Making Solutions of Multi-robot Path Planning Problems Shorter Using Weak Transpositions and Critical Path Parallelism

Pavel Surynek

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

Abstract

Two techniques for shortening solutions of multi-robot path planning problem are presented in this paper. Consider a group of mobile robots moving in a certain environment in which they must avoid obstacles and each other. The task is to move each robot from its initial position to the given goal position. An abstraction of the problem where the environment with obstacles is modeled as an undirected graph is adopted. Each robot is placed in a vertex of the graph and it can move to the neighboring vertex if no other robot is simultaneously entering the same vertex. A particular objective in multi-robot path planning is the short length of the solution (sequence of moves). Since the task of finding the shortest possible solution is intractable, alternative techniques for shortening the solutions are of interest. The first presented solution shortening technique is based on the use of optimal macros of which the overall solution is composed. The second technique is trying to increase parallelism within the resulting solutions by the method of critical path. Both shortening techniques proved to be beneficial according to the performed experimental evaluation.

Introduction and Motivation

This paper presents two techniques for improving (shortening) solutions of multi-robot path planning problems (Ryan, 2007; Surynek, 2009a). Consider a group of mobile robots moving in a certain environment where each robot needs to move from its initial position to the given goal position. The robots must avoid obstacles and each other. An abstraction where the environment with obstacles is modeled as an undirected graph is adopted throughout this work. This abstraction of the task is widely known as a problem of *multi-robot path planning* (Ryan, 2007). It ranks among the most challenging problems of artificial intelligence (Russel & Norvig, 2003) and it motivates efforts of theorists as well as practicians.

A particular objective in multi-robot path planning is short length of solutions. Since the task of finding the shortest possible solution is intractable (Ratner & Warmuth, 1986), alternative techniques for shortening the solutions are of interest.

The primary motivations for the problem of multi-robot path planning are tasks of moving objects within a limited free space. These tasks include rearranging containers in storage yards, coordination of movements of a large group of automated agents, or optimization of dense traffic. However, this is not the only motivation. Many tasks from virtual spaces can be also viewed as problems of path planning for multiple robots. An example may be data transfer with limited buffers at communication nodes, a coordination of a group of agents in strategic computer games, or planning movements in mass scenes in computer-generated imagery.

This work builds upon existing sub-optimal methods for generating solutions of multi-robot path planning problems. The **first technique** for shortening the solution is based on using optimal macros (solution of a sub-problem) of which the overall (sub-optimal) solution is composed. This approach is very similar to that proposed in (Surynek, 2009b). However, in this work, optimal macros satisfying weaker conditions are used for composing overall solutions. The weaker condition that the macro has to satisfy allows the macro to be shorter.

The **second technique** is trying to increase parallelism within the solution of multi-robot path planning problem. The basic observation is that motions of robots not interfering with each other (*independent* motions) can be performed in parallel. The use of *critical path method* (Russel & Norvig, 2003) is suggested for determining what motions can be performed in parallel. This method for shortening solutions is intended as post-processing step applied on the generated solution.

This paper formally describes the problem of multi-robot path planning first. Then techniques for shortening the solutions are presented. And finally, some initial experiments illustrating the benefits of suggested techniques are presented.

Path Planning for Multiple Robots

The problem of *path planning for multiple robots* consists in finding a sequence of moves for rearranging robots in a certain environment. The robots occupy certain positions in the environment initially. The task is to move robots to the given goal positions. The robots must avoid obstacles in the environment and must not collide with each other.

The widely accepted formalization of the multi-robot path planning problem models the environment by an undirected graph (Ryan, 2007). The vertices of this graph represent positions in the environment and edges represent unblocked way from one vertex (position) to another vertex. Robots are placed in vertices of the graph while at least one vertex remains unoccupied. A robot can move from a vertex to the neighboring target vertex if no other robot is simultaneously entering the target vertex. The formal definition of the problem is given in the following definition. An example of multi-robot path planning problem and its solution is shown in figure 1.

