Dynamic Path Planning for Cooperative Robots Based on the Hybrid SA-PSO Algorithm

Qian Wang, Hong Li

Guangzhou Railway Vocational and Technical College, Guangzhou, Guangdong Province, China

Abstract:

Addressing the issues of particle swarm optimization (PSO) in robot path planning, such as getting trapped in local optima, improper parameter settings, and low computational efficiency, this paper proposes a novel hybrid optimization strategy that combines PSO with simulated annealing (SA). This hybrid algorithm integrates the advantages of both algorithms, leveraging the rapid convergence ability of PSO and the global search capability of SA to provide a more efficient, flexible, and robust solution. It is particularly suitable for complex, dynamic, and multi-objective scenarios and can be used to solve the path planning problem of cooperative robots in complex and changing environments. Through simulation experiments, this article has verified the effectiveness of the algorithm. The results show that, compared to traditional PSO and SA algorithms, this algorithm demonstrates higher performance and efficiency in planning the trajectory of a robot in a complex dynamic environment.

Keywords: Simulated annealing, cooperative robots, dynamic path planning.

INTRODUCTION

In recent years, cooperative robots were increasingly used in various fields, particularly in warehousing and logistics, smart manufacturing,[1] smart cities,[2] medical services,[3] and smart agriculture.[4] Path planning, as a core component of cooperative robot technology, is crucial for enhancing the overall performance of robotic systems. However, in dynamic and complex environments, path planning for cooperative robots faces numerous challenges, such as real-time requirements, multi-objective optimization, dynamic changes in the environment, and collision avoidance among robots. Currently, path planning has become one of the key technologies for achieving autonomous navigation of robots and is also one of the important research topics in the field of artificial intelligence and robotics.[5-7]

various intelligent algorithms, represented by Particle Swarm Optimization (PSO),[8] have been applied to robot path planning to seek more efficient and robust solutions. However, the PSO algorithm also suffers from issues such as easily falling into local optima and low search accuracy.To solve the above problems, many researchers have tried to optimize the algorithm,For example, using PSO, DE algorithm to smooth the trajectory while improving the efficiency of the robot.[9] Although existing research has made significant progress in cooperative robot path planning, most existing methods are primarily designed for static or quasi-static environments and have poor adaptability to dynamically changing environments, making it difficult to adjust path planning strategies in real-time.

To address these deficiencies, this article proposes a dynamic trajectory planning method for collaborative robots based on the hybrid SA-PSO algorithm. This method aims to improve the real-time performance, accuracy, and adaptability of trajectory planning by combining the global search capabilities of the SA algorithm with the rapid convergence characteristics of the PSO algorithm. As a result, it provides strong support for the efficient use of collaborative robots in dynamic and complex environments, surpassing the limitations of the traditional Particle Swarm Optimization (PSO) algorithm.

TRADITIONAL PARTICLE SWARM OPTIMIZATION ALGORITHM

Basic Principles

The basic idea of the particle swarm optimization algorithm is to treat solutions in the search space as a group of "particles," where each particle represents a potential solution. The update of a particle is influenced by its personal historical best position and personal cognition (individual extreme pbest) as well as the historical best position and social cognition (global extreme gbest) of the entire swarm. In this algorithm, the velocity update of a particle follows the following formula:

$$\nu_{i,k}(n+1) = \omega \cdot \nu_{i,k}(n) + c1 \cdot r1 \cdot (pbest_{i,k} - x_{i,k}(n)) + c2 \cdot r2 \cdot (gbest_k - x_{i,k}(n)) \quad (1)$$

$\nu_{i,k}(n)$ represents the velocity of particle i in the k-th dimension during the n-th iteration;ω is the inertia weight, which controls the degree of preservation of the particle's initial direction of motion; c1 and c2 are the learning coefficients, corresponding to the particle's self-awareness (personal best) and social awareness (global best) respectively; r1 and r2 are random numbers between 0 and 1, used to introduce an element of randomness; $pbest_{i,k}$ is the position of particle i's personal best in the k-th dimension; $gbest_k$ is the position of the group's global best in the k-th dimension; xi,k is the position of particle i in the k-th dimension during the n-th iteration..

