ICIC Express Letters
Volume **11**, Number **1**, January **2017**
ICIC International ©2017 ISSN 1881-803X
pp. 205–211

NAIVE RESEARCH ON PATTERN REFORMATION ON HEXAGONAL GRID PLANE

Kuo-Hsien Hsia
Department of Electrical Engineering
Far East University
No. 49, Zhonghua Rd., Xinshi Dist., Tainan City 74448, Taiwan
khhsia@cc.feu.edu.tw

Received June 2016; accepted September 2016

Abstract. *Pattern reformation problem is an interesting problem. A pattern reformation problem with a group of mobile robots moving on a squared grid plane has been studied for several years and a systematic algorithm for this problem has been developed. In this paper, the motion plane for the group of mobile robots is a hexagonal grid instead of a squared grid plane. The properties of hexagonal grid plane are different from squared grid plane. A distance matrix is defined and a simple algorithm has been proposed to solve the pattern reformation problem in the scope of task assignment. Examples are illustrated in this paper to show that the proposed algorithm can complete the task of pattern reformation on a hexagonal grid plane.*
Keywords: Hexagonal grid, Pattern reformation, Task assignment

1. **Introduction.** Pattern reformation problem is an interesting problem. In ancient war, a warring army often changed the battle array to confuse the enemy and thus could eventually win the game by a surprise. The match-past performance of the cheerleader parade is also an example of pattern reformation. For a pattern reformation problem, each member has to move from his current position to his target position as soon as possible. This is evidently a minimum-time problem. Usually, a minimization problem is to minimize the total cost or the total spending time. However, the overall task for a pattern reformation problem is completed only when all of the members reach their target positions. The member who needs the longest time to achieve his target position plays the key role in this problem. We must minimize this maximal spending time so that the pattern reformation problem has been solved. So we say that members of the team in a pattern reformation problem play a cooperative game [1]. For a cooperative game, the best of each individual does not mean the best of the overall group.

Without loss of generality, the motion plane for a pattern reformation problem could be assumed as a grid plane. In the past few years, we spent much time on studying the pattern reformation problem on squared grid plane [2-4]. The pattern reformation problem can be divided into two sub-problems: one is the task assignment problem, and the other one is the route programming problem. We supposed that there are some mobile robots moving on a squared grid plane. As mentioned earlier, the task assignment problem is a game problem. The target position of every mobile robot should be determined in this stage in order to have the maximal spending time for all mobile robots moving to their corresponding target position being minimal. A distance matrix is defined and a simple algorithm has been proposed to solve the task assignment problem. With our proposed algorithm, the solution of the game problem, task assignment problem of pattern reformation on squared grid plane, could be solved. Once the task assignment problem is solved, the route programming problem will be solved consequently since the distance matrix defined all possible routes for each mobile robot.

More and more researchers applied hexagonal grid cells on path finding problems. For example, Tsatcha et al. applied hexagonal meshes and iterative deepening A* algorithm on maritime routing problem [5]. Moser et al. applied hexagonal grid on the analysis of network mechanisms [6]. These motivate us to extend our previous studies to the scenario of hexagonal grid plane. Therefore, in this paper, we will focus on the pattern reformation problem for mobile robots moving on a hexagonal grid plane. The properties of hexagonal grid plane are quite different from squared grid plane. Consider the scenario as shown in Figure 1. There are 5 mobile robots on the plane. The plane is composed of hexagonal cells. The dotted positions form the target pattern of the group of mobile robots. Each robot can move in six directions to cross the edge to a neighboring cell, as shown in Figure 2. The question is: how do the mobile robots move to change their formation to the new formation assigned by the dots as soon as possible?

