ISSN 2616-5775 Vol. 6, Issue 2: 28-35, DOI: 10.25236/AJCIS.2023.060205

A new multi-objective point path planning algorithm for mobile robots

Zeping Yang*, Xiangguo Sun

College of Mechanical Engineering, Sichuan University of Science & Engineering, Sichuan, China
**Corresponding author*
2521348347@qq.com

Abstract: *Aiming at the path planning problem of mobile robot passing through multiple target points with the shortest path under two-dimensional raster map, this paper proposes a method combining an improved A* algorithm with an annealing genetic algorithm. For the problem that A* algorithm will search adjacent nodes in turn, Jump Point Search (JPS) algorithm is introduced to avoid the search of surrounding nodes by searching the jump point of the current node and reduce the calculation time. In view of the prematurity problem of genetic algorithm, the simulated annealing algorithm is introduced to make the simulated annealing algorithm and genetic algorithm complement each other and overcome the prematurity problem of genetic algorithm effectively. By using TSPLIB dataset, the path planning of the improved genetic algorithm is obviously better than that of the original genetic algorithm. Finally, the feasibility and effectiveness of the method are verified by simulation, and the shortest safe path can be found quickly in the process of optimization.*

Keywords: *path planning; A* algorithm; multi-objective point; genetic algorithm*

1. Introduction

Multi-objective point path planning of a mobile robot is to find an optimal path in the environment that passes through all target points and is safe and collision free[1]. Most of the path planning of conventional mobile robots takes a single target point as the optimization target, and there are few researches on the path planning of multi-target points. Reference[2] proposes an improved bidirectional search A* algorithm, which indeed speeds up the operation efficiency of the algorithm, but increases the requirements for running memory, and doesn't take into account the path planning of multi-objective points; Reference [3] compares the genetic algorithm with the ant colony algorithm in terms of scale, time, correctness and stability. The accuracy of genetic algorithm (GA) is higher than that of Ant Colony algorithm (ACO), and the stability is also higher. Reference [4] proposes a combination algorithm that uses hierarchical Particle Swarm Optimization (PSO) -genetic algorithm to first calculate the order of target points, and then uses A* algorithm to carry out path planning according to the order of target execution. However, this algorithm cannot guarantee that the global path planning is optimal.

An improved A* algorithm combined with simulated annealing genetic algorithm is proposed to solve the multi-objective path planning problem of the mobile robot. The path planning of multi-target points is divided into two steps. Firstly, JPS algorithm is introduced to optimize A* algorithm to calculate the shortest path between any two target points, and then simulated anneal-genetic algorithm is used to calculate the global optimal path of multi-target points. The combined algorithm avoids the search of surrounding nodes and reduces the calculation time. For the premature problem of genetic algorithm, simulated annealing algorithm is introduced to effectively overcome the premature problem of genetic algorithm.

2. Planning of the shortest path

2.1. The original A algorithm*

In the path planning of grid maps, the A* algorithm[5] is one of the common shortest path planning algorithms. It is a heuristic search algorithm. The specific process is to start from the current starting point, search for the movable nodes around, calculate the cost from the node to the starting point and the target point, select the node with the lowest cost, and update the status of the node until the target point

ISSN 2616-5775 Vol. 6, Issue 2: 28-35, DOI: 10.25236/AJCIS.2023.060205

is searched. The set of nodes with the minimum cost is the optimal path. The evaluation function of its heuristic search is:

$$f(n) = g(n) + h(n) \tag{1}$$

Where $f(n)$ is the evaluation function of node n, $g(n)$ is the actual cost from the starting point to n, and $h(n)$ is the evaluation cost of the best path from node to target point.

2.2. Integrating JPS algorithm to improve the efficiency of path planning

JPS algorithm[6][7] takes A* algorithm as the framework, and further optimizes the operation of A* algorithm to find successor nodes. JPS expands the successor node based on the direction of the current node and the strategy of jumping points. Definition of jump point: A) X is a jump point if X is a starting point or a destination point. B) If X has neighbors and is a forced neighbor, X is a jump point. C) If the parent moves diagonally to X, and X can reach the jump point by moving horizontally or vertically, then X is a jump point.

