Task Assignment and Route Programming for Pattern Reformation on Grid Plane

Kuo-Hsien Hsia
Department of Electrical Engineering, Far East University
No.49, Zhonghua Rd., Xinshi Dist., Tainan City 74448, Taiwan (R.O.C.)

Bo-Yi Li
Graduate School of Engineering Science & Technology (Doctoral Program)
National Yunlin University of Science & Technology
No. 123, University Road, Section 3, Douliou, Yunlin 64002, Taiwan (R.O.C.)

Kuo-Lan Su
Department of Electrical Engineering, National Yunlin University of Science & Technology
No. 123, University Road, Section 3, Douliou, Yunlin 64002, Taiwan (R.O.C.)
E-mail: khhsia@cc.feu.edu.tw, g9910813@yuntech.edu.tw, sukl@yuntech.edu.tw
www.feu.edu.tw

Abstract

Pattern reformation problem on a grid plane is an interesting problem. Suppose there is a group of mobile robots on a square grid plane with the number of robots equaling to the number of columns/rows of the grid plane. The mobile robots are commanded to change to another formation as soon as possible. Any one of them has to prevent himself from colliding with other mobile robots during the moving of reformation process. In this paper, we will provide a systematic process on task assignment of route programming for the pattern reformation problem on a square grid plane. We will also provide a checking process to check if the suggested program is really good or not.

Keywords: Task assignment, pattern reformation, grid plane.

1. Introduction

There are mainly two category of programming problem: one is maximization problem and the other one is minimization problem. Minimum cost and minimum time are two of the most commonly used criterion of the minimization problem. Usually, the objective function of a minimum time problem is the sum of the spent time. However, for a problem that people start their works in cooperation at the same time and they want to complete the works as soon as possible, the objective function should be the maximum of the spent time. Pattern reformation problem is one of these problems.

2. Problem Formulation

Suppose there are some mobile robots placed on a square grid plane and forming certain pattern. The number of the mobile robots are the same as the rows of the square grid plane. There may or may not be obstacles on the grid plane. The mobile robots are commanded to reform another pattern in the shortest time and they can only move along the rows or the columns of the grid plane. This is definitely a minimum-time programming problem. Figure 1 is an example for six mobile robots on a 6x6 grid plane where the black squares form the target pattern, i.e. the 6 mobile robots are commanded to form the new pattern designated by the circles.

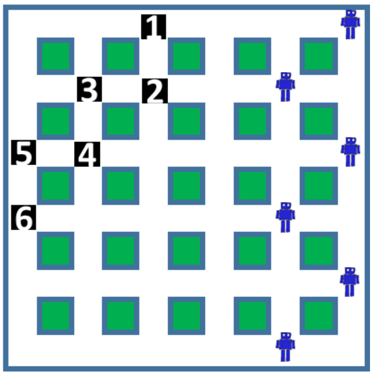

Fig. 1. Example of 6 mobile-robots on a 6x6 grid plane with the black squares forming the target formation.

There are two sub-problems that have to be considered for solving this problem. The first one is the task-assignment problem that each mobile robot's destination has to be determined for the minimum time requirement. The second one is the route programming problem that how each mobile robot should move to the target position in the shortest time without colliding with any other one mobile robot once the task assignment is completed. While dealing with the second sub-problem, it is possible to return to the first sub-problem because of the route conflict of any two mobile robots and one of them should wait for a step or detour to a further routing.

3. Algorithm for Task Assignment with Verification

Let each mobile robot has its ID and each target is also assigned an ID for convenience. The grid points on the grid plane are numbered sequentially with column majored from up to down and from left to right. For example, the grid point on the cross of the i^{th} row and j^{th} column on a 6x6 grid plane is numbered as $(j\text{-}1)\text{x}6+i$. With these information, we can form a distance matrix with the row number represents the robot ID and the column number represents the destination ID. The element at row i and column j represents the steps needed for the mobile robot i to the destination j. This matrix can be determined by the diffusion method [1-2].

For the case of Fig. 1, suppose the robots are numbered from top to bottom as 1 to 6. By diffusion method¸ the distance matrix is:

$$\Delta = \begin{array}{c} \\ R1 \\ R2 \\ R3 \\ R4 \\ R5 \\ R6 \end{array} \begin{array}{c} \begin{array}{cccccc} D1 & D2 & D3 & D4 & D5 & D6 \end{array} \\ \begin{bmatrix} 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 2 & 3 & 4 & 5 & 6 \\ 5 & 4 & 5 & 4 & 5 & 6 \\ 5 & 4 & 5 & 4 & 5 & 4 \\ 7 & 6 & 7 & 6 & 7 & 6 \\ 7 & 6 & 7 & 6 & 7 & 6 \end{bmatrix} \end{array}$$

After the initial formation and target formation are determined, one can do the task assignment process. We have developed two algorithms for task assignment process [3-4]. The algorithm in [3] is simple but cannot guarantee the obtained result being with shortest time. When there is 0's in the distance matrix, the assignment result will often be wrong. Hence we developed a new algorithm, which is more complex but with better result, in [4]. However, there is still no mathematical proof for the optimality. Hence, in this paper, we develop a new task assignment algorithm, with which we believe that the shortest time can be guaranteed, as in the following.