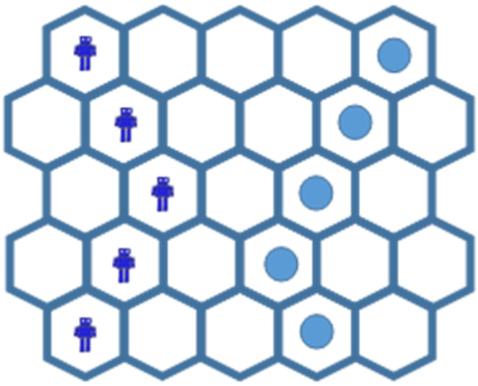

FIGURE 1. Scenario of the considered pattern reformation problem

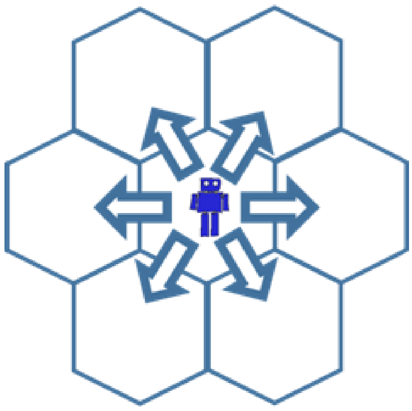

FIGURE 2. Moving direction of a robot

This paper is organized as follows. Some properties of hexagonal grids are illustrated in Section 2. Section 3 shows how to obtain the distance matrix for the problem and also the task assignment procedure. An example of pattern reformation on hexagonal grid plane is shown and solved in Section 4. Finally, some discussions and conclusions are made in Section 5.

2. **Properties of Hexagonal Grid.** Hexagonal grid plane, like honeycomb, is a common planar structure. However, it is not so straightforward as squared grid plane. For a squared grid plane, each cell has 4 neighbors and the moving path forms squares on the plane. However, each cell on a hexagonal grid plane has 6 neighbors and the moving paths form triangles on the plane, as shown in Figure 3.

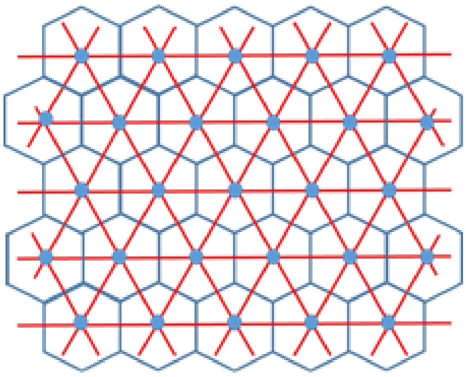

Figure 3. Moving paths on a hexagonal grid plane

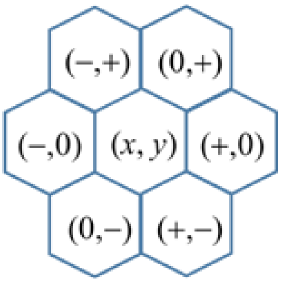

Figure 4. Coordinate change for a hexagonal grid plane

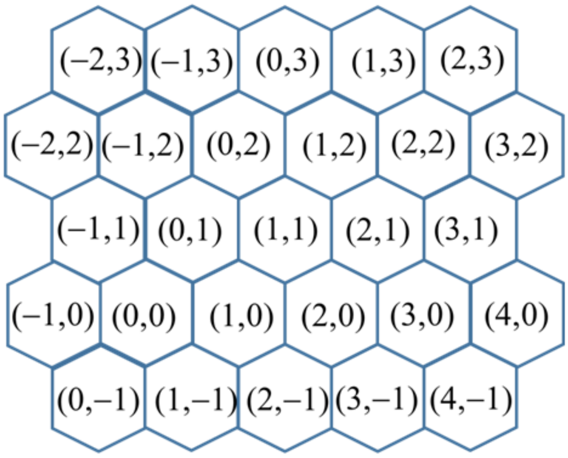

Figure 5. Coordinates of some cells on a hexagonal grid plane

There are several coordinate systems for a hexagonal grid plane, and the most common approach is to offset every other column or row. One can either offset the odd or the even column/rows, so the horizontal and vertical hexagons each have two variants [7]. Suppose that we want to coordinate the hexagonal grid plane like Figure 3, we can use an ordered pair (x, y) to represent the coordinate of each cell. A move directly right or left means a positive or negative change in the x coordinate only, and a move to the upper-right means a positive change in the y coordinate only. Thus, a move to the upper-left means a negative change in the x coordinate and a positive change in the y coordinate [8]. The change in coordinates, from any cell, can be represented like Figure 4, where 0 means unchanged, + means incremental and – means decremental. The coordination of a portion of hexagonal grid plane is shown in Figure 5.

