

Research on Model and Algorithm of Task Allocation and Path Planning for Multi-Robot

Zhenping Li, Xueting Li*

School of Information, Beijing Wuzi University, Beijing, China

Email: *1137186920@qq.com

How to cite this paper: Li, Z.P. and Li, X.T. (2017) Research on Model and Algorithm of Task Allocation and Path Planning for Multi-Robot. *Open Journal of Applied Sciences*, 7, 511-519.

<https://doi.org/10.4236/ojapps.2017.710037>

Received: September 15, 2017

Accepted: October 13, 2017

Published: October 16, 2017

Copyright © 2017 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Based on the modeling of robot working environment, the shortest distance matrix between points is solved by Floyd algorithm. With the objective of minimizing the sum of the fixed cost of robot and the cost of robot operation, an integer programming model is established and a genetic algorithm for solving the model is designed. In order to make coordination to accomplish their respective tasks for each robot with high efficiency, this paper uses natural number encoding way. The objective function is based on penalty term constructed with the total number of collisions in the running path of robots. The fitness function is constructed by using the objective function with penalty term. Based on elitist retention strategy, a genetic algorithm with collision detection is designed. Using this algorithm for task allocation and path planning of multi-robot, it can effectively avoid or reduce the number of collisions in the process of multi-robot performing tasks. Finally, an example is used to validate the method.

Keywords

Path Planning, Task Allocation, Collision Detection, Mathematical Model, Genetic Algorithm

1. Introduction

With the development of industrial robot technology [1], a single robot cannot meet the needs of society, and many robots cooperate to accomplish complex and dangerous tasks [2]. In the process of multi-robot collaborative performing tasks, if the collision occurs between robots, it will cause losses such as robot damage, prolong the working time, and even make the system paralyzed, so planning the path for robots should reduce the occurrence of collisions as far as possible.

For the problem of robot's path planning, many scholars have studied it. Par-

ticle swarm optimization algorithm or navigation algorithm are used to solve the problem [3] [4] [5] [6] [7]. In the aspect of robot's task allocation, task allocation strategy [8] and ant colony algorithm are adopted [9] [10]. In the part of collision avoidance, improved artificial potential field obstacle avoidance method [11] and robot's collision avoidance algorithm based on RRT are adopted [12].

These scholars have proposed different algorithms to solve the problems of robot's path planning, task allocation and collision avoidance respectively. But few people establish integer programming models and corresponding algorithms to solve them. Considering the maximum running time of a single robot, this paper aims at minimizing the sum of the robot's fixed cost and the robot's running cost. An integer programming model for task allocation and path planning of multi-robot with collision detection is proposed. Introducing collision penalty term, a genetic algorithm with collision detection is designed to calculate the optimal number of robots, the tasks completed by each robot and the collision free path of each robot, making the total cost of the detection tasks lowest.

2. Problem Description and Mathematical Model

2.1. Problem Description

A factory uses m robots to complete the monitoring of n monitoring points. Each robot filled with power can travel limited distance continuously and each monitoring point is monitored with equal time. When the task is executed, each robot starts from the platform at the same time, and tries to avoid collisions among them during the process of performing tasks, and finally returns to the departure platform. In the process of completing tasks, the robot performs only one task, each task requires only one robot to complete, each task is only performed once, and all tasks will be executed. The call cost and the operating cost of per unit distance are known for each robot, calculating the optimal number of robots to complete tasks, and planning the path of each robot, making the total cost of monitoring tasks completed lowest.

2.2. Mathematical Model

In order to establish the model, the following symbols are defined:

- 0: Represents departure platform;
- $M = \{1, 2, \dots, m\}$: Represents a set of available robots;
- L : Represents a set of feasible walking points for all robots;
- $D = \{1, 2, \dots, n, n+1\}$: Represents a set of points to be monitored;
- $n+1$: Represents return platform can be considered as a replicate departure platform;
- G : Represents a set of generally walk able grid points, and $L = D \cup G$;
- $T = \{1, 2, \dots, t\}$: Represents the running time set of robots;
- d_{ij} : Represents the distance from the i point to the j point of robots
- $i, j = 0, 1, 2, \dots, n, n+1$;
- t_{ij} : Represents the driving time from the i point to the j point of robots,
- $i, j = 0, 1, 2, \dots, n, n+1$;

s : Represents the monitoring time of each robot at each point;

g : Represents the fixed cost of using a robot;

f : Represents the per unit distance cost of each robot;

$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1r} \\ a_{21} & a_{22} & \cdots & a_{2r} \\ \vdots & \vdots & & \vdots \\ a_{r1} & a_{r2} & \cdots & a_{rr} \end{bmatrix}$: Represents the adjacency distance matrix between

feasible grids; when $a_{ij} = 1$ represents the point i adjacent to the point of j , i and j are feasible grid points, otherwise $a_{ij} = 0$.