The improved A* algorithm is equivalent to changing the way of expanding child nodes. It is no longer just expanding nodes around the parent node, but looking for jump points. the specific process is as follows:

1) Grid the search area.

2) Judge whether the starting point and the target point are in the same connectivity area. If they are in the same connectivity area, add the starting point to the OpenList; otherwise, the pathfinding ends.

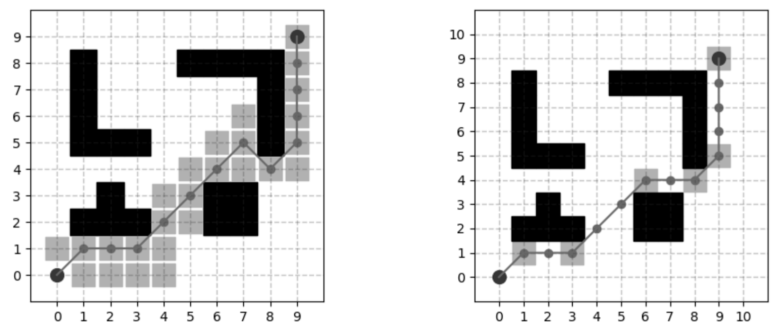
3) Find the node with the smallest F value in OpenList and set it as the current node current.

4) Determine whether the current node is the target point. If it is the target point, the pathfinding ends. Otherwise, delete current from OpenList and add it to Closelist.

5) Expand from the current node to eight directions to find the jump point.

6) Judge whether the jump point is found. If the jump point is found, proceed to the next step; otherwise, this node is ignored.

7) Determine whether the jump point exists in OpenList. If so, calculate the G value of the jump point and compare it with the parent node to determine whether to update the data; otherwise, jump points are added to OpenList. Skip to Step 3 and look for the point in the OpenList with the lowest F value.



a) the original A algorithm* *b) the improved A* algorithm*

Figure 1: The search process of path planning

According to Fig 1, compared with the original A* algorithm, the search area of the optimized A* algorithm is less than that of the original A* algorithm, and the search path efficiency of the A* algorithm optimized by JPS is obviously faster than that of the original A* algorithm, and the path found is shorter.

2.3. Floyd algorithm is used to optimize A* algorithm

The result of the original A* algorithm is a series of coordinate points representing the coordinates of the waypoints traversed. But such paths are often "jagged" and do not correspond to real-world intelligence. Therefore, further smoothing is needed. In this paper, Floyd algorithm is used to optimize the A* algorithm to reduce the inflection point in the path and reduce the path length.

Floyd algorithm[8], also known as Interpolation Point Method, is an algorithm used to solve the shortest path between two points. Floyd's principle is shown in Figure 2.

$L(A,D)$ represents the shortest distance from A to D. It can be seen from Figure 4 that there is an obstacle between AD, and $L(A,D) = +\infty$ at this time. Insert point B between AD, and $L(A, D) = L(A, B) + L(B, D)$ at this time, and the path is $A \rightarrow B \rightarrow D$, and there is no obstacle in this path. Then insert point C between two AD points, at this time $L(A, D) = L(A, C) + L(C, D)$, the path is $A \rightarrow C \rightarrow D$, and $L(A, C) + L(C, D) < L(A, B) + L(B, D)$. So, the shortest path from A to D is $A \rightarrow C \rightarrow D$.

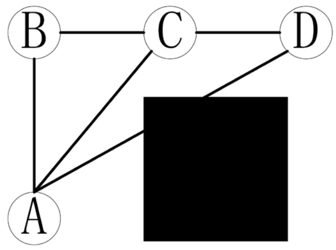
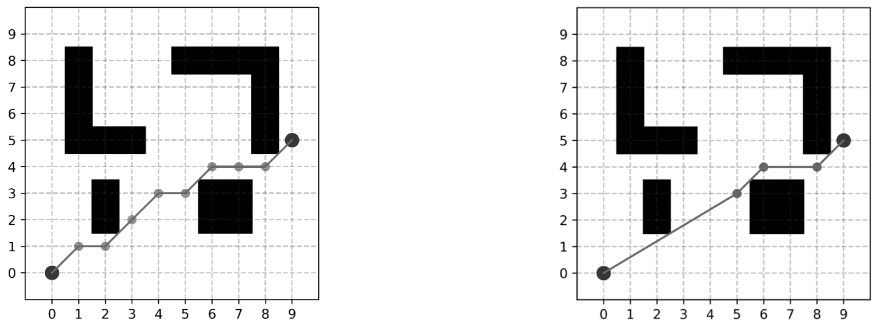