Step 1: Use the algorithm in [3] or [4] to determine the first assignment for the original distance matrix, Δ_1. Determine the necessary moving time t_1 for the reformation.

Step 2: Replace the elements greater than or equal to t_1 of the distance matrix by a sufficiently large number Z, e.g. 999. This forms the 2nd distance matrix.

Step 3: Use the Hungarian algorithm [5] to check the minimum working time T_2 for the 2nd distance matrix Δ_2. If $T_2 >= Z$, then the obtained task assignment is the optimal one, else add 1 to the indices ofΔ and t, and go to Step 1.

Because the Hungarian algorithm is a well-known method for solving minimum-cost task assignment problem, and the sufficiently large number Z is much larger than the required time for the task, the check can guarantee that there is no better alternative for the task assignment. Hence, each mobile robot will have its destination to form the overall new pattern.

There is one thing that should be noticed. The optimal task assignment is not unique. But one cannot find any other assignment way that will have a shorter time to complete the task.

After the target position determined, there may be more than one possible route to the target position. Refer to Fig.

2, both Routes 1 and 2 for Robot 1 are of distance 3. If both robots choose the dashed route, then they will collide after one step. If anyone of them waits a step to prevent the collision, then the task will be delayed for 1 unit of time. If any one and only one of them chooses the solid route, then there will be no collision and the task will be completed on time.

4. Experimental Results

Consider the example shown in Fig. 1. This means that the distance matrix is as equation (1).

$$\Delta_1 = \begin{bmatrix} 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 2 & 3 & 4 & 5 & 6 \\ 5 & 4 & 5 & 4 & 5 & 6 \\ 5 & 4 & 5 & 4 & 5 & 4 \\ 7 & 6 & 7 & 6 & 7 & 6 \\ 7 & 6 & 7 & 6 & 7 & 6 \end{bmatrix} \quad (1)$$

Use the algorithm in [3] to solve this task assignment problem and we will have:

Robot 1 → Destination 1, Robot 2 → Destination 3
Robot 3 → Destination 5, Robot 4 → Destination 2
Robot 5 → Destination 4, Robot 6 → Destination 6

The required shortest time t_1 for the pattern reformation is 6. Replace the elements greater than or equal to 6 of the Δ_1 by 99, and we have the second distance matrix as:

$$\Delta_2 = \begin{bmatrix} 3 & 4 & 5 & 99 & 99 & 99 \\ 3 & 2 & 3 & 4 & 5 & 99 \\ 5 & 4 & 5 & 4 & 5 & 99 \\ 5 & 4 & 5 & 4 & 5 & 4 \\ 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix} \quad (2)$$

Applying the Hungarian Algorithm on Δ_2, we have the minimum cost for this problem is 211 which is much more large than 99. Thus the obtained task assignment result is the best one. And the routes for the pattern reformation are shown in Fig. 3.

For this problem, there is another assignment way that will result in the same time 6 for the pattern reformation. This way is:

Robot 1 → Destination 1, Robot 2 → Destination 2
Robot 3 → Destination 5, Robot 4 → Destination 3
Robot 5 → Destination 6, Robot 6 → Destination 4

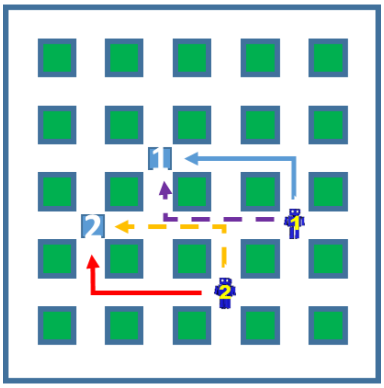

Fig. 2. Route select on route programming.

Consider another example shown in Fig. 4. There are 7 mobile robots on a 7x7 grid plane. The target positions are prescribed. The distance equation is:

$$\Delta_1 = \begin{bmatrix} 1 & 2 & 1 & 6 & 4 & 3 & 2 \\ 6 & 5 & 4 & 5 & 5 & 8 & 7 \\ 2 & 5 & 4 & 3 & 1 & 4 & 3 \\ 3 & 4 & 3 & 4 & 2 & 5 & 4 \\ 4 & 3 & 2 & 5 & 3 & 6 & 5 \\ 2 & 5 & 4 & 5 & 1 & 2 & 1 \\ 4 & 3 & 2 & 9 & 7 & 6 & 5 \end{bmatrix} \quad (3)$$

Use the algorithm in [3] to solve this task assignment problem and we will have:

Robot 1 → Destination 7, Robot 2 → Destination 4
Robot 3 → Destination 5, Robot 4 → Destination 1
Robot 5 → Destination 2, Robot 6 → Destination 6
Robot 7 → Destination 3

