

Robot Path Planning Based On The Travelling Salesman Problem

Guoqing Wang^a, Jun Wang^{a*}, Ming Li^a, Hanjun Li^b, Yuan Yuan^a

^aSchool of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, 221116, China;

^bAir Force Logistics Academy, Xuzhou, 221000, China
 Wj999lx@163.com.

Against the background of "Robot Travelling China" competition, this paper introduces the idea of TSP to seek optimal motion path. Applying the knowledge of graph theory to analyze and transform the map of competition. In the paper, we introduce the Dijkstra algorithm to solve the shortest path problem, and add angle increasing function in the cost matrix to determine priorities. Finally the paper uses the discrete particle swarm algorithm to do the simulation, which achieves good experiment effect.

1. Introduction

Robot path planning problem is to find the optimal or sub-optimal path from the starting point to the target point (J. He, X. Yao, J. Li, 2005). The paper is against the background of "Robot Travelling China" competition (Mian Dong et al, 2014), the main goal of the competition is that the robot should travel as many attractions as possible and complete the plan of tourism activities at a predetermined time, then return to the place of departure (As shown in Figure. 1). This is a typical example of finding the optimal path planning.



Figure 1: Map of TSP

According to the competition rules, the robot will get higher score in the more far position while getting lower in the nearer position. If the robot wants to get higher score, it will have to travel more far with the risk of unable to return the starting area in a specified period of time. This is very similar to the traditional travelling salesman problem (TSP) (Yuren Zhou, 2009). TSP requires finding a minimum weight closed path which goes through each vertex at least once. Therefore, this paper is attempting to convert the path planning problem of "Robot Travelling China" into the standard TSP problem, which makes use of the discrete particle swarm algorithm to do the simulation.

2. The conversion from "Robot Travelling China" into the standard TSP problem

2.1 The establishment of scenic spot coordinates

In order to solve the practical problem of robot path planning, the scenic spot coordinates should be established firstly, seeing it as particle, getting the spread information with sequential codes and display it in the Matlab as shown in Figure 2;

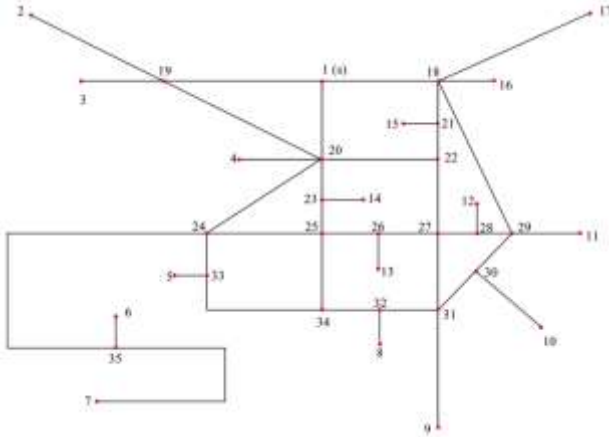


Figure 2: Abstract field map

2.2 Simplified field map

Do NOT begin a new section directly at the bottom of the page, but transfer the heading to the top of the next page. In order to solve and tectonic map of Hamilton conveniently, in the premise of not affecting the effect of Figure ring out the answers(A.T. Hayes, A. Martinoli, R.M. Goodman, 2002).we should simplify the abstract robot competition field map, the specific process is as follows :

2.2.1 As shown in Figure. 3, taking the node 19 for example, the robot has to go through the node 19 firstly if it wants to reach node 2&3,t hen returns by the way it comes with the situation of passing node 19 again, so when we are trying to find the optimal path for the robot, we can omit all the peripheral points as the nodes 2&3, including nodes 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, the following is the information of Figure. 3 converted from that of Figure. 3:

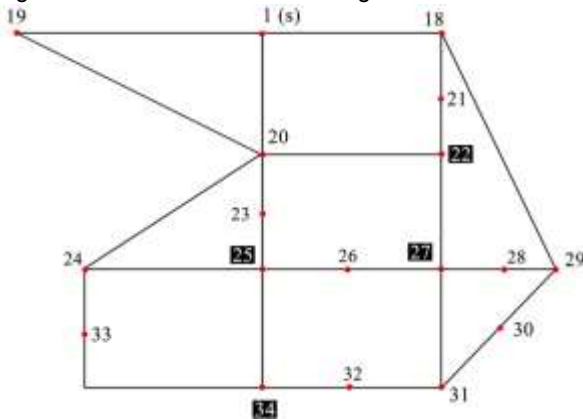


Figure 3: Simplified field map1

2.2.2 The nodes in the Black shadow including nodes 22,25,27,34 are not attraction sand don't directly connect with the attractions of nodes, which are the necessary nodes the tourism robots that have to go through connecting over 2 nodes usually. We omit these nodes and repaid the new assignment to the robot "tourism" line, which is shown in Figure. 4:

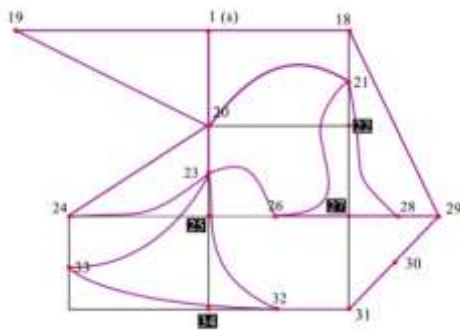


Figure 4: Simplified field map 2

The formula for the calculation of weights is $W(23, 33) = W(23, 25) + W(25, 33)$, and so on.

2.2.3 Node number conversion:

According to the weight formula we can get the weight matrix W , and remark it from 1 to 14. During the simplification process, we will record the controlling table of the numbers as shown in Figure 4. In the Figure 4, the node in NO.4 row is omitted during the process of the problem convert. It is a spot label the robot will reach, which is correspondingly connected with the third column. We continue the convert to get the diagram 5 based on the graph theory, and renumber the endpoints in the corresponding Figure. 4 as shown in the first column of the table, the blank place is the omitted points in Figure 5, which is vacated to show the relationship.

Node in Figure3-5	Node in Figure3-5	Node in Figure3-5	Node in Figure3-5
1	1	1	1(Starting Points)
2	2	18	16, 17
3	3	19	2, 3
4	4	20	4
5	5	21	15
	6	22	
6	7	23	14
7	8	24	35, 6, 7
	9	25	
8	10	26	13
	11	27	
9	12	28	12
10	13	29	11
11	14	30	10
12	15	31	9
13	16	32	8
14	17	33	5
	18	34	
		35	6, 7

Figure 5: Node number conversion

2.3 The introduction of Dijkstra algorithm to construct the shortest distance matrix

After the completion of the above work, the problem needed to be solved is to find the shortest distance between any points in the simplified map, thus the path planning problem can be converted to the TSP problem to solve. For it, this paper introduced Dijkstra matrix algorithm (LyubovYotova, Ahmed Hassaan, SpaskaYaneva, 2005). The Dijkstra algorithm can calculate the shortest distance between two points in the weighted graph.

The specific implementation process: According to the coordinates of the attractions in the simplified field map 1 (Figure. 4), and the distance formula between two points, we calculate the vertex distance matrix between any two points, which is marked "d"; Running the Dijkstra matrix algorithm in Matlab to get the shortest path

arbitrary matrix between any two points, which is named “dshortest”; finding the shortest distance of the corresponding points in map3 (Figure. 6), which is saved as “W”, thus we will get the shortest distance matrix between nodes which is named as the weight matrix “W”.

$$w_{ij} = \begin{cases} 0, & i = j; \\ w_{ij}, & i \neq j; \\ \infty, & i \neq j \end{cases} \quad (1)$$

We can see that the W is a symmetric matrix, using Dijkstra algorithm in the line k of W, then we can find the shortest distance from K to every other vertex, and save it in the line k of W to get the result in matrix W which is the shortest distance between any two vertices.

3. The construction of the robot path planning cost rules

When we are finding the optimal planning of robot competition path, the final path is the ideal one in which condition we ignore the practical contest site factors and some restrictions of competition rules. When it is applied to the actual competition, the effect may not be ideal. The following is the actual robot path planning optimization in the situation of real competition.

According to the competition rules, we will get the different score when arriving each of the spots, thus the nearer it gets, the higher it scores; that is to say score within easy reach of the attractions is low (S.D. Muller et al, 2002), higher score with spots is more difficult to reach, along with the condition that the time the contest robot spent in parading is limited. On the robot's own case, it has stable performance, which can run at a relatively high speed and has a less possibility to run away while running in the straight line; but when in the intersection, the robot will deviate from the white lead line even run away while turning comers. By changing the robot to the assignment matrix for each site, which is called cost matrix to solve these practical problems. Thus we should establish a set of perfect assignment rules firstly (Bernard K. –S Cheung et al, 2008) to improve the weight matrix W into the cost matrix F.

