Finding Solvable Priority Schemes for Decoupled Path Planning Techniques for Teams of Mobile Robots

Maren Bennewitz **Wolfram Burgard**

Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany

Abstract. Coordinating the motion of multiple mobile robots is one of the fundamental problems in robotics. The predominant algorithms for coordinating teams of robots are decoupled and prioritized, thereby avoiding combinatorially hard planning problems typically faced by centralized approaches. In this paper we present a method for finding solvable priority schemes for such prioritized and decoupled planning techniques. Existing approaches apply a single priority scheme which makes them overly prone to failure in cases where valid solutions exist. By searching in the space of priorization schemes, our approach overcomes this limitation. To focus the search, our algorithm is guided by constraints generated from the task specification. To illustrate the appropriateness of this approach, this paper discusses experimental results obtained with real robots and through systematic robot simulation. The experimental results demonstrate that our approach can successfully solve many more coordination problems than previous decoupled and prioritized techniques.

1 Introduction

Path planning is one of the fundamental problems in mobile robotics. As mentioned by Latombe [11], the capability of effectively planning its motions is "eminently necessary since, by definition, a robot accomplishes tasks by moving in the real world."

In this paper we consider the problem of motion planning for multiple mobile robots. The goal is to compute trajectories for the individual robots such that collisions between the robots are avoided. Especially in the context of multi-robot systems different undesirable situations can occur like congestions or even deadlocks. Since the size of the joint state space of the robots grows exponentially in the number of robots, planning paths for teams of mobile robots is significantly harder than the path planning problem for single robot systems. Therefore, the existing approaches for single robot systems cannot directly be transferred to multi-robot systems.

The approaches for multi-robot path planning can roughly be divided into two major categories [11]: centralized and decoupled. In the *centralized* approach [3, 22] the configuration spaces of the individual robots are combined into one composite configuration space which is then searched for a path for the whole composite system. Because the size of the joint configuration space grows exponentially in the number of robots, this approach suffers intrinsic scaling limitations. The major alternative are *decoupled approaches* [8, 18, 15, 7, 21, 14, 6, 24, 1, 9]. Decoupled approaches first compute separate paths for the individual robots and then resolve possible conflicts of the generated paths.

While centralized approaches (at least theoretically) are able to find the optimal solution to any planning problem for which a solution exists, their time complexity is at least exponential in the dimension of the composite configuration space [19]. In practice one is therefore forced to use heuristics for the exploration of the huge joint state space. Many methods use potential field techniques [2, 3, 23] to guide the search. These techniques apply different approaches to deal with the problem of local minima in the potential function. Other methods restrict the motions of the robots to reduce the size of the search space. For example, [10, 22, 12] restrict the trajectories of the robots to lie on independent roadmaps. The coordination is achieved by searching the Cartesian product of the separate roadmaps.

Decoupled planners, in contrast, determine the paths of the individual robots independently and then employ different strategies to resolve possible conflicts. According to that, decoupled techniques are incomplete, i.e., they may fail to find a solution even if there is one. A popular decoupled approach is planning in the configuration time-space [8], which can be constructed for each robot given the positions and orientations of all other robots at every point in time. Techniques of this type assign priorities to the individual robots and compute the paths of the robots based on the order implied by these priorities. The method presented in [24] uses a fixed order and applies potential field techniques in the configuration time-space to avoid collisions. The approach described in [9] also uses a fixed priority scheme and chooses random detours for the robots with lower priority.

Another approach to decoupled planning is the path coordination method which was first introduced in [18]. The key idea of this technique is to keep the robots on their individual paths and let the robots stop, move forward, or even move backward on their trajectories in order to avoid collisions (see also [5]). To reduce the complexity in the case of huge teams of robots [14] recently presented a technique to separate the overall coordination problem into sub-problems. This approach, however, assumes that the overall problem can be divided into very small sub-problems, a serious assumption which, as various examples described below demonstrate, is often not justified. In general, therefore, a prioritized variant has to be applied.