For this coordinate system, the distance d (moving steps) between two cells (x_1, y_1) and (x_2, y_2) is given by

$$d = \max\left(|x_2 - x_1|,\ |y_2 - y_1|,\ |(x_2 + y_2) - (x_1 + y_1)|\right). \tag{1}$$

For example, the distance between $(0, -1)$ and $(3, 1)$ is $\max(|3 - 0|, |1 - (-1)|, |(3 + 1) - (0 + (-1))|) = \max(3, 2, 5) = 5$. We can verify this from Figure 5 that it needs 5 steps to move from $(0, -1)$ to $(3, 1)$.

If we straighten the saw-tooth-like line segments on a hexagonal grid plane, we will have an interlocked squared grid plane, as shown in Figure 6. It looks like squared grid plane but with every other row being shifted half grid. For any cell on an interlocked squared grid plane, there are also 6 moving directions. This is the same as moving on a hexagonal grid plane. Hence, we can claim that all properties of hexagonal grid plane are suitable for interlocked squared grid plane.

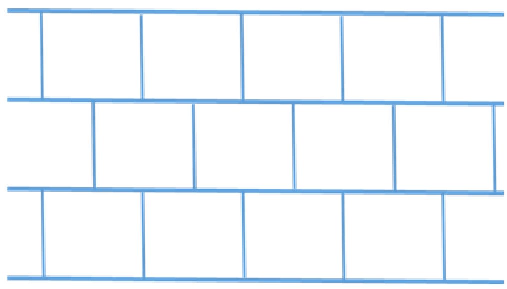

FIGURE 6. Interlocked squared grids

3. **Distance Matrix and Task Assignment Procedure.** Once the initial and the target patterns being determined, the distance from any one robot to any possible target position can be obtained by (1). Assigning an ID to every robot and every target position, one can easily form a distance matrix Δ where the row represents the robot and the column represents the target position. Hence the element $\Delta_{i,j}$ means the distance from the position of Robot i to Target j. For the case shown in Figure 1, let the robot from top to down be numbered from R1 to R5, respectively. And the target position from top to down is numbered from T1 to T5. Then we can have the distance matrix Δ as:

$$\Delta = \begin{array}{c} \\ \mathrm{R1} \\ \mathrm{R2} \\ \mathrm{R3} \\ \mathrm{R4} \\ \mathrm{R5} \end{array} \begin{array}{c} \begin{array}{ccccc} \mathrm{T1} & \mathrm{T2} & \mathrm{T3} & \mathrm{T4} & \mathrm{T5} \end{array} \\ \begin{bmatrix} 4 & 4 & 4 & 4 & 5 \\ 4 & 3 & 3 & 3 & 4 \\ 4 & 3 & 2 & 2 & 3 \\ 5 & 4 & 3 & 2 & 3 \\ 6 & 5 & 4 & 3 & 3 \end{bmatrix} \end{array} \tag{2}$$

The task assignment procedure is summarized as follows.

Step 1: Number the mobile robots and target positions.

Step 2: Find the first distance matrix Δ_1.

Step 3: Find the minimum, the second minimum, number of the second minimum, and the sum for each column and row of the distance matrix. If there are double minimum in some column or row, then set the second minimum the same as the minimum.

Step 4: If the maximum of the second minimums is not repeated, then pivot the column/row and go to Step 8.

Step 5: If the maximum of the column/row-sum is not repeated, then pivot the column/row and go to Step 8.

Step 6: If the maximum of the second minimums of the columns or rows satisfying the condition of Step 5 is not repeated, then pivot the column/row and go to Step 8.
Step 7: Pivot the larger maximum of the minimums of the columns or rows satisfying the condition of Step 6. If there are multiple columns or rows satisfying the condition, choose the first column/row in the searching order.
Step 8: Circle any one of the nonzero minimum elements of the pivoted column/row. And discard the column and row where the circled element located.
Step 9: Repeat Steps 3-8 until all rows or columns are discarded.
Step 10: The maximum of the circled elements, symbolled as τ_1, is the temporary task completing time. Replace the elements greater than or equal to τ_1 of the distance matrix by a sufficiently large number Z, e.g., 99. This forms the 2nd distance matrix Δ_2.
Step 11: Use the Hungarian algorithm [9] to check the minimum working time t_2 for Δ_2. If $t_2 >= Z$, then stop and the obtained task assignment is the optimal one, i.e., the temporary task completing time is the optimal task completing time; else go to Step 3 and find the next τ.