The position update formula is as follows:

$$si,(n+1)=si,(n)vi,k(n+1) \tag{2}$$

Here, si, k(n+1) represents the new position of particle i in the k-th dimension in the next iteration, and vi, k(n+1) is the new velocity calculated through the velocity update formula. The inertia weight ω is usually set higher at the beginning of the algorithm to facilitate global search, typically within the range of [0.4, 0.9]. A smaller ω is more beneficial for local search, while a larger ω value is more conducive to global search. It gradually decreases as the number of iterations increases, promoting local search.

Current Status of PSO Algorithm Application

The Particle Swarm Optimization (PSO) algorithm is capable of global search, helping to avoid local optimal solutions, which is particularly important for robot path planning as it can attempt to find the globally optimal path for all robots to work together. It possesses excellent parallel processing capabilities. Essentially, the PSO algorithm is parallel, enabling it to efficiently handle path planning problems in multi-robot systems, where each robot or path can be represented by a particle and optimized simultaneously. The PSO algorithm can adapt to different environments and task requirements by adjusting parameters, making it more flexible when dealing with multi-robot path planning in dynamic or uncertain environments. The implementation of the PSO algorithm is relatively simple and easy to integrate with existing robot control systems.

However, traditional methods based on Particle Swarm Optimization also have some inherent limitations. For example, they are prone to getting trapped in local optima. Although the PSO algorithm has global search capabilities, in some cases, especially in high-dimensional spaces or under multiple constraints, it may still converge too early to local optimal solutions. The performance of the algorithm heavily relies on the choice of initialization parameters such as inertia weight and acceleration constants. Improper parameter settings may lead to slow convergence or poor results. Computational resource consumption: For large-scale multi-robot systems, the computational load of the PSO algorithm can become very large, especially when real-time planning is required, which may limit its efficiency in practical applications. Additionally, the convergence and stability of the PSO algorithm are not fully proven theoretically, which may cause some issues in certain industrial applications with strict requirements.

IMPROVED PSO-SA ALGORITHM

Basic Idea of the Improved PSO-SA Algorithm

This paper intends to develop a hybrid algorithm, leveraging the rapid convergence characteristics of PSO and the global search capability of SA to achieve a better balance in optimization problems, avoid falling into local optimal solutions, improve the execution efficiency and computational accuracy of the PSO algorithm, and assist the algorithm in discovering the global optimal solution more rapidly during the search process.

Random Acceptance Mechanism of SA

Initialize the particle swarm, setting initial positions and velocities for each particle. Set the initial temperature T0, cooling rate, and the number of internal iterations n for simulated annealing, and update the positions and velocities of the particles using the standard PSO algorithm.

The particle swarm guides its next evolutionary path based on the individual best solutions of each particle and the global best solution. However, when gbest is a local optimum, all particles are influenced and move towards the local optimum, leading to rapid convergence of the particle swarm and the emergence of local extrema or stagnation.[10] Therefore, during the velocity update process of the PSO algorithm, the disturbance mechanism of the simulated annealing algorithm[11] is introduced to assign a certain jump probability to the currently suboptimal particles. Individuals with fitness closer to the current optimal particle have a correspondingly higher probability of jumping. When such a situation occurs, we choose to replace the existing optimal particle to reduce the risk of the algorithm falling into a local optimal solution. This strategy helps improve global search capabilities and ensures that we can explore the optimal solution in a broader solution space.

For each particle, the fitness value of its new position is first calculated. If the fitness of the new position is better or equal, the position is accepted directly; otherwise, a decision on whether to accept the new position is made based on a certain acceptance probability. The formula is as follows:

$$P = e^{-\frac{\Delta E}{T}} \tag{3}$$

Here, ΔE = f(xi(t+1)) - f(xi(t)) represents the energy difference, and T is the current temperature. If the randomly generated number is less than p, the new position is accepted; otherwise, the old position is retained.