Define the following decision variables:

$x_{ijkt} = \begin{cases} 1, \text{Robot } k \text{ from the point of } i \text{ arrives at the point of } j \text{ directly at time } t \\ 0, \text{otherwise}; k \in M; t \in T; i, j \in L \end{cases}$

$y_{ikt} = \begin{cases} 1, \text{robot } k \text{ services for the point of } i \text{ at time } t \\ 0, \text{otherwise}; k \in M, t \in T, i \in D \end{cases}$

$C_{jkpt} = \begin{cases} 1, x_{ijkt} = x_{ujpt} = 1, \text{Robot } k \text{ and Robot } p \text{ occur collision} \\ \text{at the point of } j \text{ at time } t \\ 0, \text{otherwise}; t \in T; i, u \in L; j \in L; k \neq p \in M \end{cases}$

In order to reduce collisions between robots' paths, a penalty term is constructed based on the number of collisions between paths. By introducing the penalty term into the objective function, an integer programming model for the task allocation and path planning of multi-robot with collision penalty is established.

$$\min z = g \sum_{k \in M} \sum_{j \in D} \sum_{t \in T} x_{0jkt} + f \sum_{k \in M} \sum_{i \in L} \sum_{j \in L} \sum_{t \in T} d_{ij} x_{ijkt} + \lambda \sum_{j \in L} \sum_{k \in M} \sum_{p \neq k \in M} \sum_{t \in T} C_{jkpt} \quad (1)$$

$$\sum_{j \in L} x_{0jkt} = 1, k \in M \quad (2)$$

$$\sum_{i \in L} \sum_{t \in T} x_{in+1kt} = 1, k \in M \quad (3)$$

$$\sum_{i \in L} x_{i0kt} = 0, k \in M; t \in T \quad (4)$$

$$\sum_{j \in L} x_{n+1jkt} = 0, k \in M; t \in T \quad (5)$$

$$\sum_{k \in M} \sum_{t \in T} y_{ikt} = 1, i \in D \quad (6)$$

$$\sum_{i \in L} x_{ijkt} = \sum_{u \in L} x_{juk(t+s+1)} \quad j \in D; k \in M; t \in T \quad (7)$$

$$\sum_{i \in L} x_{ijkt} = \sum_{u \in L} x_{juk(t+1)} \quad j \in G; k \in M; t \in T \quad (8)$$

$$\sum_{i \in L} x_{ijkt} = y_{jkt} \quad j \in D; k \in M; t \in T \quad (9)$$

$$2C_{jkpt} \leq y_{jkt} + y_{jpt} \quad j \in L; k, p \in M; t \in T \quad (10)$$

$$y_{jkt} + y_{jpt} \leq 1 + C_{jkpt} \quad j \in L; k, p \in M; t \in T \quad (11)$$

$$x_{ijkt} \leq a_{ij} \quad i, j \in L; k \in M; t \in T \quad (12)$$

$$x_{ijkt} = 0, 1 \quad i, j \in L; k \in M; t \in T \quad (13)$$

$$C_{jkpt} = 0, 1 \quad j \in L; k, p \in M; t \in T \quad (14)$$

$$y_{jkt} = 0, 1 \quad j \in L; k \in M; t \in T \quad (15)$$

(1) Represents the minimum total cost with a penalty item; (2) Represents each robot must leave from platform 0; (3) Represents each robot must return to the platform point after completing the inspection task; (4) Represents each robot cannot return to the departure platform 0; (5) Represents each robot cannot leave the platform $n + 1$; (6) Represents each monitoring point is also being serviced by one robot; (7) Represents each robot arrived at the monitoring point of j must leave from the monitoring point of j ; (8) Represents each robot arriving at the general grid point J must leave from the first J point; (9) Represented a robot serviced for a monitoring point must have reached the monitoring point; (10), (11) Represent robot k and robot p occur collision at time t ; (12) Represents robots can only walk into an adjacent grid within a unit time; (13), (14), (15) Represent a decision variable value constraint;

Since it will take too long to solve the integer programming model directly, we will design a genetic algorithm with collision detection to solve the task allocation planning problem for large scale robots.