Definition 1 ***(path planning for multiple robots).*** Let us have an undirected graph $G=(V,E)$ where $V=\{v_1,v_2,\dots,v_n\}$ that models the environment. Next, let us have a set of robots $R=\{r_1,r_2,\dots,r_\mu\}$ where $\mu<n$. 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 $i,j=1,2,\dots,\mu$ 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 $i,j=1,2,\dots,\mu$ with $i\neq j$. The problem of *path planning for multiple robots* is a task to find a number m and a path $P_r=[p_1^r,p_2^r,\dots,p_m^r]$ for every robot $r\in R$ where $p_i^r\in V$ for $i=1,2,\dots,m$, $p_1^r=S_0(r)$, $p_m^r=S^+(r)$, and either $\{p_i^r,p_{i+1}^r\}\in E$ or $p_i^r=p_{i+1}^r$ for $i=1,2,\dots,m-1$. Furthermore, paths $P_r=[p_1^r,p_2^r,\dots,p_m^r]$ and $P_q=[p_1^q,p_2^q,\dots,p_m^q]$ for every two robots $r\in R$ and $q\in R$ such that $r\neq q$ must satisfy that $p_i^r\neq p_i^q$ for $i=1,2,\dots,m$ (no other robot is simultaneously entering the target vertex). □

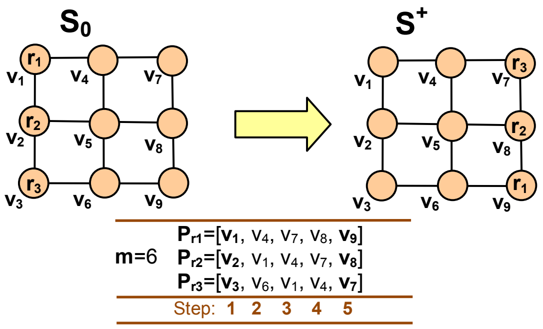

Figure 1. *A problem of path planning for multiple robots.* The task is to move robots from their initial positions denoted as S_0 to the goal positions denoted as S^+. A solution of length 5 is shown. Notice the parallelism within the solution.

Multi-robot path planning with bi-connected graphs modeling the environment and with one unoccupied vertex represents an important class of the problem (Surynek, 2009a). This is due to the fact that they are almost always solvable (almost all the goal arrangements of robots are reachable from the given initial arrangement). Moreover, the single unoccupied vertex in the graph induces the most constrained situation (more unoccupied vertices simplifies the situation).

A bi-connected graph is an undirected graph with at least three vertices that remains connected after removal of any vertex. A well known property of bi-connected graphs (West, 2000) that is utilized in solving algorithms is that every bi-connected graph can be constructed from a cycle by adding a sequence of loops. The structurally simplest bi-connected graph in that the multi-robot path planning problem is non-trivial is a graph consisting of a cycle and a single loop (in a graph consisting of a single cycle, robots can be only rotated around the cycle; but arrangements requiring exchange of robots are not reachable). This type of bi-connected graph is called a θ-like graph. It is formally defined in the following definition. An example of θ-like graph is shown in figure 2.

Definition 2 ***(θ-like graph).*** Let $A=\{x_1,x_2,\dots,x_a\}$, $B=\{y_1,y_2,\dots,y_b\}$, and $C=\{z_1,z_2,\dots,z_c\}$ be the finite sets (of vertices) where $|A|\geq 1 \wedge |B|\geq 2 \wedge |C|\geq 1$. A *θ-like graph* $G_\theta(A,B,C)=(V_\theta,E_\theta)$ is an undirected graph where $V_\theta = A\cup B\cup C$ and $E_\theta = \{\{x_1,x_2\},\dots,\{x_{a-1},x_a\},\{y_1,y_2\},\dots,\{y_{b-1},y_b\}, \{x_1,y_1\},\{x_a,y_b\},\{y_1,z_1\}, \{x_1,y_1\},\{x_a,y_b\}, \{y_1, z_1\}, \{y_b,z_c\}\}$. □

Solutions of multi-robot path planning problems over θ-like graphs are used as macros of which the solution of every problem over bi-connected graph can be constructed (Surynek, 2009b).