Unfortunately the problem of finding the optimal schedule is NP-hard for most of the decoupled approaches. To see, we notice that the NP-hard Job-Shop Scheduling problem with the goal to minimize maximum completion time with unit processing time for each job [13] can be regarded as a special instance of the path coordination method. The decoupled and prioritized methods described above leave open how to assign the priorities to the individual robots. In the past, different techniques for selecting priorities have been used. For example, in [6] heuristic techniques are described that assign higher priority to robots which can move on a straight line from the starting point to their target location. In [1] all possible priority assignments are considered. Due to its (exponential) complexity this approach has only been applied to groups of up to three robots.

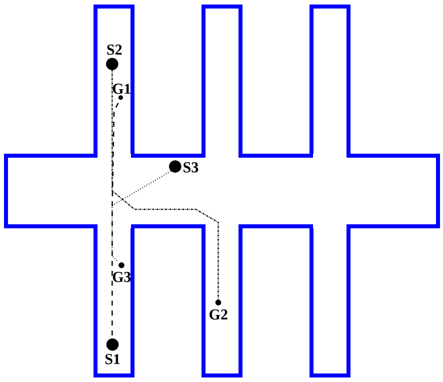

Fig. 1. Situation in which no solution can be found if robot 3 has higher priority than robot 1.

However, for decoupled and prioritized methods the order in which the paths are planned has a serious influence on whether at all a solution can be found. Figure 1 shows a situation in which no solution can be found if robot 3 has a higher priority than robot 1. Since the path of robot 3 is planned without considering robot 1, it enters the corridor containing its target location marked G_3 before robot 1 has left this corridor. Because the corridors are too narrow to allow two robots to pass by, robot 3 blocks the way of robot 1 so that it cannot reach its target point G_1. However, if we change the priorities and plan the trajectory of robot 1 before that of robot 3, then robot 3 considers the trajectory of robot 1 during path planning and thus will wait in the hallway until robot 1 has left the corridor.

Please note, that in order to find a solution one generally has to consider different priority schemes. Since each change of a scheme requires the computation of the paths for many of the robots, it is of utmost importance to minimize the time required to find priority schemes for which a solution to the path planning problem can be computed.

In this paper we present a technique which interleaves the search for an appropriate priority scheme with the planning of the paths for the individual robots. Our approach is a randomized search technique which starts with an initial priority scheme and changes this by swapping two randomly chosen robots. Thereby it exploits constraints between the optimal paths of the individual robots in order to focus the search. This way the number of problems for which a solution can be found in a given amount of time is increased significantly. Our technique has been implemented and tested on real robots and in extensive simulation runs. In all experiments it has been shown to be very effective even for large teams of robots, for different environments, and using two different decoupled path planning techniques.

The paper is organized as follows. The following section describes the prioritized and decoupled path planning techniques we apply our algorithm presented in Section 3 to. Section 4 contains experimental results illustrating the capabilities of our approach.

2 Prioritized A^*-based Path Planning and Path Coordination

The basic algorithm to compute optimal paths for single robots applied throughout this paper is the well-known A^* search procedure. The next section briefly describes the variant we are using. To represent the environment of the robots we apply occupancy grids [17] which separate the environment into a grid of equally spaced cells and store in each cell $\langle x, y\rangle$ the probability $P(occ_{x,y})$ that it is occupied by a static object. In the remainder of this section we then present the key ideas of decoupled prioritized path planning and discuss how the A^* procedure can be utilized to plan the motions of teams of robots by this approach.

2.1 A^*-based Path Planning

The A^* procedure simultaneously takes into account the accumulated cost of reaching a certain location $\langle x, y\rangle$ from the starting position as well as the estimated cost of reaching the target location $\langle x^*, y^*\rangle$ from $\langle x, y\rangle$. In our case, the cost for traversing a cell $\langle x, y\rangle$ is proportional to its occupancy probability $P(occ_{x,y})$. Furthermore, the estimated cost for reaching the target location is approximated by $c \cdot ||\langle x, y\rangle - \langle x^*, y^*\rangle||$ where c is chosen as the minimum occupancy probability $P(occ_{x,y})$ in the map and $||\langle x, y\rangle - \langle x^*, y^*\rangle||$ is the straight-line distance between $\langle x, y\rangle$ and $\langle x^*, y^*\rangle$. Since this heuristic is admissible, A^* determines the cost-optimal path from the starting position to the target location.