Figure 2: Floyd schematic diagram

The optimization steps of Floyd algorithm are as follows: (1) remove the adjacent collinear points to reduce the amount of calculation; (2) Remove the extra turning points. As can be seen from Fig. 3, the path optimized by Floyd algorithm has fewer inflection points, smoother paths, and shorter paths, which are more consistent with the motion path of the robot.



a) The original A algorithm* *b) Floyd optimization A* algorithm*

Figure 3: Result of path planning

3. Planning the global optimal path

3.1. The original genetic algorithm

Genetic algorithm[9][10] is a global search optimization algorithm, which simulates the phenomena of replication, crossover and mutation occurring in selection and heredity under natural environment. Starting from the initial population, a group of individuals more suitable for the environment can be generated through random selection, crossover and mutation operations, so that the population evolves to a better and better area in the search space. In this way, they will continue to multiply and evolve generation after generation, and eventually converge to a group of individuals that are most adapted to the environment, so as to obtain a high-quality solution to the problem. The core operations of genetic algorithm are as follows:

1) Selection

Selection is to select the best individuals from the current population, so that they have the chance to become the parents and reproduce for the next generation. Genetic algorithm embodies this idea through the selection process. The principle of selection is that adaptable individuals have a greater probability of contributing to the next generation. The operation of choice embodies the Darwinian principle of

ISSN 2616-5775 Vol. 6, Issue 2: 28-35, DOI: 10.25236/AJCIS.2023.060205

survival of the fittest.

2) Crossover

Crossover operation is the main genetic operation in genetic algorithms. By crossing over we can get a new generation of individuals that combines the characteristics of their parents. Cross operation embodies the idea of information exchange.

3) Mutation

Mutation operation firstly selects an individual randomly in the population, and then randomly changes a certain value in the structural data for the selected individual with a certain probability. As in the biological world, the probability of mutation in genetic algorithms is very low, and the value is generally very small.

3.2. Improved genetic algorithm

In the early stage of genetic algorithm, there are great differences among individuals. After selection operation, the fitness of selected individuals is relatively large. Therefore, in the early stage of the algorithm, it is easy to make the offspring of individual with good fitness fill the whole population, resulting in premature development. In the later stage of genetic algorithm, the fitness tends to be the same, and the advantage of excellent individuals is less and less obvious when they produce offspring, which leads to the stagnation of the whole population evolution. Therefore, the fitness can be appropriately stretched by combining simulated annealing algorithm[11][12] with genetic algorithm. In this way, when the temperature is high (in the early stage of genetic algorithm), individuals with similar fitness will have similar probability of producing offspring. However, when the temperature drops, the effect of stretching is strengthened, which magnifies the fitness difference of individuals with similar fitness, so that the advantage of excellent individuals is more obvious.

In addition, in order to improve the local search ability of GA, successive evolutionary reverse transcription operators and insertion operators are introduced after selection, crossover and mutation. The reverse transcription operator (RTO) is to randomly select a gene interval on a chromosome and flip all gene sequences within this interval. The genes outside the chromosome interval remain unchanged and are directly copied to the child chromosome. Only those with improved fitness value will remain, otherwise the reversal is invalid. Insert the operator (ITO) is to randomly select a gene interval on a chromosome and insert all the gene sequences in this interval into any other position of the chromosome. Only the chromosome with improved fitness value will be left, otherwise the insertion will be invalid.

The specific steps of the improved genetic algorithm are as follows:

1) Initial control parameters: population size, maximum evolution times MAXGEN, crossover probability, mutation probability, initial annealing temperature, cooling coefficient K, and termination temperature $\mathrm{T_{end}}$.

2) Calculate the fitness f_i of each individual in the population.

3) Set the cycle count variable $\mathrm{gen} = 0$.

4) A new population is formed by genetic manipulation such as selection, crossover, mutation, reverse transcription and insertion.