Solving Algorithm for the Problem

There already exist several algorithms for multi-robot path planning. As is it stated in the introduction, the purpose of this paper is to report some improvement techniques. These techniques cooperate with existing algorithms. Thus let us recall these algorithms briefly.

Bi-connected Graphs with Two Unoccupied Vertices

As the basic starting point, the *BIBOX* algorithm (Surynek, 2009a) for problems with bi-connected graphs modeling the environment is used.

The original version of the *BIBOX* algorithm requires at least **two** unoccupied vertices. Suppose that the graph modeling the environment was decomposed into loops and the initial cycle. The algorithm works in two phases. In the first phase, robots are placed to their goal positions in all the loops of the decomposition except the first loop and the original cycle (which form a θ-like graph). Having a loop of the decomposition, robots in this loop are placed in the stack-like manner. A new robot comes at the beginning of the loop and then all the robots are pushed by the rotation of the loop deeper into the loop. The final push moves the robots to their right positions. After finishing the loop, the problem of the same type but smaller is obtained. The task is to place robots in the bi-connected graph which is now one loop smaller.

Except the original θ-like graph, the process suffices with a **single** unoccupied vertex. The situation is more complicated with the original θ-like graph. The standard *BIBOX* algorithm uses a process of exchanging pair of robots using **two** unoccupied vertices. Together with the fact that every permutation can be composed of exchanging pairs of elements we are able to obtain every permutation of robots in the original θ-like graph. This version of the *BIBOX* algorithm is described in (Surynek, 2009a).

Bi-connected Graphs with One Unoccupied Vertex

The *BIBOX* algorithm has been further improved to suffice with only **one** unoccupied vertex in the last phase. These improvements can be found in (Surynek, 2009b). The paper addresses search for the shortest possible solutions of multi-robot path planning problems in θ-like graphs where the initial and the goal arrangements of robots differs by a single exchange of a pair of robots (it called a **transposition case**) or by a single **rotation along a 3-cycle** (sequence of three vertices). In both cases a **single** unoccupied vertex is supposed.

The situation with an exchange of pair of robots (transposition case) is solvable if there is an odd cycle in the θ-like graph. The situation with the rotation along a 3-cycle is always solvable (Kornhauser *et al.*, 1984).

The well known fact is that it is possible to compose any permutation by using the linear number of transpositions of pairs of elements. It is also possible to compose any even permutation by the linear number of rotations along 3-cycles. Using these facts a solution to any solvable multi-robot path planning problem in the θ-like graph can be composed of the **optimal solutions** (*macros*) to problems in the same graph representing transposition or 3-cycle rotation cases. Since the macros used for composition of the overall solution are optimal it is expectable that the overall solution is of good quality (short). This claim has been experimentally shown in (Surynek, 2009b).

Solution Shortening Technique Using Weak Transpositions

The significant drawback of using optimal solutions for transposition cases and 3-cycle cases is that all the other robots are forced to stay at their original positions in θ-like graph. This is relatively strong requirement (constraint).

Actually, it is only required not to disarrange robots in the already arranged part of the graph. The arranged part of the graph incrementally grows and finally covers the whole θ-like graph. Observe that in the course of placing robots to their goal positions it is not necessary to care about the large parts of the graph. This observation leads to the concept of a so called *weak transposition case* of the problem. A weak transposition case is a modification of multi-robot path planning problem where a specified pair of robots must be exchanged while a specified subset of robots are required to stay in their positions and the goal positions of the rest of robots is arbitrary. The task is to find optimal solution of this case. The formal definition of weak transposition case is given in the following definition. The example of weak transposition case is shown in figure 2.

