the following constraints need to be taken. First, set the range of the particle motion position, and only the movement between the specified upper and lower limits, and the objects that cannot be touched during driving (such as lane lines, guardrails), cars or pedestrians are defined as obstacles; at the same time, according to traffic rules, the solid line is not allowed to cross the road, the dotted line can be crossed, but it cannot be driven for a long time, and the real lane line is defined as a strong constraint obstacle. The lane line is defined as a weak constraint obstacle. Strong restraint obstacles must be used as obstacles to avoid obstacles while the car is in motion. Weakly constrained obstacles must not be used as obstacle avoidance when they need to change lanes or overtaking during driving. At that time, for the sake of driving safety, keep the car and the lane line at a safe distance and treat it as obstacle avoidance. If the position of the particle coincides with the obstacle, search again; secondly, if the line between p_k and p_{k+1} is connected with the obstacle If there is a crossover, it indicates that there is a possibility of collision between the self-driving car and the obstacle. Then the planned path is the wrong path. Search for p_{k+1} again until you find a collisionless path that meets the constraints.

III. Particle Swarm Optimization Algorithm with Mixed Simulated Annealing

PSO has larger inertia weight and higher temperature in the early stage of search, which makes SAPSO algorithm has better global optimization ability. As the search progresses, the inertia weight coefficient and temperature gradually decrease, and the SAPSO algorithm gradually locates the "optimal search" near the optimal solution. The PSO algorithm clusters the particles to the global optimal position p_g , and SA guarantees the diversity of the population, and prevents the PSO algorithm from becoming premature and falling into a local minimum by giving the search process a time-varying and eventually zero probability jump. Therefore, the hybrid algorithm has strong global and local search capabilities.

The specific SAPSO algorithm flow is described [10-13].

(a) Randomly initialize the position and velocity of each particle in the population;

(b) Calculate the target value of each particle; store the best position and target value of each particle in the individual extreme p_i, and store the position and target value of the optimal particle in the population in p_g ;

(c) Using Equation (5) to determine the initial temperature;

$$t_0 = f(p_g)/\ln 5 \tag{5}$$

(d) Use Equation (6) to update the velocity of each particle, and update the position of each particle with Equation (7);

$$v_{ij}^{k+1} = \chi[v_{ij}^k + c_1 r_1 (p_{ij}^k - x_{ij}^k) + c_2 r_2 ((p_{gj}^k)' - k_{ij}^k)] \tag{6}$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad j = 1, 2, \cdots, d \tag{7}$$

In Equation (6), j is the current number of iterations; c_1 and c_2 are the learning factor; r_1 and r_2 are random number distributed in the $[0,1]$ interval; V is the speed of the particle; X is the particle position. φ is inertia weight, $\varphi = 2 \Big/ \left| 2 - C - \sqrt{C^2 - 4C} \right|$,where $C = c_1 + c_2$.

(e) Calculate the position of each particle and its target value, update the p_i and its target value of each particle, update the population p_g and its target value;

(f) The annealing operation is performed and the annealing method is performed according to Equation (8). If the termination condition is satisfied, the operation accuracy or number of iterations reaches a predetermined value, the search stops, and the result is output; otherwise, the process goes to Step (b).

$$t_{k+1} = \lambda t_k \tag{8}$$

In Equation (8), λ is an annealing coefficient.

IV. Simulation and Analysis

The computer used in the simulation experiment was configured as a memory 64G, the CPU was an E5-2630 server, and the simulation software was MATLAB 2014a. Set the map for the size 180×180 of the self-driving car's sports environment, $(15,15)$ represents the starting point, $(140,140)$ represents the target point, dimension $d = 19$, the maximum number of iterations $I_{\max} = 20$, the annealing initial temperature $T_{start} = 100$, the lowest temperature $T_{end} = 0$. In order to facilitate the comparison, the PSO algorithm, the simulated annealing-teaching optimization (SATLBO) algorithm and the SAPSO algorithm proposed in this paper were compared in the same environment. All algorithms have a population size of 25, a search space dimension of 19, and an iteration count of 20, where the PSO and SAPSO algorithm parameters are: Learning factor $c_1 = c_2 = 1.5$, Inertia weight $w = 0.7$; The TLBO algorithm has no parameters. After several algorithms are separately executed, the optimal path map of each algorithm shown in Figure 3-5 and the comparison curve of convergence curves shown in Figure 6 are obtained.

