Finding Plans for Rearranging Robots in θ-like Environments *

Pavel Surynek

Charles University, Faculty of Mathematics and Physics,
Department of Theoretical Computer Science and Mathematical Logic,
Malostranské náměstí 2/25, 118 00 Praha 1, Czech Republic
pavel.surynek@mff.cuni.cz

Abstract

An overview of a problem of rearranging robots in θ-like environments is given in this short paper. Generally, the problem consists in finding a sequence of moves that transforms a given initial arrangement of robots in the vertices of a given graph into a desired goal arrangement. The graph is used as an abstraction of the environment. A special case of the problem with θ-like graphs is reported in this paper.

Introduction and Motivation

The problem addressed in this paper is a task of finding a sequence moves for a group of robots that need to move from a given initial positions to a given goal positions in a certain environment. This task is called a *multiple-robot path planning problem.* The environment in which the robots move is modeled as an undirected graph where robots are placed in the vertices (a robot can move to an adjacent unoccupied vertex). We are particularly dealing with an extreme case where there is only one unoccupied vertex in the graph. This case is in fact a generalization of a well known *15-puzzle* (Ratner & Warmuth, 1986; Wilson, 1974).

Many problems from the industrial practice can be viewed as instances of multi-robot path planning (storage yards operations, heavy traffic optimization). Moreover, many problems from virtual spaces can be also modeled as multi-robot path planning (to name an example consider a planning of data transfers with limited size of the cache memory at communication nodes). Thus, we consider the problem to be important not only theoretically.

The rest of the paper is dedicated to the description of a case of the problem with so called *θ-like graphs*. This case is interesting since it forms the most complicated sub-problem in general variants of the problem (Kornhauser et al., 1984) In addition, the solving procedure for this case can be used as a building block for a general solver.

Problem of Path Planning for Multiple Robots

Consider a group of robots in a certain environment that need to move from their initial positions to the given goal positions. The robots are required to avoid collisions. Thus, the task is to find spatial-temporal paths from the initial to the goal position for each robot such that these paths do not intersect. The environment is modeled as an undirected graph where the robots are placed in the vertices.

A robot can move from a vertex to a target adjacent vertex if there is no robot in the target vertex and no other robot is simultaneously entering the target vertex. The problem of *path planning for multiple robots* is formally defined as follows.

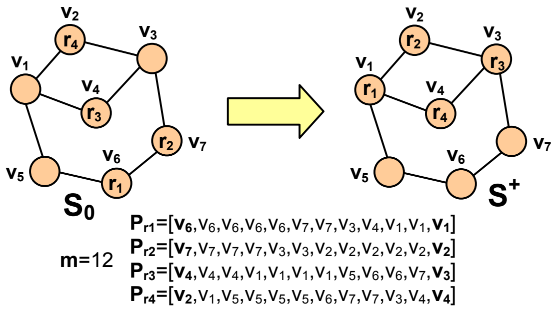

Figure 1: Illustration of the problem of path planning for multiple robots. The task is to move robots from their initial positions denoted as S_0 to the goal positions denoted as S^+. A solution of length 12 is shown. Notice, that the solution contains parallel movements.

Let us have an undirected graph $G = (V, E)$ that models the environment. Next, let us have a set of robots $R = \{r_1, r_2, \ldots, r_\mu\}$ where $\mu < |V|$. The initial positions of the robots are defined by a function $S_0 : R \rightarrow V$ where $S_0(r_i) \neq S_0(r_j)$ for $i, j = 1,2,\ldots,\mu$ with $i \neq j$. The goal positions of the robots are defined by a function $S^+ : R \rightarrow V$ where $S^+(r_i) \neq S^+(r_j)$ for $i, j = 1,2,\ldots,\mu$ with $i \neq j$. The problem of *path-planning for multiple robots* is a task to find a number m and a path $P_r = [p_1^r, p_2^r, \ldots, p_m^r]$ for every robot $r \in R$ where $p_i^r \in V$ for $i = 1,2,\ldots,m$, $p_1^r = S_0(r)$, $p_m^r = S^+(r)$, and either $\{p_i^r, p_{i+1}^r\} \in E$ or $p_i^r = p_{i+1}^r$ for $i = 1,2,\ldots,m-1$. Furthermore, paths $P_r = [p_1^r, p_2^r, \ldots, p_m^r]$ and $P_q = [p_1^q, p_2^q, \ldots, p_m^q]$ for every two robots $r \in R$ and $q \in R$ such that $r \neq q$ must satisfy a condition that $p_{i+1}^r \neq p_i^q$ for $i = 1,2,\ldots,m-1$ (the target vertex is unoccupied) and $p_i^r \neq p_i^q$ for $i = 1,2,\ldots,m$ (no other robot is simultaneously entering the target vertex).

The problem of path planning for multiple robots is illustrated in figure 1. Notice that if there is more than one unoccupied vertex, parallel movements of robots are al-

*This work is supported by the Czech Science Foundation (201/05/H014).

lowed by the definition. An optimization variant of the problem requires m to be smallest possible. However, Ratner and Warmuth (1986) showed that the decision version of this problem is *NP*-complete while without the requirement on the optimality of m it is in *P*. The specific case with a θ-like graph is similar to a so called *top-spin puzzle* (Felner et al., 2007).