Update the temperature according to the cooling rate α.

$$T_{new} = T \cdot \alpha \tag{4}$$

Repeat steps 2 to 4 until a termination condition is met, such as reaching the maximum number of iterations or when the temperature drops below a certain threshold. In the aforementioned steps, the choice of T0, α, and n has a significant impact on the algorithm's performance and needs to be adjusted based on the specific problem. Initially, you can try setting a larger initial T, a higher α (e.g., 0.95), and a moderate n (e.g., 100 to 500 iterations). Monitor the convergence speed of the algorithm and the quality of the planned path, and adjust the parameters accordingly. Consider using adaptive strategies to adjust T and α, such as dynamically adjusting the decay rate of T based on the improvement magnitude of the current iteration, which can enhance the algorithm's flexibility and efficiency. In practical applications, further adjustments to the above process may be necessary, such as alternating between PSO and SA or introducing SA mechanisms into certain stages of PSO. The key to combining PSO with SA lies in balancing their contributions, ensuring both the efficiency of global search and avoiding premature convergence to local optima. In practical applications, fine-tuning of the algorithm may also be required to adapt to the characteristics of specific problems.

The flowchart of the improved PSO-SA algorithm is shown in Figure 1.

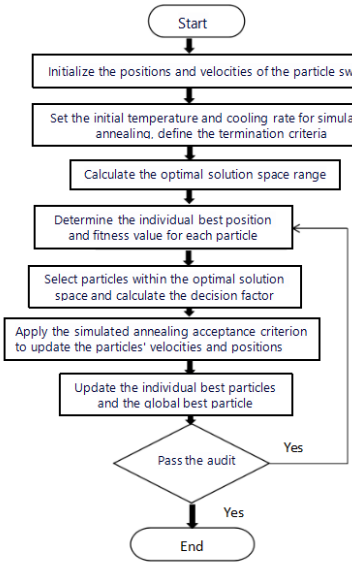

Figure 1. Flow chart of the algorithm

EXPERIMENTS RESULTS

Experimental Environment and Parameters

To verify the effectiveness of the proposed algorithm, we designed simulation experiments with a few and many obstacles in a static environment. The algorithm was run on a WINDOWS 10 64-bit system using MATLAB R2018a, and all simulations were conducted using MATLAB language.

The population size is set to 200 particles, with a maximum iteration limit of 500.To prevent collisions, the coefficient of the penalty function is set to 1000, learning factors (r1, r2) = 1.5, 1.5, initial inertia weight (w) = 1, inertia weight damping factor (wd) = 0.98; for simulated annealing (SA) parameters, initial temperature (T_initial) = 1000, cooling rate = 0.99, and final temperature (T_final) = 1.

Simulation Results

To erify the effectiveness of the virtual target heuristic function, experiments were conducted using the Particle Swarm Optimization (PSO) algorithm and the proposed algorithm in a semi-enclosed map. The resulting path planning diagrams are shown in Figures 2 and 3, respectively.

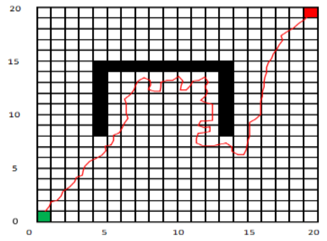

Figure 2. Path planning map of the semi-surrounded topographic particle swarm algorithm

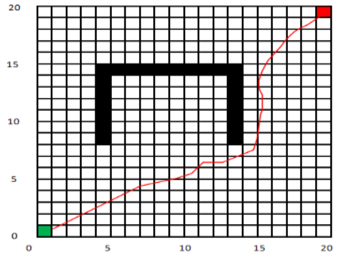

Figure 3. The algorithm path planning map of this paper

In the above figure, the robot will move from the marked point at the lower left corner to the marked point at the upper right corner. The research findings indicate that when using the algorithm proposed in this study for path planning, the robot demonstrates faster response times and higher efficiency in obstacle avoidance, successfully escaping semi-enclosed terrain environments while adopting a more optimized travel route. Compared to the traditional particle swarm algorithm, this method significantly reduces the path length to reach the destination.

Method Comparison

To verify the advantages and effectiveness of the algorithm proposed in this paper, this study conducted 50 sets of simulation experiments for the proposed algorithm, the classic particle swarm algorithm, the simulated annealing algorithm, and the differential evolution (DE) algorithm in the same map environment, focusing on scenarios with dense obstacles. In data processing, the maximum and minimum path lengths were excluded, and subsequently, the average value and the mean squared error of the remaining data were calculated to assess the stability of these four algorithms in terms of performance. The data is presented in Table 1.

