Journal of Artificial Intelligence Practice (2023)
Clausius Scientific Press, Canada
DOI: 10.23977/jaip.2023.060705
ISSN 2371-8412 Vol. 6 Num. 7

Research on Robot Path Planning Based on Simulated Annealing Algorithm

Wencheng Wang[1,2,3,*], Pengcheng Zhang[1,2,3], Hongrun Wang[4]

[1]*Department of Mechanical Engineering, Hebei University of Water Resources and Electric Engineering, Changzhou, 061000, China*
[2]*Industrial Manipulator Control and Reliability Technology Innovation Center of Hebei, Cangzhou, Hebei, 061001, China*
[3]*Industrial Manipulator Control and Reliability Technology Innovation Center of Cangzhou, Cangzhou, Hebei, 061001, China*
[4]*State Grid Corporation of China Hebei Electric Power Co., Ltd. Nanpi County Power Supply Subsidiary, Cangzhou, Hebei, 061001, China*
[*]*Corresponding author*

Keywords: Simulated annealing algorithm, path planning, MATLAB

Abstract: Taking the path planning of inspection robots as the research object, a shortest path planning method based on simulated annealing algorithm was proposed. By analyzing the conditions of the shortest path generation, the mathematical solution model of the problem was established, and the global search strategy was formulated. Finally, the shortest path of the robot was solved through MATLAB software programming. The correctness and effectiveness of the algorithm are verified by a large number of examples. In addition, in view of the low efficiency of the traditional simulated annealing algorithm to solve the large-scale shortest path problem, the output result is unstable, and the path is easy to cross, a new solution is constructed in the form of random coordinate exchange. The results show that the shortest path output of this method not only has less crossover, but also the operation efficiency is obviously improved, and the result is more stable.

1. Introduction

The robot path planning problem is mainly used to help robots find an optimal motion path from a starting point to a goal point in a complex environment, allowing the robot to efficiently accomplish predefined tasks during the motion. Path planning techniques have important applications and great research value in the fields of artificial intelligence, game design, and traffic route navigation [1, 2]. The simulated annealing algorithm is a stochastic search algorithm with good convergence properties, particularly suitable for handling complex combinatorial optimization problems. Its application scope has surpassed traditional search algorithms, and with the rapid development of computer technology, it can effectively handle complex nonlinear problems and has been successfully applied in various fields [3].

In recent years, with the rapid development of technology, obtaining optimal solutions has become an important issue in various fields. From traditional graph theory to advanced computer models and algorithms, as well as the integration of various application technologies, all provide possibilities for solving this complex challenge. As a shortest path problem, it is no longer just a problem of finding the shortest path between two points. Many optimization problems in different industries can be solved by constructing mathematical models and transforming them into shortest path problems. In this process, the development of computers and algorithms has provided a more diverse range of solutions for the shortest path problem, which has been widely covered in many fields such as computer science, transportation, communication, and operations research. The simulated annealing algorithm is one of the most effective methods for effectively solving such shortest path planning problems.

Robot path planning can be divided into global path planning and local path planning, and the environment can be further divided into static and dynamic environments. Therefore, any path planning problem can be subdivided into the following categories: global static environment path planning, global dynamic environment path planning, local static environment path planning, and local dynamic environment path planning.

2. Simulated Annealing Algorithm

In 1983, S. Kirkpatrick et al. successfully introduced the annealing concept to the field of combinatorial optimization [4]. It is a random optimization algorithm based on the Monte Carlo iterative strategy. It originated from the similarity between the annealing process of solid materials in physics and general combinatorial optimization problems. It aims to enhance the thermal stability of simulated materials and then gradually reduce the temperature until reaching a relatively stable state, which can be achieved through simulating the annealing of materials. With the continuous evolution of the simulated annealing algorithm, the Monte Carlo method has been successfully used to simulate the equilibrium state of solids under stable temperature conditions. The advantage of this approach is its simplicity, but it also has some limitations. For example, it requires large-scale data sampling, which leads to increased computational complexity. The Metropolis criteria, proposed in 1953, use importance sampling and multiple iterations to improve reliability and achieve higher system efficiency [5]. Its basic principle is to continuously adjust parameters to reach a higher level of stability.

Today, through the relentless efforts of experts and scholars, the simulated annealing algorithm has developed various forms, including adaptive simulated annealing, genetic-simulated annealing, etc., making it play a greater role in the field of combinatorial optimization [6].