A Special Case with θ-like Graph

We are particularly dealing with the case of the problem of path planning for multiple robots with *θ-like graphs* modeling the environment where there is only one unoccupied vertex (that is, $\mu = |V| - 1$). Moreover, we are interested in short solutions (that is, we require m to be small but not necessarily optimal).

A *θ-like graph* is an undirected graph $G_\theta(a,b,c) = (V_\theta, E_\theta)$ where $a,b,c \in \mathbb{N} \wedge b \geq 2$ are parameters, $V_\theta = \{x_1, x_2, \ldots, x_a, y_1, y_2, \ldots, y_b, z_1, z_2, \ldots, z_c\}$, and $E_\theta = \{\{x_1,x_2\},\ldots,\{x_{a-1},x_a\},\{y_1,y_2\},\ldots,\{y_{b-1},y_b\}, \{z_1,z_2\},\ldots, \{z_{c-1},z_c\}, \{z_1,z_2\},\ldots,\{z_{c-1},z_c\}, \{x_1,y_1\},\{x_a,y_b\}, \{y_1,z_1\}, \{y_b,z_c\}\}$. An example of θ-like graph is shown in figure 2.

An instance of the multi-robot path planning problem we are interested in consists of a θ-like graph $G_\theta(a,b,c)$ where the vertex y_1 is unoccupied and the goal arrangement of robots is made from the original one by exchanging a pair of robots. The task is to find a short solution. An example of the instance of this case of the problem is shown in figure 2.

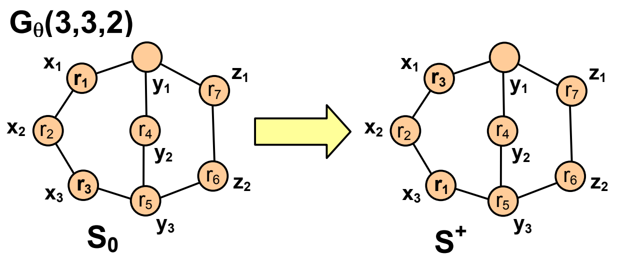

Figure 2: Illustration of the special case of the problem of path planning for multiple robots with θ-like graph. The task is to exchange robots r_1 and r_3 using the smallest possible number of moves.

The General Case via the Special Case

We plan to apply the knowledge of solutions of the special cases for solving the general variant of the problem. Any goal arrangement of robots in a θ-like graph can be reached by exchanging appropriate pairs of robots. Hence, a solution to any problem on a θ-like graph can be assembled by concatenating solutions for the defined special case.

In each θ-like graph $G_\theta(a,b,c) = (V_\theta, E_\theta)$ there is a quadratic number of pairs of robots (namely, $(|V_\theta|-1)(|V_\theta|-2)/2$ pairs robots). Hence, it seems to be feasible to try to find optimal solutions for all the robot exchanges **off-line** in all the small θ-like graphs and store them in a database. Optimally, the **on-line** solving algorithm then just looks into the database and concatenates the found records to form the overall solution. At most $(|V_\theta|-1)$ look-ups are necessary since any arrangement of robots can be reached using at most $(|V_\theta|-1)$ exchanges.

The sketched approach prefers short overall solutions since they are concatenated from the optimal solutions for the sub-problems. The length of the solution directly corresponds to the execution time and hence its short length is out objective. If the input θ-like graph is not preprocessed in the database we may use a fast alternative solving process from (Kornhauser et al., 1984). However, this solving method does not prefer short solutions. Our preliminary experiments showed that solutions constructed of the optimal solutions of the sub-problems are about **10 times shorter** than that produced by the method from (Kornhauser et al., 1984).

Solving Algorithm for θ-like Environments

Although the construction of the database of the optimal solutions for exchanging robots is created off-line we still need an efficient algorithm for this task.

We propose an algorithm based on *IDA** enhanced with a learning heuristic. The arrangement of robots in vertices of the graph can be regarded as a permutation of robots. It is possible to identify the minimum number of moves that are necessary to transform one permutation of robots into another during the search and this information can learned on-line. This knowledge can be subsequently exploited for pruning the state space. If the minimum number of steps for reaching the goal permutation plus the current depth exceeds the depth limit, the current branch of the search can be pruned.

Our preliminary experiments showed that the proposed algorithm is faster than *iterative deepening*, *A**, and *IDA** (with the standard distance based heuristics). Our learning *IDA** is more than **10 times faster** than the *IDA** while the other two algorithms are completely uncompetitive.

References

Felner, A., Korf, R. E., Meshulam, R., Holte, R. C., 2007. *Compressed Pattern Databases.* Journal of Artificial. Intelligence Research (JAIR), Volume 30, pp. 213-247, AAAI Press.

Kornhauser, D., Miller, G. L., Spirakis, P. G., 1984. *Coordinating Pebble Motion on Graphs, the Diameter of Permutation Groups, and Applications.* Proceedings of the 25th Annual Symposium on Foundations of Computer Science (FOCS 1984), pp. 241-250, IEEE Press.

Ratner, D., Warmuth, M. K., 1986. *Finding a Shortest Solution for the N × N Extension of the 15-PUZZLE Is Intractable.* Proceedings of the 5th National Conference on Artificial Intelligence (AAAI 1986), pp. 168-172, Morgan Kaufmann Publishers.

Wilson, R. M., 1974. *Graph Puzzles, Homotopy, and the Alternating Group.* Journal of Combinatorial Theory, Ser. B 16, pp. 86-96, Elsevier.