2.2 Decoupled Path Planning for Teams of Robots

In this paper we apply our search technique to two different decoupled path planning methods which plan the paths in the configuration time-space. Such approaches proceed as follows. First, one computes for each robot its path without considering the paths of the other robots. Then one checks for possible conflicts in the trajectories of the robots (we regard it as a conflict between two robots if their distance is less than δ where $\delta = 1.2m$ in our current system). Conflicts between robots are resolved by introducing a priority scheme. A priority scheme determines the order in which the paths for the robots are re-planned. The path of a robot is then planned in its configuration time-space computed based on the map of the environment and the paths of the robots with higher priority.

Our system applies the A^* procedure to compute the cost-optimal paths for the individual robots, in the remainder denoted as the independently planned optimal paths for the individual robots. We also apply A^* search to plan the motions of the robots in the configuration time-space. In this case the cost of traversing a location $\langle x, y\rangle$ at time t is determined by the occupancy probability $P(occ_{x,y})$ plus the probability that one of the other robots with higher priority covers $\langle x, y\rangle$ at that time.

In this paper we consider two different strategies. The first method is the general A^*-based planning in the configuration time-space. The second method is a restricted version of this approach denoted as the path coordination technique [14]. It differs from the general approach in that it only explores a subset of the configuration time-space given by those states which lie on the initially optimal paths for the individual robots. The path coordination technique thus forces the robots to stay on their initial trajectories. The overall complexity of both approaches is $O(n \cdot m \cdot \log(m))$ where n is the number of robots and m is the maximum number of states expanded by A^* during planning in the configuration time-space (i.e. the maximum length of the OPEN-list).

Due to the restriction during the search, the path coordination method is more efficient than the general A^* search. Its major disadvantage, however, lies in the fact that it fails more often.

3 Searching for Solvable Priority Schemes

As already mentioned above, prioritized and decoupled approaches to multi-robot path planning are incomplete. However, as the example given in Figure 1 illustrates, the order in which the paths are planned has a significant influence on whether a solution can be found. This raises the question of how to find a solvable priority scheme, i.e. an order for which collision-free paths can be computed using a decoupled approach.

3.1 The Randomized Search Technique

Recently, randomized search techniques have been used with great success to solve constraint satisfaction problems or to solve satisfiability problems [20]. Our algorithm presented here is a variant which performs a randomized search in order to find a solvable planning order for decoupled and prioritized path planning techniques. Thereby it interleaves the search for collision-free paths with the search for a solvable priority scheme. It starts with an arbitrary initial priority scheme Π and randomly exchanges the priorities of two robots in this scheme. If we get a scheme Π for which collision-free paths can be found, we return this order. In order to escape from possible dead-ends in the search space, we perform random restarts with different initial orders of the robots. The complete algorithm is listed in Table 1.

Table 1. The algorithm to find solvable priority schemes.

```
FOR tries := 1 TO maxTries BEGIN
    select random order Π
    FOR flips := 1 TO maxFlips BEGIN
        choose random i, j with i<j
        Π := swap(i, j, Π)
        if solvable(Π)
            RETURN Π
    END FOR
END FOR
RETURN "No solution found"
```

3.2 Exploiting Constraints to Focus the Search

Whereas the plain randomized search technique produces good results, it has the major disadvantage that often a lot of iterations are necessary to come up with a solution. For example, we found that for ten robots more than 20 iterations on average were necessary to find a solvable priority scheme. In this section we therefore present a technique to focus the search. As an example again consider the situation depicted in Figure 1. As already mentioned, it is impossible to find a path for robot 1 if the path of robot 3 is planned first, because the target location of robot 3 is too close to the optimal trajectory for robot 1. The key idea of our approach is to introduce a constraint $p_i > p_j$ between the priorities of two robots i and j, whenever the target position of robot j is too close to the initially optimal path of robot i. In our example we thus obtain the constraint $p_1 > p_3$ between the robots 1 and 3. Additionally, we get the constraint $p_2 > p_1$, since the target location of robot 1 lies too close to the trajectory of robot 2.