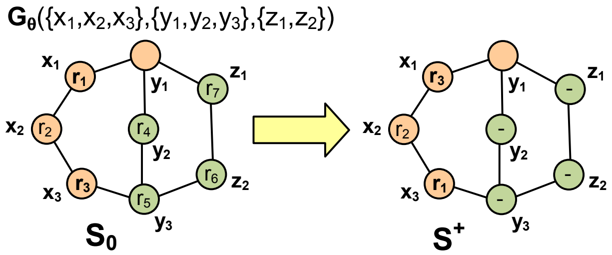

Figure 2: *A weak transposition case of the problem of path planning for multiple robots* with a θ-like graph. The task is to transpose robots r_1 and r_3 using the smallest possible number of moves, while there is not constraint (except that y_1 must be left unoccupied) on goal positions of robots r_4, r_5, r_6, and r_7.

Definition 3 ***(weak transposition case).*** Suppose a multi-robot path planning problem with a θ-like graph $G_\theta(A,B,C)=(V_\theta,E_\theta)$ with a set of robots $R=\{r_1,r_2,\ldots,r_\mu\}$ where $\mu=|V_\theta|-1$ and with a simple function $S_0:R\rightarrow V$ determining the initial positions of robots. Let $r,q\in R$ be two distinct robots and let $\{p_1,p_2,\ldots,p_k\}\subseteq R$ be a set of robots such that $\{r,q\}\cap\{p_1,p_2,\ldots,p_k\}=\emptyset$. A goal $S^+:R\rightarrow V$ where $S^+(q)=S_0(r)$, $S^+(r)=S_0(q)$, and $S^+(p_i)=S_0(p_i)$ for $i=1,2,\ldots,k$ forms a *weak transposition case.* □

Again, it is possible to compose a solution to any problem in a θ-like graph using weak transposition cases. Since the constraint on the goal arrangement of robots is weaker than in the case of standard transposition case and 3-cycle rotation case it is justly expectable that the optimal solution will be significantly shorter. This claim is experimentally supported in the experimental evaluation section.

Solution Shortening Technique Using Critical Path Parallelism

The second technique for shortening solutions of multi-robot path planning problems is based on increasing parallelism within the problem. Observe that the definition of the problem intrinsically allows parallelism (see figure 1). However, all the referred solving algorithms produce sequential solutions in the form of sequences of moves between neighboring vertices. This is extremely time-wasting in the situation when there is higher number of unoccupied vertices in the graph modeling the environment.

This disadvantage can be overcome by an application of *critical path method* (Russel & Norvig, 2003). A move must precede another move if the former precede the latter in the sequence of moves produced by the solving algorithm and these moves are dependent. The following definition formalizes the concept of dependence between moves.

Definition 4 ***(dependent/independent moves).*** A move $q: v_1 \rightarrow v_2$ (a robot q is moved from a vertex v_1 to a vertex v_2) generated at time step t_q is *independent* on a move $r: u_1 \rightarrow u_2$ generated at time t_r where $t_r < t_q$ if $r \neq q$ and either $\{u_1, u_2\} \cap \{v_1, v_2\} = \emptyset$ or $u_2 \neq v_2$. Otherwise the moves are *dependent*. □

This (anti-symmetric) relation of dependence induces a directed acyclic graph on the set of moves generated by the solving algorithm. Then the method of critical path can be applied to calculate the earliest time step for each move when it can be executed. The consequence of this process is that the solution is made parallel and the plan duration is shortened.

Experimental Evaluation

An initial experimental evaluation of both ideas for shortening the solutions of multi-robot path planning has been made. The experimental evaluation was concentrated on the measuring of the improvements that can be gained using the described shortening techniques.