3. Genetic Algorithm with Collision Detection

The steps of the genetic algorithm will not be described, and then the collision detection method proposed in this paper will be described in detail [13].

In order to avoid the robot's collision loss, the collision penalty term is added to the calculation of the objective function and the fitness value. The collision penalty term is defined according to the number of collisions between the robot paths. The following examples are combined with specific examples to describe the collision number detection algorithm.

According to the layout of a warehouse, the working environment after modeling by using the serial number grid method is shown in **Figure 1**. The gray squares represent the infeasible areas, the red squares represent the departure platforms, and the blue squares represent the points to be monitored. There are 10 monitoring points in the warehouse, which are indicated by serial numbers {18, 30, 112, 135, 157, 196, 223, 235, 245, 267}, and the red departure platform is indicated by serial number 264.

1) First, **Figure 1** is shown with the adjacency distance matrix A , and the adjacency matrix is used to store the graph. Floyd algorithm [14] [15] is used to solve the shortest path between any two points in the network. In this paper, the Floyd algorithm is implemented by MATLAB, and the shortest distance matrix and the routing matrix between any two points in the graph are calculated by the adjacency distance matrix, and can query the shortest distance between any two points and routing.

2) Suppose the robots walk within one grid per unit time, and the distances traveled by robots are quantified by the number of grids. Two units of time are

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68
69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85
86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102
103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136
137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153
154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170
171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187
188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204
205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238
239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272
273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289

Figure 1. Working environment of robots.

needed to be monitored for the grid of the monitoring points, and the monitoring points are quantized into two walk able grids. For example, with the maximum travel time of the robot as a restriction, decoding a feasible solution is 264-245-196-235-267-264 264-157-18-223-264 264-135-30-112-264, in which the path of the R_1 can be simply described as 264 → 246 → 245 → 229 → 213 → 196 → 197 → 215 → 233 → 234 → 235 → 251 → 267 → 266 → 265 → 264. Similarly, the path of R_2 and the path of R_3 can be obtained.

3) The paths for each robot to complete their monitoring tasks can be traced by the routing matrix calculated by the Floyd algorithm. Then the shortest distance matrix is solved according to the Floyd algorithm, and the driving distance of each robot is obtained. Each robot starts at the same time from the starting platform and runs at the same speed, In the process of performing tasks, the robot will collide when and when the two robots arrive at the same grid point at the same time. It can be described as follows:

$$\begin{cases} 2C_{jkpt} \leq y_{jkt} + y_{jpt} & j \in L; k, p \in M; t \in T \\ y_{jkt} + y_{jpt} \leq 1 + C_{jkpt} & j = L; k, p \in M; t \in T \end{cases} \quad (16)$$

4) In order to reduce the collision, a penalty term is added to the individual objective function according to the total number of collisions of each robot path in the individual, so that the population is continuously optimized. First, a collision algorithm is used to solve the total number of collisions CN between all paths of two robots corresponding to a chromosome, which is used as a measure of punishment. Take λCN as the penalty term, among which $\lambda (\lambda = 0.6)$ is coefficient.

$$CN = \sum_{j \in L} \sum_{k \in M} \sum_{p \neq k \in M} \sum_{t \in T} C_{jkpt} \quad (17)$$

$$\text{penalty} = \lambda CN \quad (18)$$

$$f_{new}(x_i) = f_{old}(x_i) + \text{penalty} \quad (19)$$

CN is the total number of collisions on a single chromosome, $\lambda (0 < \lambda < 1)$ is the Correction parameter of penalty. $f_{old}(x_i)$ is the corresponding objective function value of individual x_i , $f_{new}(x_i)$ is the sum of corresponding objective function and collision penalties of individual x_i .

5) Selection, crossover, mutation and other operations are performed according to genetic algorithms until the iteration goes to the maximum iteration path. The approximate optimal solution or optimal solution obtained is the optimal path.

4. Case Solving

In the warehouse layout shown in **Figure 1**, the gray grid represents the infeasible area, the red grid represents the departure platform and the blue grid represents the monitoring point. At the same time, each robot starts from the departure platform and completes the monitoring task of ten points to be monitored, which is $\{18, 30, 112, 135, 157, 196, 223, 235, 245, 267\}$. The warehouse has a total of 10 robots available, each robot filled with electricity can work continuously for 60 units of time and the speed of each robot is equal. The fixed cost of call a robot is 60 yuan and the cost of each robot driving unit distance is 2 yuan.