5) Calculate the fitness f_i' of each individual in the new population, and replace the old individual with the new one if $f_i > f_i'$; Otherwise, accept the new individual and discard the old one with probability $\mathrm{P} = \exp(f_i - f_i') \times T$).

6) If $\mathrm{gen} < \mathrm{MANGEN}$,then $\mathrm{gen} = \mathrm{gen} + 1$, go to Step 4; Otherwise, go to Step 7.

7) If $T_i < T_{end}$, the algorithm ends successfully; Otherwise, perform the cooling operation $T_{i+1} = \mathrm{kT}$.

In order to verify the feasibility of the improved genetic algorithm, the original genetic algorithm and the improved genetic algorithm are respectively simulated in this paper, and the experiments are carried out in maps with different number of target points. The coordinates of the target points are mainly in the range of [90-125, 25-45]. The comparison of algorithm path planning is shown in Figure 4.

It can be seen from Fig. 4 that the iteration curve of the improved genetic algorithm is significantly slower than that of the original genetic algorithm, which indicates that the simulated annealing algorithm

ISSN 2616-5775 Vol. 6, Issue 2: 28-35, DOI: 10.25236/AJCIS.2023.060205

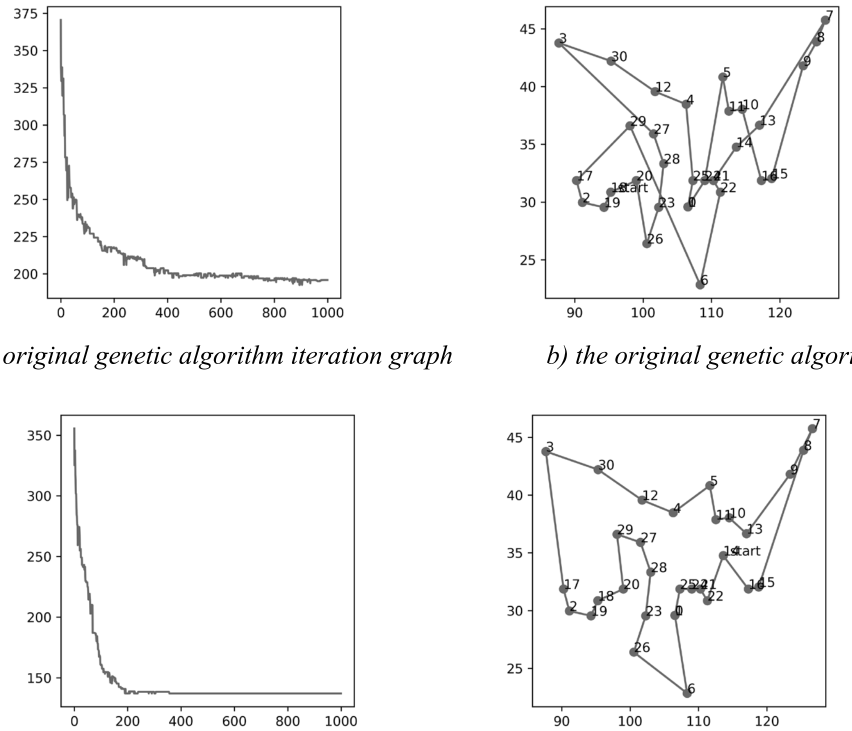
has effectively solved the problem of "premature" of the genetic algorithm, and it can be seen from the planned path that the path becomes regular and the path becomes shorter, which indicates that the improved genetic algorithm is feasible.

In order to verify the effectiveness of the improved genetic algorithm, examples in TSPLIB dataset were selected to test the improved genetic algorithm, and the results were compared with those obtained by the traditional genetic algorithm. In order to ensure the reliability of the comparison results, the initial number of experimental populations, crossover probability, mutation probability and iteration times of the traditional GA and the improved GA were set as 100, 0.6, 0.2 and 1000, respectively. In addition, the initial temperature, endpoint temperature and temperature change rate of the improved GA were set as 3000, 10-5 and 0.98, respectively. Under the same experimental environment, the average (AVG), optimal solution (Best), Bias (Bias) and variance (DX) were recorded after 20 separate runs. The calculation formula of deviation rate and variance is as follows:

$$\text{Bias} = \frac{Best - Opt}{Opt} \times 100\% \tag{2}$$

$$\text{DX} = \frac{AVG - Opt}{Opt} \times 100\% \tag{3}$$

Where Opt is the known optimal solution of TSPLIB dataset.