Although the satisfaction of the constraints does not guarantee that collision-free paths can be found for a priority scheme, orders satisfying the constraints more often have a solution than priority schemes violating constraints. Unfortunately, depending on the environment and the number of the robots it is possible that there is no order satisfying all constraints. In such a case the constraints produce a cyclic dependency. The key idea of our approach is to reorder only those robots which are involved in such a cycle in the constraint graph. Thus, we separate all robots into two sets. The first group R_1 contains all robots that, according to the constraints, do not lie on a cycle and have a higher priority than the robot with highest priority which lies on a cycle. This set of robots is ordered according to the constraints and this order is not changed during the search. The second set, denoted as R_2 contains all other robots. During the search only the order of the robots in the second group is changed.

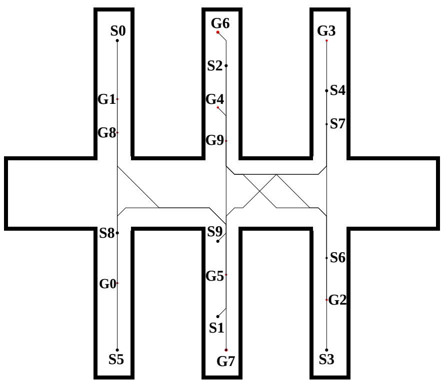

Fig. 2. Independently planned paths for ten robots.

Figure 2 shows a simulated situation with ten robots. Whereas the starting positions are marked by $S_0, \ldots, S_9$ the corresponding goal positions are marked by $G_0, \ldots, G_9$. The independently planned optimal trajectories are indicated

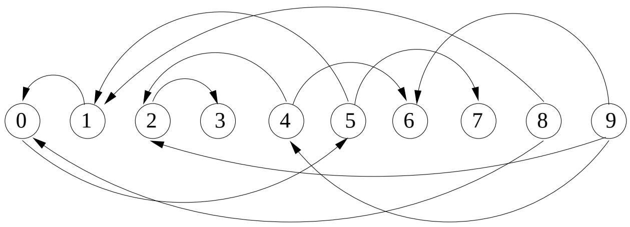

Fig. 3. Constraints generated according to the paths shown in Figure 2.

by solid lines. Given these paths we obtain the constraints depicted in Figure 3. According to the constraints, six robots belong to the first group of robots whose order remains unchanged during the search process. The robots in their order of priorities are 3, 6, 7, 2, 4, 9. Only the other four robots are considered during the search for a solvable priority scheme. Our approach starts with the order 0, 1, 5, and 8 for the remaining robots for which our system immediately can determine the collision-free paths shown in Figure 4. Thus, the constraints immediately lead to a solvable priority scheme.

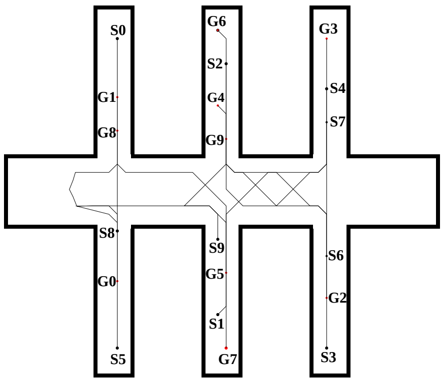

Fig. 4. Paths resulting after priority optimization.

4 Experimental Results

The algorithm described above has been tested thoroughly on real robots and in extensive simulation runs. The key question addressed in our experiments was: Does our approach succeed more frequently in finding valid multi-robot paths than approaches with fixed prioritization? All experiments were carried out using different environments. To evaluate the general applicability, we applied our method to the two decoupled and prioritized path planning techniques described above. The current implementation is highly efficient. It requires less than 0.1 seconds on a 1000 MHz Pentium III to plan a collision-free path for one robot in all environments described below.