3. Problem Models and Solution Principles

3.1 Methods for Solving Path Planning Problems

Let the initial point be labeled as 1, and the target points are labeled sequentially as 2, 3, ..., n, and $n+1$, and point 1 is defined as the initial point for the sake of convenience. The distance matrix $D=(d_{ij})_{n\times n}$, where d_{ij} represents the distance between points i and j, $d_{1,2}=\sqrt{(x\mathbf{2}-x\mathbf{1})^2+(y\mathbf{2}-y\mathbf{1})^2}$, where i, j = 1, 2, n. D is a real symmetric matrix. The problem can be described as finding a shortest path starting from point 1, visiting all intermediate points, and reaching point n.

The simulated annealing algorithm for solving is described as follows:

(1) Solution Space

The solution space S can be represented as the set of all cyclic permutations of $\{1,2,...,n,n+1\}$ with fixed starting and ending points, i.e.,

$S = \{(p_1,...,p_n) \mid p_1=1, (p_2,...,p_n)$ is a cyclic permutation of $\{2,3,...,n\}, p_{n+1} = n+1\}$.

Each cyclic permutation represents a loop that traverses n targets, where $p_i = j$ indicates visiting point j at the i-th step. The initial solution can be chosen as $(1,2,...,n)$.

(2) Objective Function

The objective function in this case is the length of the path that traverses all the targets, given by:

$$\min f(p_1,p_2,...,p_{n+1})=\sum_{i=1}^{n} d_{p_i,p_{i+1}} .$$

(3) Generation of new solutions

1) 2-opt move

Choose indices u and v ($u < v$) and swap the order of the target points between u and v. The new path is: $p_1\ldots p_{u-1}p_v p_{v+1}\ldots p_{u+1}p_u p_{v+1}\ldots p_{n+1}$.

2) 3-opt move

Choose indices u, v, and w ($u < v < w$) and insert the path between u and v after w. The corresponding new path is: $p_1\ldots p_{u-1}p_{v+1}\ldots p_w p_u\ldots p_v p_{w+1}\ldots p_{n+1}$.

(4) Cost function difference

For the 2-opt move, the difference in path can be represented as:

$\Delta f=(d p_{u-1} p v+d p_u p_{v+1})-(d p_{u-1} p_u+d p_v p_{v+1})$.

(5) Acceptance criterion

$$P=\begin{cases} 1 & \Delta f < 0 \\ \exp(-\Delta f / T) & \Delta f \geq 0 \end{cases}$$

If $\Delta f < 0$, then accept the new path. Otherwise, accept the new path with a probability $exp(-\Delta f / T)$, if $exp(-\Delta f / T)$ is greater than the random number between 0 and 1.

(6) Cooling

Apply the selected cooling coefficient α to reduce the temperature, i.e.,$T \leftarrow \alpha T$, to obtain a new temperature. Here, α is chosen as 0.999.

(7) Termination condition

Use a selected termination temperature e=10~30 to determine if the annealing process is finished. If $T<e$, the algorithm terminates and outputs the current state.

3.2 The flow of the simulated annealing algorithm

The flowchart of the simulated annealing algorithm is as follows:

Step 1: Set the initial temperature T_0 and randomly select a traversal path $P(i)$ as the initial path. Calculate its length $L(P(i))$.

Step 2: Generate a new traversal path $P(i+1)$ randomly and calculate its length $L(P(i+1))$.

Step 3: If $L(P(i+1))<L(P(i))$, accept $P(i+1)$ as the new path. Otherwise, accept $P(i+1)$ with the probability determined by the simulated annealing criteria. Then, decrease the temperature.

Step 4: Repeat steps 1 and 2 until the temperature reaches the minimum value *Tmin*.