a) The original genetic algorithm iteration graph *b) the original genetic algorithm path*

c) Improved genetic algorithm iteration graph *d) Improved genetic algorithm path*

Figure 4: Iterative process and results of genetic algorithms

In addition to comparing the performance of the original genetic algorithm and the improved genetic algorithm, five other algorithms for solving the TSP problem were also selected for comparison. The validity of the five algorithms has been reflected in the references [13-17]. The specific results are shown in Table 2.

Through the analysis of the data in Table 1, the deviation rate of the improved genetic algorithm is less than 1%, which can be approximated as the optimal solution, indicating the effectiveness of the improved genetic algorithm. By comparing the optimal solution and average value of the original genetic algorithm and the improved genetic algorithm, the improved genetic algorithm is higher than the original genetic algorithm, which shows that the improved genetic algorithm has better global search ability and stability. When the number of target points is small, the deviation rate of the improved genetic algorithm is 0, and with the increase of the number of target points, the deviation rate of the improved genetic

ISSN 2616-5775 Vol. 6, Issue 2: 28-35, DOI: 10.25236/AJCIS.2023.060205

algorithm is also very small, and the variance of the improved genetic algorithm is also significantly higher than that of the original genetic algorithm. It shows that the improved genetic algorithm has higher precision and stronger robustness for solving the TSP problem. By observing the data in Table 2, it can be found that in the 9 groups of test examples, the solution results of the improved genetic algorithm are mostly better than the other 5 algorithms selected in the literature, which further proves the advantages of the improved genetic algorithm in solving TSP problems.

Table 1: Comparison of genetic algorithm and improved genetic algorithm.

Set	Opt	Algorithm	Best	Average	Bias	DX
ulysses22	74	GA	75.6	78	0.022	0.054
		IGA	74	75.3	0	0.018
oliver30	420	GA	465.2	519.77	0.108	0.238
		IGA	423.7	425.7	0.009	0.014
chn31	15377	GA	15527.4	18234.77	0.01	0.19
		IGA	15380.5	15515.76	0.00	0.01
		GA	504.3	567.43	0.184	0.332
eil51	426	IGA	433.7	441.7	0.018	0.037
berlin52	7542	GA	9005.2	10192.12	0.194	0.351
		IGA	7544.4	7881.58	0	0.045
rat99	1211	GA	1975.3	2368.02	0.631	0.995
		IGA	1263.1	1310.83	0.043	0.082
eil76	538	GA	719.2	829.99	0.337	0.543
		IGA	552.2	562.1	0.026	0.045
eil101	629	GA	1044	1120.5	0.660	0.781
		IGA	668	687.36	0.062	0.093
kroA100	21282	GA	44079.2	49430.13	1.071	1.323
		IGA	21504.1	22357.2	0.010	0.051
kroB100	22140	GA	44303.8	47743.45	1.001	1.156
		IGA	22341.4	23400.57	0.009	0.057
kroC100	20749	GA	43085.9	49734.62	1.077	1.397
		IGA	2085.9	22166.69	0.005	0.068
rand50	5553	GA	7450.6	8334.16	0.342	0.501
		IGA	5555.5	5807	0	0.046
pr76	108159	GA	168840.6	186888.01	0.561	0.728
		IGA	108280	110474.5	0.001	0.021
ch150	6528	GA	15395.8	17731.49	1.358	1.716
		IGA	6790.7	7128.8	0.040	0.092

Table 2: Comparison between improved genetic algorithm and other algorithms

Set	literature[13]	literature[14]	literature[15]	literature[16]	literature[17]	IGA
oliver30	424.69	**	420	**	420	423.7
eil51	436.53	438	426	431.74	426	433.7
st70	703.96	706	**	687.24	675	677.1
eil76	580.33	575	538	551.07	539	552.2
eil101	**	658	**	672.71	634	668
kroA100	**	25196	21282	22122.75	21282	21504.1
kroB100	**	26563	22141	**	22140	22341.4
KroC100	**	25343	20812	**	20749	20852.3
pr76	112249.91	109668	**	113798.56	**	108280