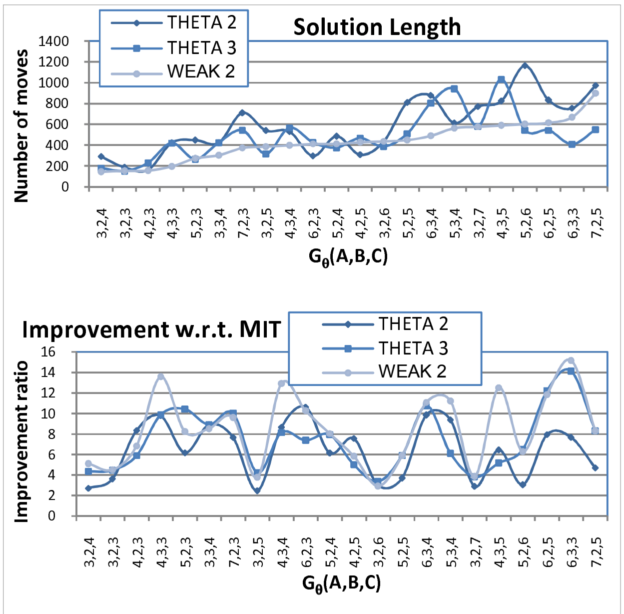

Figure 3. *Solution lengths and solution length improvements on graphs with single unoccupied vertex.* Solution lengths on small θ-like graphs is compared. Several methods are compared - exchanges of robots (THETA 2), rotation along 3-cycle (THETA 3), and application of weak transposition cases (WEAK 2). Improvement with respect to existing method from (Kornhauser *et al.*, 1984) is shown.

The first series of experiments is dedicated to the evaluation of the application of weak transposition cases. The application of weak transposition cases was compared against application of standard transpositions of robots as well as against application of 3-cycle rotation. The tests made on θ-like graphs are shown in figure 3. The comparison of improvements with respect to another existing algorithm from (Kornhauser *et al.*, 1984) is also shown (the algorithm is called *MIT* here).

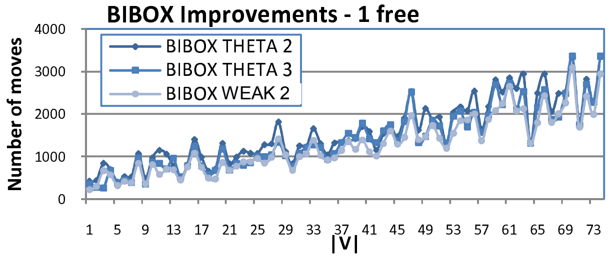

Figure 4. *Comparison of the improvements of the BIBOX algorithm for graphs with single unoccupied vertex.* Solution lengths comparison on bi-connected graphs. Several variants of the *BIBOX* algorithm are used – they differ in approach to solving the original θ-like graph. Three methods are compared - exchanges of robots (THETA 2), rotation along 3-cycle (THETA 3), and application of weak transposition cases (WEAK 2).

The observation from the results is that application of weak transposition cases brings significant improvement on the major part of the inputs. The weak transposition case requires up to 2-times less moves than normal transposition cases and 3-cycle rotation cases.

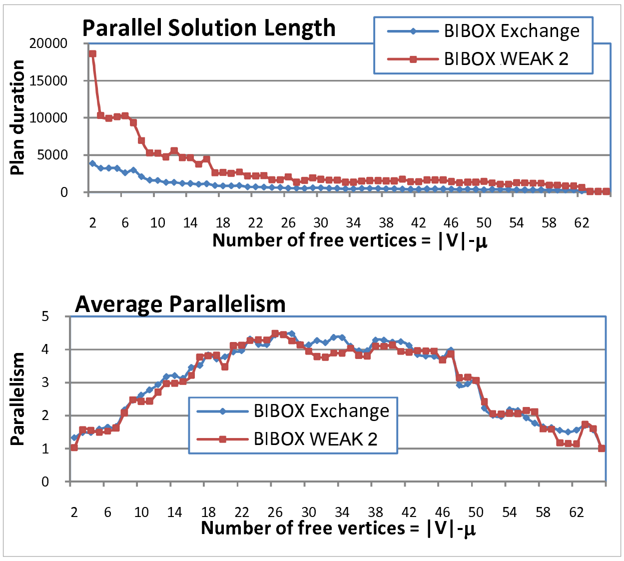

Figure 5. *Solution lengths and parallelism in graphs with more free vertices.* The original *BIBOX* algorithm (Exchange) and the *BIBOX* algorithm with weak transposition cases (WEAK 2) integrated are integrated. The original *BIBOX* algorithm is better on the graph with more unoccupied vertices.