The program code is written by MATLAB, and the total number of collisions of each chromosome is calculated by the collision detection method, and the fitness of the penalty term is further calculated. Set the population size is 50 and the maximum evolution algebra is 100. In windows 7 (g4, 2G, 32 operating system) environment run the algorithm program 30 times [13].

According to the working environment of the robot in **Figure 1**, the adjacency matrix is used to store **Figure 1**. Based on the advantages of the Floyd algorithm, the distance matrix and the routing matrix between any two points in the adjacency distance matrix can be calculated. The optimum number of robots is 3 based on a genetic algorithm with collision detection. The program with collision detection algorithm runs 30 times, and in the 30 run, the corresponding statistical results of the running distance of 3 robots are: the average value is 86.13, the maximum value is 90, the minimum value is 80, and the standard deviation is 2.36. The total cost of 3 robots completed tasks are: the average total cost is 357.62, the maximum is 364, the minimum is 340, and the standard deviation is 5.51. The statistical results of running the program for 30 times are: the average time is 33.65, the maximum is 37.86, the minimum is 29.96, and the standard deviation is 2.39.

Figure 2 is an iterative graph of an optimal solution. It can be seen that when the genetic algorithm program with collision detection runs, it converges to the

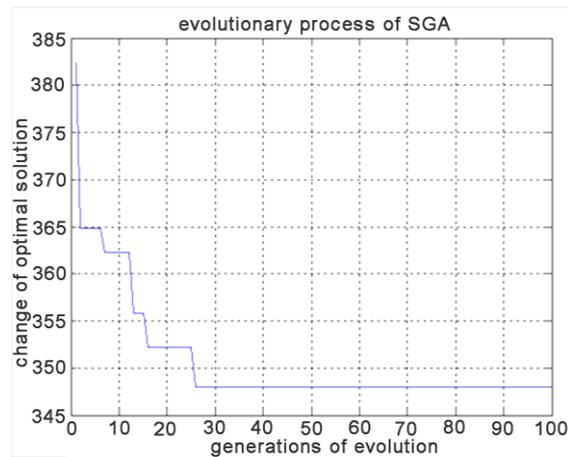


Figure 2. Convergence process of genetic algorithm.

optimal solution about 26 iterations, and the optimal solution does not change after 26 generations.

Figure 3 is a collision free path corresponding to 3 robots solved in this paper, which can be simply described as:

V1: 264--245--112--18--264;

V2: 264--223--157--196--267--264;

V3: 264--135--30--235--264.

It can be seen from **Figure 3** that the optimal running path of 3 robots can be solved by using genetic algorithm with collision detection. Although the paths of V1, V2 and V3 shown in **Figure 3** having overlapping parts, 3 robots leave the platform at the same time and the robots run at the same speed, so the time to run to the same path point is different and there will be no collisions actually.

5. Concluding Remarks

This paper studies the problem of multi-robot path planning based on genetic algorithm. Firstly, the adjacency matrix is used to store the example data. The shortest distance matrix and the routing matrix between the detected points are solved by using the Floyd algorithm according to the adjacency distance matrix [16]. The shortest time matrix is obtained from the shortest distance matrix, and the initial population is generated randomly according to the genetic algorithm. The task is assigned to robots with the longest running time of each robot as the constraint. Using the backtracking routing function of Floyd algorithm, the path of each robot is obtained according to the tasks assigned to each robot. The genetic algorithm of collision detection is used to solve the task allocation and path planning model of multi-robot, getting the optimum number of robots. The tasks each robot needs to accomplish, and the least cost collision free path is planned at the same time.

On the one hand, a collision detection algorithm is designed to avoid collisions in the running process of multi-robots; on the other hand, with the longest running time of each robot as the constraint, using the least robot completes