From Figure 4 to Figure 7, it can be seen that the path obtained by the PSO algorithm is poor, the path obtained by the SATLBO and SAPSO algorithm is good, the convergence rate of the SAPSO and SATLBO algorithms is fast, and the SAPSO algorithm can improve the search accuracy while achieving fast convergence of the ideal optimal path. The late local search ability of PSO algorithm is poor, and no better path can be found after tens of iterations. Because the particles in the PSO algorithm can easily fly from the feasible area to the infeasible area in the search process, resulting in a large number of particles in the population after the constraint processing to change the direction so as not to search for the optimal location in the optimal direction, and ultimately affect the search effect. The SATLBO and SAPSO algorithms can achieve a very good optimization path after a small number of iterations and have high accuracy.

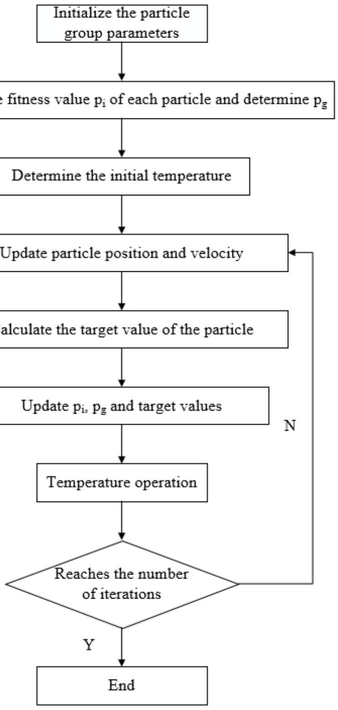

Figure 3. The flow of SAPSO algorithm

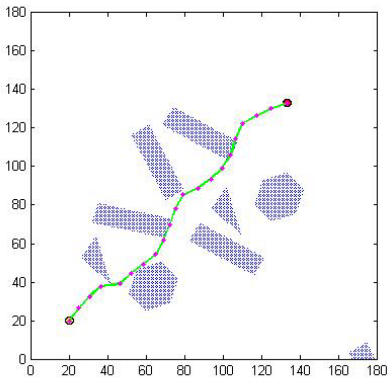

Figure 4. PSO

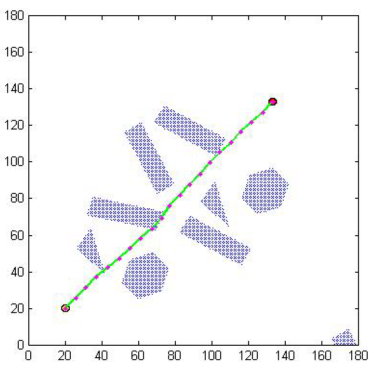

Figure 5. SATLBO

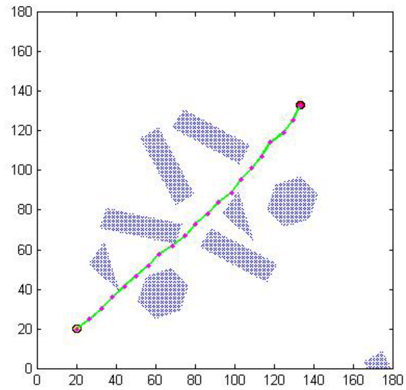

Figure 6. SAPSO

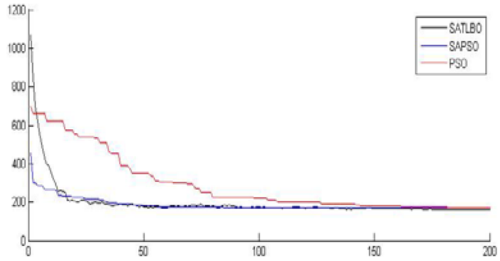

Figure 7. Convergence curve comparison chart

V. Conclusion

This paper uses SAPSO to solve the problem of the optimal path of the self-driving cars， and achieves good results. Particle Swarm Optimization (PSO) is a new swarm intelligence optimization algorithm that can solve the continuous space optimization model. It has a simple algorithm, no parameters, and has a faster convergence speed and higher search accuracy. The self-driving car global path planning method based on Simulated Annealing Particle Swarm Optimization (SAPSO) algorithm is used to create a new environment map through environment map modeling and coordinate transformation, and then SAPSO algorithm is used to find an ideal global optimal path. The results show that this algorithm has obvious advantages compared with particle swarm optimization, artificial bee colony algorithm and differential evolution algorithm when it is used to optimize the global path planning of the self-driving cars. Simulation experiments show that this method is simple in modeling, low in complexity, easy to implement, fast in convergence and high in search accuracy. In subsequent studies, we can consider combining this method with the rolling window method to solve the problem of local dynamic path planning. In this paper, the kinematics of the car is not considered, and the environmental parameters of typical driving behavior are relatively simple. In the next step of parking, starting and design of complex traffic environment, optimization and improvement are made.