Table 1. Comparison of Three Methods

Method	Average Optimal Fitness	Standard Deviation	Average Iteration Times to Convergence
Algorithm in this paper (PSO)	1.4e-06	3.5e-07	620
Standard PSO	3.2e-04	9.8e-05	850
Simulated Annealing (SA)	5.6e-04	1.7e-03	900
Differential Evolution (DE)	2.2e-05	6.3e-06	700

From the above data, we can observe that the algorithm in this paper achieves convergence with fewer average iterations than standard PSO, indicating its potential superiority in search efficiency. The proposed algorithm attains the lowest average optimal fitness, indicating that it finds solutions that are closer to the global optimum compared to standard PSO and SA, which produce higher fitness values suggesting less optimal solutions. By evaluating stability using standard deviation, the proposed algorithm shows a lower variance, emphasizing its consistent performance across multiple trials. Therefore, it converges faster, identifies solutions that are closer to being optimal, and exhibits greater stability.

CONCLUSION

This paper focuses on the path planning algorithm for collaborative robots and conducts a series of research. By combining PSO and SA algorithms, a new hybrid optimization strategy, the SA-PSO algorithm, is successfully developed. This algorithm enables faster escape from local optima and addresses path planning issues for collaborative robots in complex dynamic environments. The dynamic path planning strategy based on the algorithm in this paper can perceive environmental changes in real-time and quickly adjust strategies to adapt to dynamic and complex environments, improving the flexibility and adaptability of collaborative robots in practical applications. This dynamic path planning method optimizes paths for multiple robots simultaneously, achieving effective collision avoidance and collaboration among robots. To enhance the stability and robustness of the algorithm, additional in-depth research is necessary for achieving more refined parameter tuning. Furthermore, the dynamic environments addressed in this paper are primarily limited to simple, changing scenarios; future studies could extend the algorithm's applicability to more complex dynamic settings. Meanwhile, exploring more efficient and intelligent path planning methods by combining advanced technologies such as deep learning and reinforcement learning will bring new breakthroughs and opportunities to the development of collaborative robots.

REFERENCES

[1] Bai Y, Lv Y, Zhang J, et al. Smart mobile robot fleet management based on hierarchical multi-agent deep Q network towards intelligent manufacturing. Engineering Applications of Artificial Intelligence, 2023, 124: 106534.

[2] Islam N, Haseeb K, Almogren A, et al. A framework for topological based map building: A solution to autonomous robot navigation in smart cities. Future Generation Computer Systems, 2020, 111: 644-653.

[3] Fan Y, Hu Y, Jiang L, et al. Intelligent disinfection robots assist medical institutions in controlling environmental surface disinfection. Intelligent Medicine, 2021, 1(1): 19-23.

[4] Zheng L, Zhang X, Wang J, et al. Path planning of field robot based on macro-micro combination. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(9): 13-26. (Translated from Chinese)

[5] Mao J, He Z, Wang Y, et al. Review on path planning technology and application of power inspection robots. Control and Decision, 2023, 38(11): 3009-3024. (Translated from Chinese)

[6] Shi K, Wu P, Liu M. Research on path planning method of forging handling robot based on combined strategy // IEEE International Conference on Power Electronics, Computer Applications, 2021: 292-295.

[7] Shi Q, Niu W, Zhang R, et al. Review and prospect of path planning for underwater gliders. Chinese Journal of Ship Research, 2023, 18(1): 29-51. (Translated from Chinese)

[8] Ju Q, Zhu J. Multi-objective flexible job-shop scheduling problem for batch production. Journal of Mechanical Engineering, 2007, 43(8): 148-154. (Translated from Chinese

[9] Li Z, Du X, Li O. Dynamic integrated optimization of process planning and scheduling in PCB assembly workshop. Beijing: Science Press, 2013. (Translated from Chinese)

[10] Zou H, Zhou H. Time-optimal trajectory planning for robots based on improved particle swarm optimization. Manufacturing Automation, 2023, 45(01): 107-110. (Translated from Chinese)

[11] Fan Y, Shen K, Wang D, et al. Optimal energy consumption path planning for UAV mountain operation based on simulated annealing algorithm. Journal of Agricultural Machinery, 2020, 51(10): 34-41. (Translated from Chinese)