4.1 Real Robot Experiment

Figure 6 illustrates a typical application example carried out in our office environment with our robots Albert and Ludwig. The robots are shown in Figure 5. In this example, we used the general A^* procedure in the configuration time-space for local path planning. While Albert starts at the right end of the corridor of our lab and has to move to left end, Ludwig has to traverse the corridor in the opposite direction. Notice that no path for Albert can be found if the path of Ludwig is planned first, since Albert cannot reach its target point if Ludwig stays on its optimal trajectory. Because of that, the system alters the order of the two robots. Given the optimal path for Albert, our system plans a path for Ludwig which first leads it into a doorway in order to let Albert pass by. The resulting trajectories are shown in

Fig. 5. The mobile robots Albert (left) and Ludwig (right).

Figure 6. Notice that at some point, the robot Ludwig waits to let the robot Albert pass by. In comparison, no solution can be found in this situation if the path coordination [14] technique is used.

In various other tests operating our two robots in our narrow hallways, we frequently observed the emergence of solutions where robots sensibly coordinated their behavior, e.g., by waiting for each other. However, we also notice that with only two robots, these experiments do not evaluate the utility of our search algorithm in priority scheme space, since there exist only two such schemes. Unfortunately, we currently have only two physical robots available in our lab, so that the experiment could not be carried out with larger groups of robots.

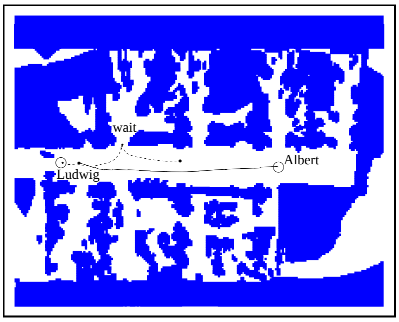

Fig. 6. Real world application of general A^*-based planning in the configuration time-space.

4.2 Simulation Experiments

This set of experiments is designed to illustrate that the overall number of failures can be reduced significantly using our randomized search technique and even more significantly by taking into account the generated constraints during the search.

In all experiments we never observed that the constrained search failed more often than the unconstrained search. Furthermore, there was no significant difference between the lengths of the generated paths.

For each number of robots considered, we performed 100 experiments. In each experiment we randomly chose the starting and target locations of the robots. We applied four different strategies to find solvable priority schemes:

1. A single randomly chosen order for the robots without considering the constraints.
2. A single order which satisfies the constraints for the robots in R_1 and consists of a randomly chosen order for the robots in R_2.
3. Unconstrained randomized search starting with a random order and without considering the constraints.
4. Constrained randomized search starting with an order computed in the same way as strategy 2).

For each technique, we performed A^*-based planning in the configuration time-space and counted the number of solved planning problems. Please note that in this experiment we chose a small number of iterations in order to

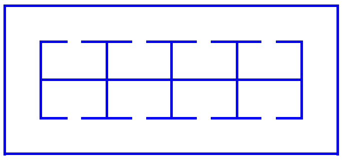

Fig. 7. Cyclic corridor environment used for the simulation runs.

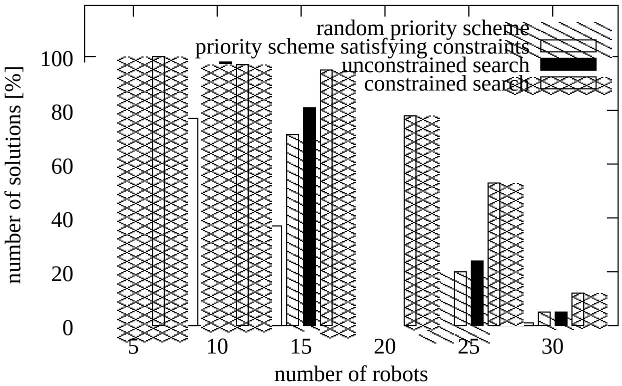

Fig. 8. Solved planning problems for different strategies using general A^*-based planning in the configuration time-space in the cyclic corridor environment depicted in Figure 7.

assess the advantages of the constrained search. Particularly, we chose a value of 3 was for the parameters `maxFlips` and `maxTries`, in both randomized search methods. Obviously, the larger the number of iterations, the higher is the probability that a solution can be found by an arbitrary randomized search.