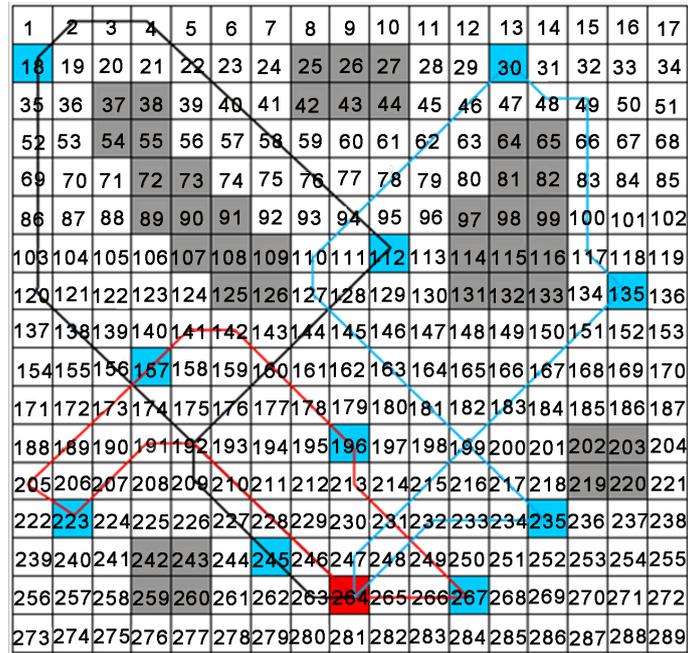


Figure 3. The running path of robots this paper.

the monitoring tasks making the total cost lowest. The genetic algorithm with collision detection designed in this paper can solve the problem of task allocation and path planning of multi-robot simultaneously.

In this paper, assuming that all robots start from the same platform, the collision between multi-robot can be further reduced by adjusting the starting platform of each robot actually; this paper does not consider the priority of robots to avoid collisions, so the robot can be based on a certain priority rules for collision avoidance. In the following research, we will consider these factors and further study the problem.

Acknowledgements

This work was supported by National Natural Science Foundation of China (71540028), Teaching Master of Beijing High Tech Project (G02040011); the Funding Project of High Level Cultivation of Beijing Wuzi University (GJB20164005).

References

- [1] Sun, F. (2012) Research on the Development of Industrial Robots in China. *Science Technology and Engineering*, **12**, 2912-2918.
- [2] Zhang, F. (2005) Improved Market Approach for Multi Robot Collaborative Exploration. *Control and Decision*, **20**, 516-524.
- [3] Zeng, N. (2016) Path Planning for Intelligent Robot Based on Switching Local Evolutionary PSO Algorithm. *Emerald Insight*, **36**, 120-126.
- [4] Dong, Y. (2016) Disordered and Multiple Destinations Path Planning Methods for Mobile Robot in Dynamic Environment. *Journal of Electrical and Computer Engineering*, **10**, 1-10. <https://doi.org/10.1155/2016/3620895>
- [5] Lu, Y. and Zhao, Y. (2015) Path Planning of Narrow Space Based on Improved Ge-

- netic Algorithm. *Application Research of Computers*, **32**, 414-418.
- [6] Lie, X. (2011) A Tabu Search Autonomous Navigation Algorithm for Mobile Robot. *Control and Decision*, **26**, 1310-1314.
- [7] Tang, W. (2012) The Improved Optimization Algorithm Is Used for Path Planning of Robots. *Science Technology and Engineering*, **12**, 7599-7601.
- [8] Chen, X. (2013) Multi Robot Task Allocation Based on Partitioning. *Journal of Southern Yangtze University (Natural Science Edition)*, **12**, 412-414.
- [9] Cao, Z. and Wu, B. (2013) Assignment of Multi Robot Tasks Based on Improved Ant Colony Algorithm. *Modular Machine Tools and Automatic Machining Technology*, **2**, 34-37.
- [10] Luo, X. and Liu, H. (2017) Application of Optimized Fuzzy Decision Algorithm in Multi Homing Vehicle Scheduling Problem. *Science Technology and Engineering*, **17**, 85-90.
- [11] Yin, X. and Hu, Y. (2016) Research on Path Planning of Robot Obstacle Avoidance in Unknown Environment. *Science Technology and Engineering*, **33**, 221-225.
- [12] Li, H. and Liang, Y. (2012) Research on Robot Collision Avoidance Motion Planning Algorithm Based on RRT. *Journal of Shenzhen Institute of Information Technology*, **10**, 21-26.
- [13] Li, Z. and Li, X. (2016) Genetic Algorithm for Task Allocation And Path Planning of Multi-Robot System. *Journal of Mathematical Sciences and Application*, **11**, 1-7.
- [14] Wang C. and Li, X. (2017) Leg Deformation Strategy of Antiriot Robot Based on Floyd Algorithm. *Science Technology and Engineering*, **17**, 70-73.
- [15] Zhu, H. and Zhang, Y. (2011) Find All the Shortest Paths among Nodes Based on Improved Floyd Algorithm. *Network and Multimedia*, **35**, 65-67.
- [16] Zuo, X. and Shen, W. (2017) Improved Algorithm for Multiple Shortest Path Problem Based on Floyd Algorithm. *Computer Science*, **44**, 222-227.