Figure 4 shows comparison of the weak transposition case integrated into the *BIBOX* algorithm on bi-connected graphs. Random bi-connected graphs were used in this setup. A random bi-connected graph is constructed by starting with the cycle of random length (uniform distribution between 5 and 11) and by adding loops of random length (uniform distribution between 3 and 9). The number of iterations of adding loops was 2 to 16. Graphs of up to 76 vertices were used in test.

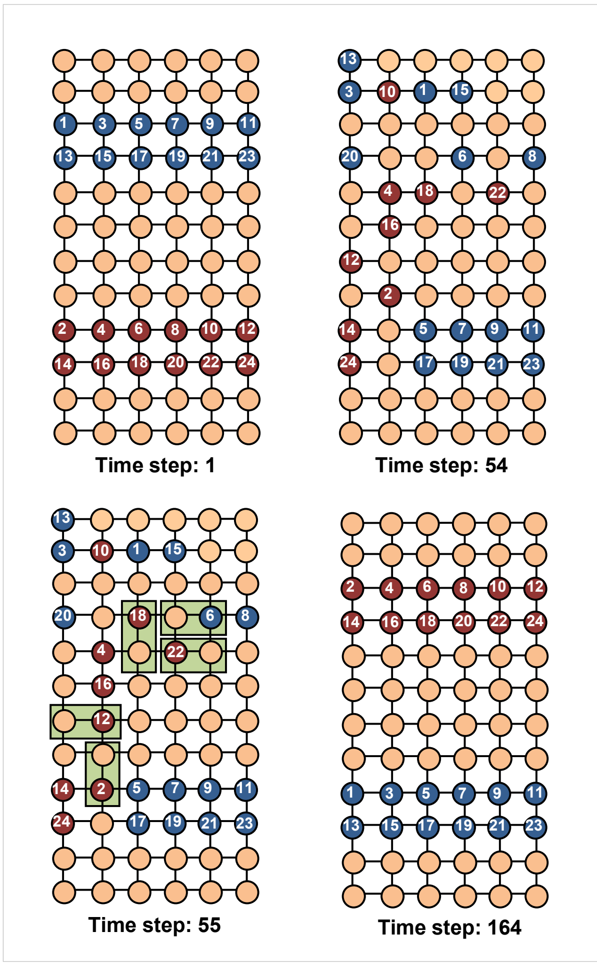

Figure 6. *Parallel solution of multi-robot path planning in a grid.* The grid represents a special instance of the bi-connected graph. The solution is produced by the *BIBOX* algorithm and the solution is parallelized using the critical path method. Parallel moves are depicted.

The second series of experiments was devoted to the evaluation of parallelism gained using critical path method. The results are shown in figure 5. Critical path parallelism was integrated into the original *BIBOX* algorithms and into its variant using weak transposition cases. A random bi-connected graph with 70 vertices was used for this experiment. The number of unoccupied vertices ranged from 2 to 69 (the graph was still the same).

The first observation is that the original *BIBOX* algorithm produces better solutions in bi-connected graphs with more than one unoccupied vertices. Hence, the use of optimal macros is not as beneficial as may be expected. The conclusion is that optimal macros worth using only in the case of single unoccupied vertex. The second observation is that critical path method is able to reach relatively high parallelism and it is beneficial under all circumstances.

The experiments with the structured problem are shown in figures 6 and 7. The structured problem is represented by a task of exchanging group of robots in a grid (which is a bi-connected graph). The snapshots of several stages of solving process are shown in figure 6. The development of parallelism in the individual time steps of the same problem in shown in figure 7.

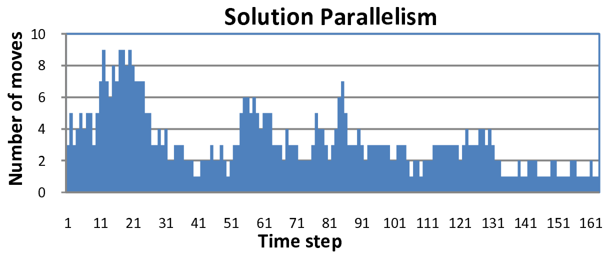

Figure 7. *Parallelism in the individual time steps of the solution of multi-robot path planning in a grid.* The development of parallelism in time is shown. The problem from the figure 6 is shown.