Numbers of scenic spots	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	Xinjiang	Dunhuang	Terracotta –CHINA	Mount Emei	The Potala Palace	Mount Everest	HongKong	Nansha isalping island	Ali Mountain	The Oriental Pearl Tower	Sun Yat-sen Mausoleum	Mount Huang	The Three Gorges	Mount Tai	Tian An Men	Changbai Mountain
Points	10	10	5	15	20	30	15	15	15	10	10	10	8	5	5	8

Figure 6: The contrast between spots number sand scores

3.1 The weight matrix assignment

According to the spot numbers and scores comparison table in Figure. 6, we can reassign the weight matrix W. The specific steps are as follows:

When competing with others, if the robot wants to get more points in a certain period of time, it will firstly get to the location from where is the nearest with the highest points. But according to the rules of competition, the higher the score, the longer the desired distance the robot to reach, the higher the difficulty coefficient. We can see the two factors as the desire and the difficulty to reach the destination point, this “desire” weakens the “difficulty” to reach a certain spot point, while the up and down of the degree of the “difficulty” weakens the “desire”. Obviously the relationship between them is inversely, so we introduce the following formula:

$$f_{ij} = \frac{w_{ij}}{0.1 \times grad_j} \quad (i, j = 1, 2, \dots, 14) \quad (2)$$

In the formula, f_{ij} is the element of cost matrix F , w_{ij} is the element of the weight matrix W , $grad_{ij}$ is the sum of points .

3.2 Add the angle function

In the above analysis, we have already known that the robots will slower the speed when turning corners to avoid the phenomenon of “deviation” or “coaster”. So we added the parameters’ in the cost function f_{ij} to avoid the unnecessary cost of time when turning corners (Nattapat Attiratanasunthron, JittatFakcharoenphol, 2008)).It is an increasing function of turning angle; we set the rules: when the robot turned right, $\tau_r=\tau$; when the robot turned left, $\tau_l=a\tau$; The decimal for ‘a’ is set between 0 and 1, which makes the left direction have a higher priority for the right direction. It has two advantages:

3.2.1 It is better to form a loop when the robot chooses the path, which means the result will be different while going forward and backward. The choice makes it easier for the robot to return the starting area after finishing the parade, which resulted in a higher score;

3.2.2 By introducing this rule, the robot will take turns to go straight then turning left and turning right finally. The parameters in the rules such as τ , a needs to be validated in the experiment to get the optimum value in the restricted ranges.

4. Simulation and analysis

According to competition requirements and site environment, we will introduce the discrete particle swarm algorithm for solving TSP (Todor Petkov, Sotir Sotirov, 2013) problems based on the rules of robot cost function and taking the shortest path matrix "W" for routing based on particle swarm velocity update. We can get the simulation results after the operation of the designed programs.

Comparing the change of scores of corresponding spots, we can see that the optimal path is a closed and no cross curve after completing iterative algorithm, which is obviously better than the initializing algorithm and solves the TSP problem reaching the expected requirements(JACK N. BOONE, 1991). Comparing the change of calculated parameters, we can find that the optimal particle values in the current position suffered a big surging after each iteration, the current position of the solutions in the optimal particle is a decreasing function and quickly converged to local optimal solution with the increasing numbers of iterations; The optimal values in Figure. 8 transited smoothly to a smaller global optimal solution, which explains that social learning behaviours can improve the robustness and efficiency of the algorithm in the particle swarm optimization algorithm (Yezheng Fan, 2015). Experimental results show that the effect of introducing TSP to solve the problem of optimization ideological Tour China robot path is good.

Matching the numbers of the attractions of the correspondence problem in the process of transformation, we get the optimal path of the robot 1,3,2,4,7,6,5,8,9,10,11,12,13,14,15,17,16, the weights for optimal path is 4478.1. As shown in Figure 7.

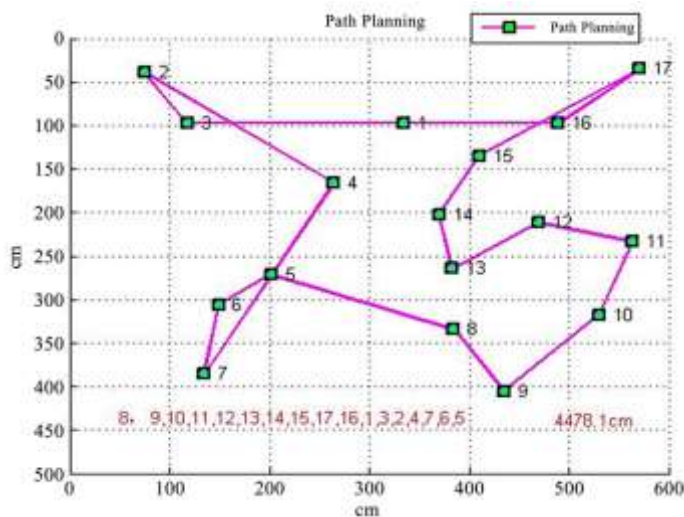


Figure 7: Optimal robot path

We can get the best attractions travel route after the comparison combining with the actual path of Figure. 1: Starting area—Xinjiang—Dunhuang—Xi’anTerracottaArmy—ThreeGorges—MountHuangshan--Nanjing--Nansha Taiping Island--Taiwan Ali Mountain--The Oriental Pearl Tower--Starting area.