The sum of each column/row in Step 3 is not necessary for each iteration since it is possible that the requirement of Step 4 may be met. For the case of Figure 1 with distance matrix (2), the optimal task completion time is 4. One of the possible task assignments is: R1→T1, R2→T2, R3→T3, R4→T4, and R5→T5.

4. **Illustrated Example.** Consider the example shown in Figure 7. There are 5 mobile robots moving on a $5 \times 5 + 2$ hexagonal grid plane. The dotted cells represent the target positions. At first, we number the robots from left to right and from top to bottom, and the target positions in counter-clockwise. Then we have the first distance matrix as:

$$\Delta_1 = \begin{bmatrix} 5 & 4 & 4 & 5 & 6 \\ 4 & 3 & 3 & 4 & 5 \\ 4 & 3 & 3 & 4 & 5 \\ 3 & 2 & 2 & 3 & 4 \\ 5 & 4 & 3 & 4 & 5 \end{bmatrix} \tag{3}$$

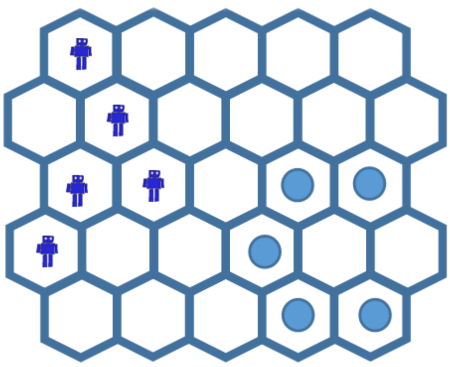

FIGURE 7. Example of 5 mobile robots on a hexagonal grid plane

The operations of finding the optimal task assignment are shown in Figure 8. The first iteration is shown with solid line, and the dashed line and dotted line are for the 2nd and 3rd iterations, respectively. The relative conditions of the pivoted column/row are marked with squares. For the 1st and 3rd iterations, the sum of column/row is not necessary since the maximum of the 2nd minimum is unique for each of these two iterations. Now we