The required shortest time t_1 for the pattern reformation is 5. Replace the elements greater than or equal to 5 of the Δ_1 by 99, and we have the second distance matrix as:

$$\Delta_2 = \begin{bmatrix} 1 & 2 & 1 & 99 & 4 & 3 & 2 \\ 99 & 99 & 4 & 99 & 99 & 99 & 99 \\ 2 & 99 & 4 & 3 & 1 & 4 & 3 \\ 3 & 4 & 3 & 4 & 2 & 99 & 4 \\ 4 & 3 & 2 & 99 & 3 & 99 & 99 \\ 2 & 99 & 4 & 99 & 1 & 2 & 1 \\ 4 & 3 & 2 & 99 & 99 & 99 & 99 \end{bmatrix} \quad (4)$$

Applying the Hungarian Algorithm on (4), we have the minimum cost for this problem is 20 which is less than 99. Thus we have to use the algorithm in [3] to solve (4) and we will have:

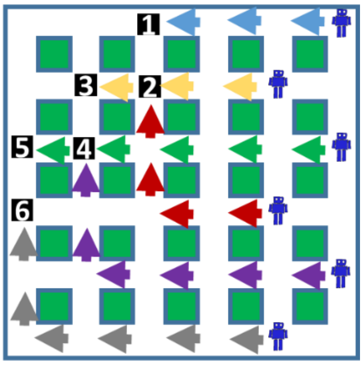

Fig. 3. Programmed routes for Example of Fig.1.

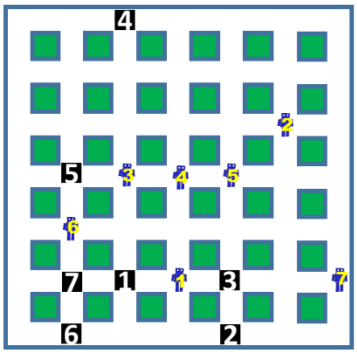

Fig. 4. Example of 7 mobile-robots on a 7x7 grid plane with the squares forming the target formation.

Robot 1 → Destination 7, Robot 2 → Destination 3
Robot 3 → Destination 4, Robot 4 → Destination 1
Robot 5 → Destination 5, Robot 6 → Destination 6
Robot 7 → Destination 2

The required shortest time t_1 for the pattern reformation is 4. Replace the elements greater than or equal to 4 of the Δ_1 by 99, and we have the third distance matrix as:

$$\Delta_3 = \begin{bmatrix} 1 & 2 & 1 & 99 & 99 & 3 & 2 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 2 & 99 & 99 & 3 & 1 & 99 & 3 \\ 3 & 99 & 3 & 99 & 2 & 99 & 4 \\ 99 & 3 & 2 & 99 & 3 & 99 & 99 \\ 2 & 99 & 99 & 99 & 1 & 2 & 1 \\ 99 & 3 & 2 & 99 & 99 & 99 & 99 \end{bmatrix} \quad (5)$$

Applying the Hungarian Algorithm on (5), we have the minimum cost for this problem is 114 which is greater than 99. Thus we can feel free that the shortest time for

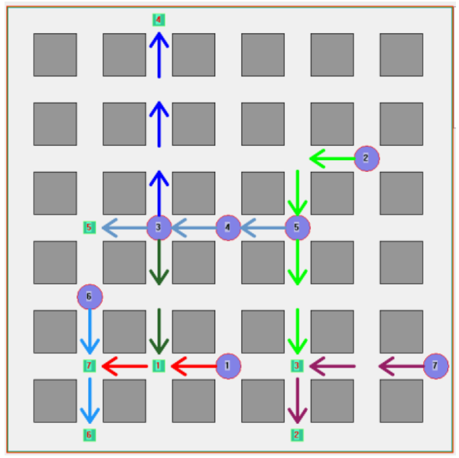

Fig. 5. Programmed routes for Example of Fig.4.

the task has been obtained. The programmed route in shown in Fig. 5.

5. Conclusions

In this paper, we developed a new algorithm, which combines our early developed algorithm with the Hungarian algorithm, to guarantee the task assignment result to be the best one. Since the Hungarian algorithm has been well applied on minimum cost programming problem, we believe that the proposed algorithm in this paper can be well applied on any size of grid plane.

References

1. E. W. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematik*, **1** (1959) 269-271.
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Section 24.3: Dijkstra's algorithm *in Introduction to Algorithms, Third ed.*, (MIT Press and McGraw–Hill, 2009) pp. 658–663.
3. K.H. Hsia, B.Y. Li and K.L. Su, Task assignment on pattern reformation for multiple mobile robots, *The nineteenth Int. Symp. on Artificial Life and Robotics 2014* (*AROB 19th 2014*) (Beppu, Japan, 2014), pp.438-441.
4. K.H. Hsia, B.Y. Li and K.L. Su (2015). Task assignment on grid plane for multiple mobile robots, *Tenth Int. Conf. on Innovative Computing, Information and Control* (*ICICIC2015*) (Dalian, China, 2015), p.193.
5. H.W. Kuhn, Variants of the Hungarian method for assignment problems, *Naval Research Logistics Quarterly*, **3** (1956) 253–258.