4. Solving the Shortest Path Planning Problem

4.1 Description of the Practical Problem

Taking the substation inspection robot as an example, the inspection points are represented in a two-dimensional coordinate system. Establishing a two-dimensional coordinate system, set the starting point of the inspection robot in the Cartesian coordinates as: [565, 575]. Set the coordinates of 51 inspection points in the Cartesian coordinate system as follows: [25, 185], [345, 750], [945, 685], [845, 655], [880, 660], [25, 230], [525, 1000], [580, 1175], [650, 1130], [1605, 620], [1220, 580], [1465, 200], [1530, 5], [845, 680], [725, 370], [145, 665], [415, 635], [510, 875], [560, 365], [300, 465], [520, 585], [480, 415], [835, 625], [975, 580], [1215, 245], [1320, 315], [1250, 400], [660, 180], [410, 250], [420, 555], [575, 665], [1150, 1160], [700, 580], [685, 595], [685, 610], [770, 610], [795, 645], [720, 635], [760, 650], [475, 960], [95, 260], [875, 920], [700, 500], [555, 815], [830, 485], [1170, 65], [830, 610], [605, 625], [595, 360], [1340, 725], [1740, 245]. By using MATLAB for programming and design, compute the shortest path length and plot the inspection robot's inspection path.

4.2 Simulated Annealing Algorithm for Program Design

4.2.1 Initialization Parameter Settings

Set the initial temperature and the termination temperature T. The initial temperature T controls the direction of annealing. Set the decay parameter, where the commonly used cooling function is $T=T\times K$, and the decay parameter K is usually close to 1. Set the length of the Markov chain, denoted as L, which represents the number of iterations performed at a given temperature.

4.2.2 Problem Constraints

There are two constraints for finding the shortest inspection path for the robot during its operation. The first constraint is to visit all target points. Based on the practical problem, it means inspecting all the points in the substation. The second constraint is to return to the starting point after visiting all target points, which means the inspection robot should return to the initial point in the practical problem.

4.2.3 Program Implementation

First, use the start function to set the starting point as [565, 575] and record the coordinates of the 51 target points in an array. Set the initial temperature T = 97, the termination temperature T_f = 0.001, the length of the Markov chain L = 3000, and the decay parameter K = 0.99. After setting the first constraint, exit the loop when the temperature value T is less than or equal to T_f. Inside the while loop, there is a nested for loop for multiple iterations with perturbation, meaning conducting multiple experiments before the temperature decreases. The loop iterates from 1 to L, where L is the length of the Markov chain. Design the len1 function to calculate the distance of the route. Then, using a random function, generate the coordinates of two random target points and perform a two-point swap in the route (exchange two targets). Record the difference in route length as delta_e. If delta_e < 0, replace the old route with the new route. If delta_e > 0, accept the new solution with a certain probability.

4.2.4 Program Output Results

Using MATLAB programming to implement the above process, the output results are shown in Figure 1 and Figure 2. Figure 1 represents the generated shortest working path for the robot, indicating the sequential order in which the robot visits all target points during its work. Figure 2 shows the fitness curve, indicating the relationship between the optimization process of the objective function and the number of iterations during the computation.

In addition, different test cases have been conducted to test the algorithm and program presented in this paper, all of which have obtained the optimal path for the problem. For example, Figure 3 represents the optimal path for 30 targets, Figure 4 represents the optimal path for 100 targets.