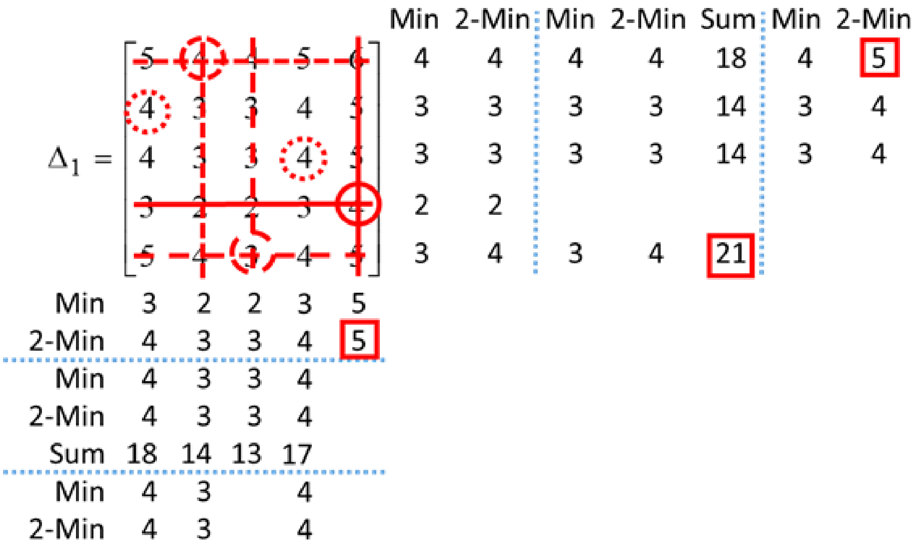

FIGURE 8. Operations on 1st distance matrix

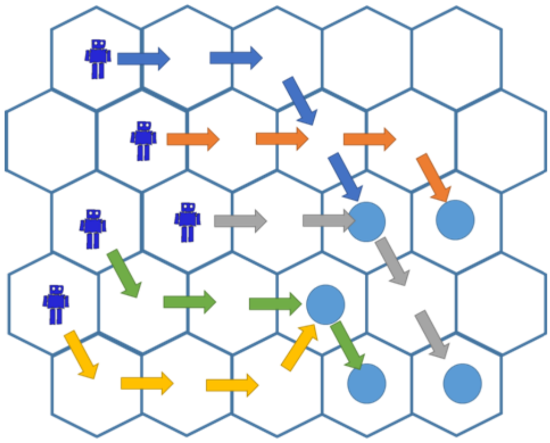

FIGURE 9. Trajectories of the mobile robots on the hexagonal grid plane

are in Step 10 with the maximum of the circled elements being 4. Replace all elements greater than or equal to 4 of the distance matrix by 99. We have:

$$\Delta_2 = \begin{bmatrix} 99 & 99 & 99 & 99 & 99 \\ 99 & 3 & 3 & 99 & 99 \\ 99 & 3 & 3 & 99 & 99 \\ 3 & 2 & 2 & 3 & 99 \\ 99 & 99 & 3 & 99 & 99 \end{bmatrix} \tag{4}$$

Applying Hungarian algorithm to (4), the minimum working time t_2 is $206 > 99$. This means that the result is an optimal solution. The best solution for the pattern reformation is: R1→T2, R2→T1, R3→T4, R4→T5, and R5→T3. And the optimal completion time is 4. The moving trajectories of the mobile robots are shown in Figure 9 after route programming. Here the Robot 5 moves to the Target 3 spending 4 steps to prevent colliding with Robot 3.

5. **Conclusions.** In this paper, we showed a naïve research result of pattern reformation on hexagonal grid plane which is much more complex than that on squared grid plane. Because the properties of hexagonal grid plane are different from squared grid plane, the solving process for the pattern reformation problem on hexagonal grid plane is slightly

different from that on squared grid plane. We also apply the Hungarian algorithm to guaranteeing the optimality of the solution of this problem.

Properties of a hexagonal grid plane are also suitable for an interlocked squared grid plane. Results obtained in this paper can be applied on an interlocked squared grid plane.

This paper is only a naïve research. We mainly focus on the task assignment process for this problem in this paper. More examinations on the route programming results of the pattern reformation problem are needed. Moreover, we will extend the results to more complex situations in the near future.

REFERENCES

[1] M. D. Davis, *Game Theory: A Nontechnical Introduction*, Dover Publications, New York, 1997.

[2] K. Hsia, B. Li and K. Su, Task assignment on pattern reformation for multiple mobile robots, *The 19th International Symposium on Artificial Life and Robotics*, Beppu, Japan, pp.438-441, 2014.

[3] K. H. Hsia, B. Y. Li and K. L. Su, Task assignment on grid plane for multiple mobile robots, *ICIC Express Letters, Part B: Applications*, vol.6, no.12, pp.3327-3332, 2015.

[4] K. H. Hsia, B. Y. Li and K. L. Su, Task assignment and route programming for pattern reformation on grid plane, *Journal of Robotics, Networking and Artificial Life*, vol.2, no.4, pp.221-225, 2016.

[5] D. Tsatcha, E. Saux and C. Claramunt, A bidirectional path-finding algorithm and data structure for maritime routing, *International Journal of Geographical Information Science*, vol.28, no.7, pp.1355-1377, 2014.

[6] E. I. Moser, M. B. Moser and Y. Roudi, Network mechanisms of grid cells, *Philosophical Trans. of the Royal Society of London B: Biological Sciences*, vol.369, no.1635, 2013.

[7] *Hexagonal Grids – From Red Blob Games*, http://www.redblobgames.com/grids/hexagons/, [Mar. 2013, updated in Mar. 2015; cited Nov. 28, 2016].

[8] C. Schetter, *Hexagon Grids: Coordinate Systems and Distance Calculations*, http://keekerdc.com/2011/03/hexagon-grids-coordinate-systems-and-distance-calculations/, [Mar. 2011; cited Apr. 4, 2016].

[9] D. Bruff, *The Assignment Problem and the Hungarian Method*, http://www.math.harvard.edu/archive/20_spring_05/handouts/assignment_overheads.pdf, [cited Jul. 20, 2016].