All the experimental results are available on-line at the web: http://ktiml.mff.cuni.cz/~surynek/research/socs2009.

The interpretation of results is that critical path method is able to reach relatively high parallelism however intuitive solving of the problem by hand may bring higher parallelism since a human solver would understand the structure of the problem. Nevertheless, the structure may not be visible automatically. In such a situation the human may not be able to solve the problem at all while the automated process (the *BIBOX* algorithm) solves the problem immediately

The use of optimal macros solving weak transposition case is beneficial only in the situation when there is a single unoccupied vertex in the graph modeling the problem.

Related Works and Conclusions

This work is significantly influenced by (Ryan, 2007). The author presents a solving method for the multi-robot path planning based on a decomposition of the environment into simpler sub-graphs that are easier to tackle. This approach has much in common with the approach presented above. However, deep theoretical results gained for *pebble motion on graphs* (*sliding box puzzles*) (Wilson, 1974), (Ratner & Warmuth, 1986), and (Kornhauser *et al.*, 1984) are not referred nor utilized in (Ryan, 2007), though they are closely related to multi-robot path planning.

One of the aims of this paper is to fill in the gap between theory and practical solving of problems of multi-robot path planning. It is necessary to emphasize that this paper intensively builds on existing works while aspect regarding the optimality of solutions are improved.

Graphical properties crucial for tackling the problem were identified in (Wilson, 1974). The solving methods for transposition and 3 cycle rotation cases ware developed in (Surynek, 2009b). The original version of the BIBOX algorithm is presented in (Surynek, 2009a). A comparison with domain-independent planners and scaling evaluation of the *BIBOX* algorithm is also given in (Surynek, 2009a). *LPG-td* and *SGPlan* were tested against the *BIBOX* algorithm; only extremely small multi-robot problems are solvable by the tested domain-independent planners. These results render the domain-independent approach to be uncompetitive.

In this work, two techniques for shortening solutions of multi-robot path planning problems are presented. The first technique is based on composing the solution of optimal pre-calculated macros for problems over so called θ-like graphs where two robots are exchanged, some robots stay in their original positions, and some robots are allowed to move arbitrarily (weak transposition case). The second technique is focused on increasing parallelism within the resulting solution. This technique exploits the definition of dependence between moves. If two consecutive moves are independent then they can be performed in parallel. The analysis of what moves can be performed in parallel is done by the method of critical path.

Both solution shortening methods proved to bring significant improvements on certain classes of multi-robot path planning problems. A deeper analysis of the possible parallelism within the solution seems to be a good research direction in the future works.

Acknowledgement

This work is supported by the Czech Science Foundation (Grantová agentura České republiky - GAČR) under the contracts number 201/09/P318, 201/07/0205, and by the Ministry of Education, Youth and Sports, Czech Republic (Ministerstvo školství, mládeže a tělovýchovy - MŠMT) under the contract number MSM 0021620838.

References

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.

Russell, S., Norvig P., 2003. *Artificial Intelligence: A Modern Approach (second edition).* Prentice Hall.

Ryan, M. R. K., 2007. *Graph Decomposition for Efficient Multi-Robot Path Planning.* Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India, pp. 2003-2008, IJCAI Conference, 2007.

Surynek, P., 2009a. *A Novel Approach to Path Planning for Multiple Robots in Bi-connected Graphs*. Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan, pp. 3613-3619, IEEE Press.

Surynek, P., 2009b. *Towards Shorter Solutions for Problems of Path Planning for Multiple Robots in θ-like Environments.* Proceedings of the 22nd International FLAIRS Conference (FLAIRS 2009), FL, USA, pp. 207-212, AAAI Press.

West, D. B., 2000. *Introduction to Graph Theory, second edition.* Prentice-Hall.

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