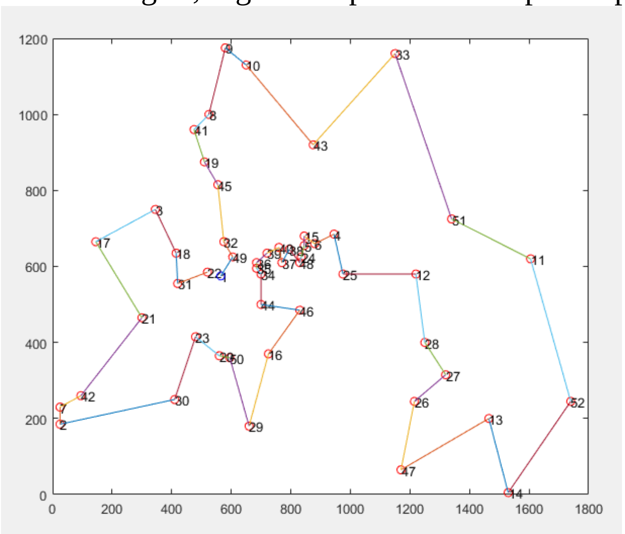

Figure 1: Shortest path visiting all target points

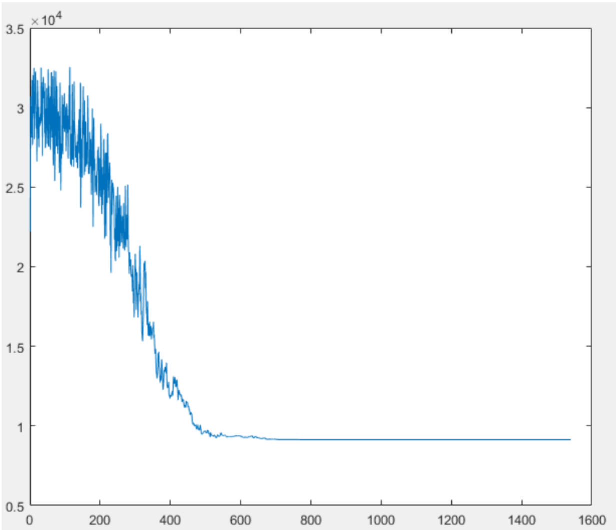

Figure 2: Fitness evolution curve

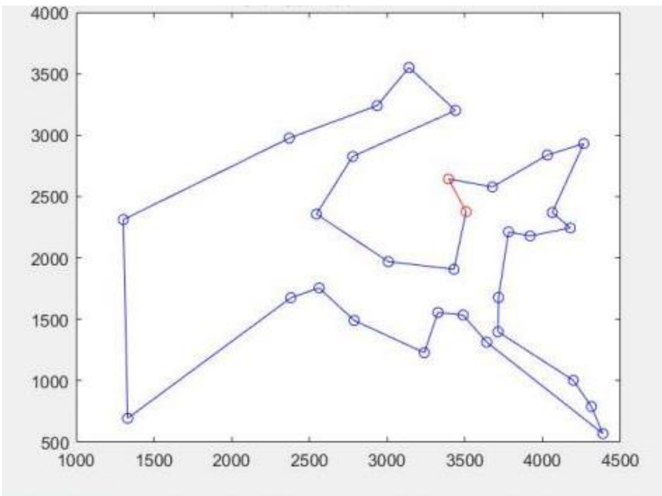

Figure 3: Shortest path for 30 target points

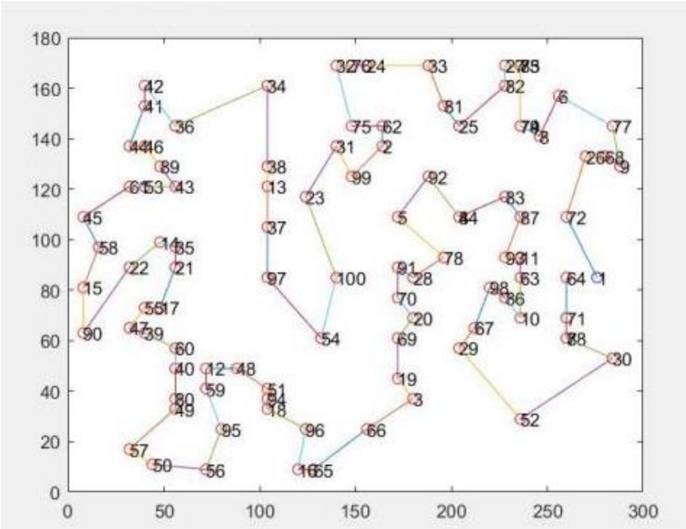

Figure 4: Shortest path for 100 target points

5. Impact of Initialization Parameters on Optimization Results

To analyze the impact of program parameters, each parameter needs to be analyzed individually. For example, when analyzing the effect of the initial temperature parameter T, keep the remaining three parameters (final temperature T_f, Markov chain length L, and decay parameter K) fixed.

5.1 Impact of Initial Temperature *T*

Assuming $T_f = 0.001$, $L = 3000$, and $K = 0.99$. Set the initial temperature T successively to 100, 200, 300, 400, and 500. Run each value three times and analyze the shortest distance in the output results. And Table 1 lists the data for the shortest distance obtained for each run of the program under different initial temperature conditions.

Table 1: Impact of initial temperature on results.

Initial temperature value	First run result	Second run result	Third run result
100	8691	9426	10438
200	8741	9144	8936
300	9864	9197	9042
400	8889	8624	8747
500	8426	8756	8650

The experimental results show that the initial temperature T plays a decisive role in the algorithm, directly controlling the direction of annealing. According to the acceptance criterion of random movement, the higher the initial temperature, the greater the probability of obtaining high-quality solutions.