4. Combining algorithms for path planning

The steps of Combinatorial Algorithm for multi-objective point path planning:

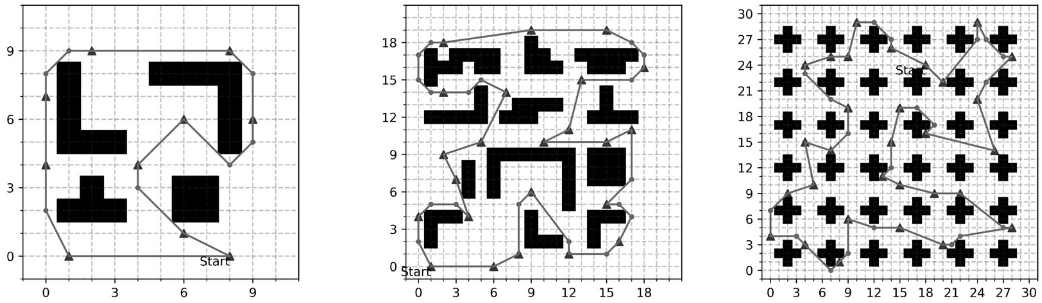
1) Calculate the shortest path of any two points;

2) Use the simulated annealing genetic algorithm for global path planning, and calculate the global shortest path;

3) Use the Floyd algorithm to optimize the global path

ISSN 2616-5775 Vol. 6, Issue 2: 28-35, DOI: 10.25236/AJCIS.2023.060205

Figure 5 shows the path planning of multi-target points. In the figure, obstacles are represented by solid black grids, each target point is represented by triangle, and the line between each target point is the path of the robot. The triangle marked Start in the lower left corner is the starting point. Figure 5 (a) has 10 target points and its planned shortest path is 40.5; Figure 5 (b) has 23 target points and its planned shortest path is 108.6; Figure 5 (c) has 30 target points and its planned shortest path is 164.4Through the contrast can be seen that the combination algorithm under the condition of static obstacles, to traverse the robot path planning of multiple target point is feasible, and whether the map obstacles is simple or complex, the number of tasks target more or less, the combination algorithm can planning for mobile robot from a safe without touch, short walking distance of the optimal path.



*a)10*10* *b)20*20* *c)30*30*

Figure 5: Raster path planning

5. Conclusion

For the path planning problem of multi-objective points, this paper proposes a combinatorial algorithm based on A* algorithm and annealing genetic algorithm. By integrating the JPS algorithm, the search efficiency of the A* algorithm is improved and the program running time is reduced. By using the Floyd algorithm to optimize the path of the A* algorithm, the inflection points in the path are greatly reduced, and the path length is shortened, so that the planned path is more in line with the motion law of the robot; The annealing algorithm and genetic algorithm are combined, and the reverse transcription operator and insertion operator are added to avoid the genetic algorithm falling into prematurity and improve the accuracy of the calculation result; finally, the improved A* algorithm is combined with the improved genetic algorithm to realize the path planning of multi-objective point motion of the mobile robot. Simulation experiments demonstrate the effectiveness of the fusion algorithm in different grid environments.

References

[1] Mengxi Li, Boxia He, Yu Zhou. Multi-objective path planning method for mobile robot based on A~ and ant colony algorithm [J]. Machinery & electronics,2021,39(06):61-65+69. https://kns.cnki.net/kcms/detail/detail.aspx?FileName=JXYD202106013&DbName=CJFQ2021*

[2] Yi Lu, Yongping Gao, Jiangteng Long. Research on Path Planning of Mobile Robot Based on A~ Algorithm [J]. Journal of Hubei Normal University (Natural Science),202 ,42(02):59-65. DOI: 10.3969/j.issn.2096-3149.2022.02.009*

[3] Jin Li, Chunlong Fu. Genetic Algorithm and ant colony algorithm based on the TSP problem study [J]. Computer Programming Skills & Maintenance, 2021(9):17-18+66. DOI: 10.16184/j.cnki.com. prg. 2021.09.007

[4] Fan Yang, Yachong Xue, Jing Li. Path planning for ergodic multi-task target robot under static obstacles [J]. Journal of Tiangong University, 2018, 37(04): 65-71. DOI: 103969/j.issn.1671-024x.2018.04.012