5. Conclusion

Against the background of "Robot Travelling China" competition, this paper has introduced the idea of TSP to seek optimal motion path (Qingni Yuan, Qingyun Yuan, Junfeng Sun, 2014). The traveling path for robot after path planning hasn't contained all the attractions competition on the map, but automatically has discarded some of the more difficult to reach, reaching the process prone to "accident" attractions through the establishment of cost function, which has reduced the possibility of the "accident" to a certain extent, and we has gotten good effect both in the simulation and in the physical robot competition. We expect that the method proposed in the paper would also be beneficial in a variety of other optimal path planning problems such as the unmanned driving path plan.

Acknowledgements

The research described in this paper has been supported by three main projects: The Fundamental Research Funds for the Central Universities of China (Grant No: 2011QNB21); The Innovative engineering developing on graduates of Jiangsu Province in China (Grant No: CXZZ11-0293); The Chinese Coal Industry Association of science and technology guiding plan project (Grant No: MTKJ2011-306).

Reference

- Attiratanasunthron N., Fakcharoenphol J., 2008, a running time analysis of an Ant Colony Optimization algorithm for shortest paths in directed acyclic graphs [J]. *Information Processing Letters*, 105(3): 88-92. doi: 10.1016/j.ipl.2007.08.013
- Bernard K.S., Cheung K.L., Choy C.L. Li W.Z., Shi J.T., 2008, Dynamic routing model and solution methods for fleet management with mobile technologies [J]. *International Journal of Production Economics*, 113(2): 694-705. doi:10.1016/j.ijpe.2007.10.018
- Bonzel H.P., Bradshaw A.M., Ertl G., Eds., 1989, *Physics and Chemistry of Alkali Metal Adsorption*. Elsevier, Amsterdam, the Netherlands
- Boone J.N., 1991, Generalized covariance analysis for partially autonomous deep space missions [J]. *Journal of Guidance, Control, and Dynamics*, 14(5):964-972. doi: 10.2514/3.20737.
- Dong M., Sun Y.G., Qiang H.Y., Wei X.T., 2014, Modeling and simulation research of active heave compensation system [J]. *REVIEW OF COMPUTER ENGINEER STUDIES*, 1(2):15-18.
- Fan Y.Z., 2015, Semantic annotation and storage for tourism information [J]. *REVIEW OF COMPUTER ENGINEER STUDIES*, 2(2):1-4.
- HayesA.T. MartinoliA., GoodmanR.M., 2002, Distributed odor source localization [J]. *IEEE Sensors Journal*, 2(3): 260-271. doi :10.1109/JSEN.2002.800682.
- HeJ. YaoX. LiJ., 2005, a comparative study of three evolutionary algorithms incorporating different amount of domain knowledge for node covering problems [J]. *IEEE Trans. on Systems, Man and Cybernetics*, 35(2):266-271. doi: 10.1109/TSMCC.2004.841903.
- MullerS.D. MarchettoJ., AiraghiS. KoumoutsakosP., 2002, Optimization based on bacterial chemotaxis [J]. *IEEE Transactions on Evolutionary Computation*, 6(6):16-29. doi:10.1109/4235.985689.
- Petkov T., Sotirov S., 2013, Generalized Net Model of the Cognitive and Neural Algorithm for Adaptive Resonance Theory 1 [J]. *INT.J.Biautomation*, 17(4):207-216.
- Yotova L., Hassaan A., Yaneva S., 2015, Covalent Immobilization of Peroxidase onto Hybrid Membranes for the Construction of Optical Biosensor [J]. *INT.J.Biautomation*, 19(2):177-186.
- Yuan Q.N., Yuan Q.Y., Sun J.F., 2014, Development research of digital manufacturing resource modeling technology based on patent information analysis [J]. *MATHEMATICAL MODELLING AND ENGINEERING PROBLEMS*, 1(1): 13-16.
- Zhou Y.R., 2009, Runtime analysis of an ant colony optimization algorithm for TSP instances [J]. *IEEE Computational Intelligence Society*, 13(5): 1083-1092. Doi: 10.1109/TEVC.2009.2016570.