5.2 Impact of Markov Chain Length *L* on Optimization Results

Assuming $T = 100$, $T_f = 0.001$, and $K = 0.99$. Set the Markov chain length successively to 100, 200, 300, 400, and 500. Run each value three times and analyze the shortest distance in the output results. And Table 2 lists the data for the shortest distance obtained for each run of the program under different Markov Chain length conditions.

Table 2: Impact of Markov Chain length on results

Markov Chain length	First run result	Second run result	Third run result
100	8691	9426	10438
200	9426	9500	9395
300	8345	8696	9901
400	8969	9355	8569
500	7756	9744	9000

Based on the analysis of the experimental data, it can be observed that the Markov chain length represents the number of iterations for iterative optimization at a constant temperature. The selection principle for Markov chain length, under the premise of a predetermined attenuation parameter, should ensure that the control parameters can restore quasi-equilibrium at each value. The longer the Markov chain length within the specified range, the better the generated solutions.

5.3 Impact of Termination Temperature T_f on Optimization Results

Assuming T =100, L = 3000, and K = 0.99. Set the termination temperature T_f successively as 1, 5, 10, 15, and 20. Run each value three times and analyze the shortest distance in the output results. And Table 3 lists the data for the shortest distance obtained for each run of the program under different termination temperature conditions.

Table 3: Impact of termination temperature on results

Termination Temperature	First Run Result	Second run result	Third run result
1	8692	9370	9669
5	8750	9562	9482
10	8898	9124	9332
15	9036	9925	9601
20	9064	9852	10082

Based on the experimental results, it can be observed that the lower the termination temperature, the higher the quality of the generated solution. Therefore, the termination temperature should be set as low as possible to obtain better optimal solutions.

5.4 Impact of Attenuation Parameter K on Optimization Results

Assuming T = 100, T_f = 0.001, and L = 300. Set the temperature attenuation parameter successively as 0.99, 0.95, and 0.90. Run each value three times and analyze the output results. And Table 4 lists the data for the shortest distance obtained for each run of the program under different temperature attenuation parameter conditions.

Table 4: Results of temperature attenuation parameter

Temperature attenuation parameter	First run result	Second run result	Third run result
0.99	8691	9426	10438
0.95	11266	11230	9764
0.90	10268	10980	10672

Based on the analysis, the attenuation parameter, which is the temperature update function, is used to modify the temperature value in the outer loop. Currently, the commonly used temperature update function is exponential cooling, i.e., $T(n+1) = k \times T(n)$, where $0<K<1$ and K is a constant that is very close to 1. From the results, it can be observed that setting the temperature attenuation parameter closer to 1 yields better solution quality.

6. Conclusion

This study focused on the shortest path in the motion process of a substation inspection robot. The conditions for generating the shortest path were analyzed, and a mathematical problem-solving model was established. The simulated annealing algorithm was adopted, and a global search strategy was formulated. The MATLAB software was used for programming, and the correctness and effectiveness of the algorithm were verified through examples, achieving the solution of the shortest path. During the solving process, a combination of two-coordinate exchanges and three-coordinate exchanges was used to further expand the solution space, resulting in shorter optimal paths, more stable results, and avoiding the problem of intersections in the traditional method's route map, significantly improving the computational efficiency.

Acknowledgements

The authors gratefully acknowledge the financial support from “Science and technology research project of colleges and universities in Hebei Province”: Design and experimental study of automatic surface garbage cleaning robot suitable for small water areas (ZD2021329).

References

[1] Fan Qi, Cai Yuqiang, Jia Sinan (2022). Multi-method Industrial Robot Trajectory Planning and Simulation [J]. Journal of North China University of Science and Technology(Natur-al Science Edition), 44(04):60-67.

[2] Zhou Qijun (2022). Research on industrial robot trajectory planning and control system[D]. Tianjin:Tianjin University of Technology.

[3] Kang Wenxuan (2022). City delivery route planning for shared bikes based on simulated annealing algorithm[J]. Science and Innovation, 13:104-106+109.

[4] Yuan Jiaquan, Guo Jian (2019). Robot path planning method based on simulated annealing ant colony algorithm[J]. Path Planning, 36(10):329-333.

[5] Chen Yao (2015). Design and implementation of global path planning for intelligent inspection robots in substations[D]. Jinan: Shandong University.

[6] Vadim Indelman (2018). Cooperative multi-robot belief space planning for autonomous navigation in unknown environments. [J]. Auton. Robots, 42(2):353-373.