[5] Zhong, X., Tian, J., Hu, H. et al. Hybrid Path Planning Based on Safe A Algorithm and Adaptive Window Approach for Mobile Robot in Large-Scale Dynamic Environment. J Intell Robot Syst 99, 65–77 (2020). https://doi.org/10.1007/s10846-019-01112-z*

[6] Jiacheng CAI, Keqiang Bai, Xuchun Li, Zhengliang Huang, Zhigui Liu. Improved Path Planning Algorithm for Mobile Robot Based on JPS [J/OL]. Application Research of Computers:1-8[2022-07-20]. DOI: 10.19734/j.issn.1001-3695.2021.0671

ISSN 2616-5775 Vol. 6, Issue 2: 28-35, DOI: 10.25236/AJCIS.2023.060205

[7] Jianmeng Huang, Yuxiong Wu, Xiezhao Lin. Smooth JPS path planning and trajectory optimization method for mobile robot [J]. Transactions of the Chinese Society for Agricultural Machinery,2021,52(02):21-29+121. DOI: 10.6041/j.issn.1000-1298.2021.02.002

[8] Shuangquan Tang, Bofeng Zhang, Sen Mu, Zixi Chen, Haoming Feng, Jing Xu. Optimal path planning problem based on Floyd algorithm [J]. Scientific and Technological Innovation,2021(24):16-17. https://kns.cnki.net/kcms/detail/detail.aspx?FileName=HLKX202124007&DbName=CJFQ2021

[9] Ou, J., Hong, S.H., Ziehl, P. et al. GPU-based Global Path Planning Using Genetic Algorithm with Near Corner Initialization. J Intell Robot Syst 104, 34 (2022). https://doi.org/10.1007/s10846-022-01576-6

[10] Jidai Wang, Xindong Wang, Qunhong Tian, Aiqin Sun, Xinchao Zhang, Liang Yuan. Path planning of mobile robot based on improved fuzzy adaptive Genetic Algorithm [J]. Machine tool & Hydraulics, 2021, 49(23):18-23. DOI: 10.3969 /j. issn. 1001-3881. 2021. 23. 004

[11] Kesheng Chen, Sidong Xian, Peng Guo. Adaptive temperature Simulated Annealing algorithm for traveling salesman problem [J]. Control theory & applications, 2021, 38(02):245-254. DOI: 10.7641/ CTA. 2020.00090

[12] Jian Sun, Songzuo Liu, Xiaoxiao Wu, Simin Wu. Parallel Simulated Annealing algorithm to solve TSP based on Spark [J]. Electronic Measurement Technology, 2022, (4): 53-58, DOI: 10.19651 / j.carol carroll nki emt. 2108429.

[13] Jin Zhang, Guotong Bi, Lili Li. A discrete bat algorithm for solving TSP problem [J]. Computer Engineering & Science, 2018, 40(11):2085-2091. DOI: 10.3969/j.issn.1007-130X.2017.11.024

[14] Y. Wu, W. Song, Z. Cao, J. Zhang and A. Lim, "Learning Improvement Heuristics for Solving Routing Problems," in IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS. 2021.3068828.

[15] Juan Li, Xiaoming You, Sheng Liu, Jia Chen. Adaptive fuzzy ant colony system [J]. Computer Engineering and Applications, 2019, 55(15): 75-81. DOI: 10.3778/j.issn.1002-8331.1811-0349

[16] GÜNDÜZ, MESUT; KIRAN, MUSTAFA SERVET; and ÖZCEYLAN, EREN (2015) "A hierarchic approach based on swarm intelligence to solve the traveling salesman problem," Turkish Journal of Electrical Engineering and Computer Sciences: Vol. 23: No. 1, Article 9.https://doi.org/10.39 06/elk-1210-147

[17] Eneko Osaba, Xin-She Yang, Fernando Diaz, Pedro Lopez-Garcia, Roberto Carballedo, An improved discrete bat algorithm for symmetric and asymmetric Traveling Salesman Problems, Engineering Applications of Artificial Intelligence, Volume 48,2016, Pages 59-71, ISSN 0952-1976. https://doi.org/10.1016/j.engappai.2015.10.006