Acknowledgment

This work is supported by Project (10872160) of the National Natural Science Foundation of China, Shaanxi Province Important Disciplines of the Car Engineering Construction of China, Scientific Research Projects of Shaanxi Province Education Office Key Laboratory (13JS070), and Scientific Research Project in Shaanxi Province Department of Education (14jk1796).

REFERENCES

[1] Vincent Roberge, Mohammed Tarbouchi, Giilles Labonte. Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning[J]. IEEE Transactions on Industrial Informatics, 2013, 9(1): 132-141.

[2] WANG Hui, ZHU Long-biao. Research on path planning of parking system based on PSO-genetic hybrid algorithm[J]. Chinese Journal of Engineering Design,2016,23(2):195-200.

[3] WEI Jing-xuan. Evolutionary Algorithms for Single-Objective and Multi-Objective Optimization Problems [D]. XIAN: XIDIAN UNOVERSITY.2009.

[4] FAN Ming, GUO Yi, YUN Chao. Adaptive hybrid algorithm for dynamic path planning problem of intelligent access system[J]. Journal of system simulation, 2013,25(7): 1543-1548.

[5] WANG Bo, ZHANG Xiao-lei. Task scheduling algorithm based on particle swarm optimization genetic algorithms in cloud computing environment[J]. Computer Engineering and Applications, 2015,51(6):84-88.

[6] QIAO Jun-fei, WANG Chao, LIU Chang-fen. Optimal design of a water supply system based on improved self-adaptive particle swarm algorithm[J]. Journal of Beijing University of Technology, 2014,40(7):1035-1040.

[7] ZHAO Tian-tian, WANG Si-ming. Path planning of mobile robot based on improved PSO algorithm[J]. Transducer and Microsystem Technologies,2018,37(2):57-60.

[8] Wu Zong-sheng, Fu Wei-ping. SA and Teaching-learning-based Optimization Algorithm for Mobile Robots Global Path Planning [J].Mechanical Science and Technology for Aerospace Engineering, 2016 35(5):678-685.

[9] QIANG Ning, Gao Jie, Kang Feng-ju. Multi-Robots Global Path Planning Based on PSO Algorithm and Cubic Spline[J]. Journal of System Simulation,2017,29(7):1397-1404.

[10] YAN Jing-yu, LI Chong-guo, QIAN Hui-huan. Multi-objective Parameters Optimization of Electric Assist Control Strategy for Parallel Hybrid Electric Car[C]. IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2009:1992-1997.

[11] B Zhang, Z Chen, C Mi, LM Yi. Multi-objective Parameter Optimization of a Series Hybrid Electric Car Using Evolutionary Algorithms [C]. IEEE Car Power & Propulsion Conference,2009:921-925.

[12] ZHENG Jiachun, WU Jianhua, MA Yong. A Hybrid optimization algorithm of Simulated anealing and Particle Swarm for Unmanned Surface Vessel Path Planning[J]. Periodical of Ocean University of China, 2016,46(9):116-122