Figure 8 summarizes the results we obtained for the cyclic corridor environment depicted in Figure 7. Whereas the x-axis represents the number of robots, the y-axis contains the number of solved problems in percent. As this figure shows, our constrained search technique is significantly more often able to find a solution compared to all other strategies. Interestingly, the second strategy, which exploits the constraints but considers only one scheme in each experiment, shows a similar performance than the unconstrained randomized search.

Additionally, we performed a similar series of experiments for the noncyclic corridor environment depicted in Figure 4. The results are shown in Figure 9 (left). Again, our constrained-based search outperforms all other strategies.

Furthermore, we analyzed all four strategies to find solutions for the path coordination method. Throughout these experiments we used a variant of the environment depicted in Figure 4 with five corridors on both sides. Since the path coordination method restricts the robots to stay on their independently planned optimal trajectories, the number of unsolvable problems is much higher compared to the general A^*-based planning in the configuration time-space. As can be seen from Figure 9 (right) our constrained based search again leads to a much higher success rate.

4.3 Speed-up Obtained by Exploiting the Constraints

The previous experiments illustrated that the number of cases in which a solution can be found is increased significantly by focusing the search according to the constraints. In this section we want to analyze the speed-up obtained by restricting the search. More precisely, we pose the question how much time the unconstrained search would require in order to achieve the same performance as our constraint-based search technique.

Figure 10 shows for both corridor environments considered here the number of robots which could not be sorted topologically with respect to the generated constraints because they lie on a cycle in the constraint graph. Obviously, this number increases in the number of robots in both environments. Since our constrained search only reorders the robots lying on the cycle, the search is significantly focused in both environments. Obviously, the noncyclic corridor environment depicted in Figure 4 poses harder planning problems than the cyclic corridor environment shown in Figure 7. Accordingly, the search requires more iterations in the first case. Averaged over 1200 experiments, the unconstrained search required over five as many iterations in the first case and four as many iterations in the second case to achieve the same performance as our constrained search.

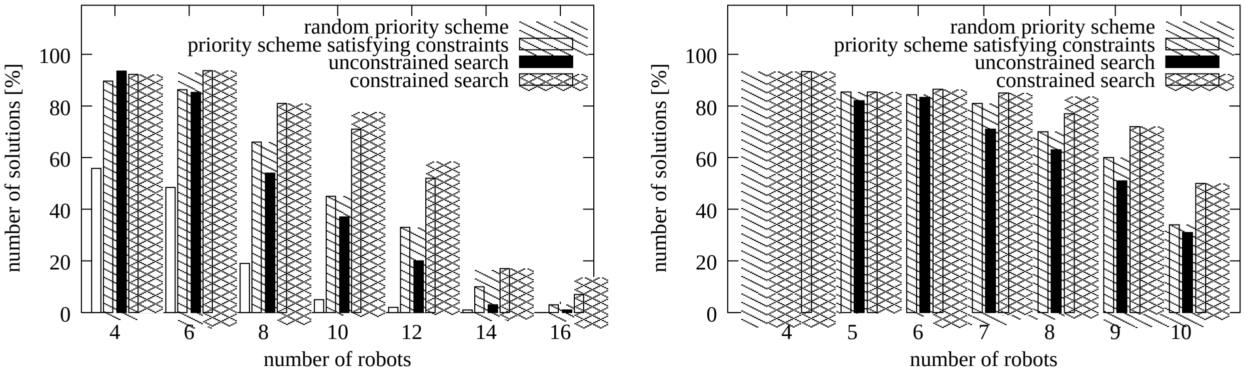

Fig. 9. Number of solved planning problems in the noncyclic corridor environment shown in Figure 4 for the different strategies using general A^*-based planning in the configuration time-space (left) and using the path coordination method (right).

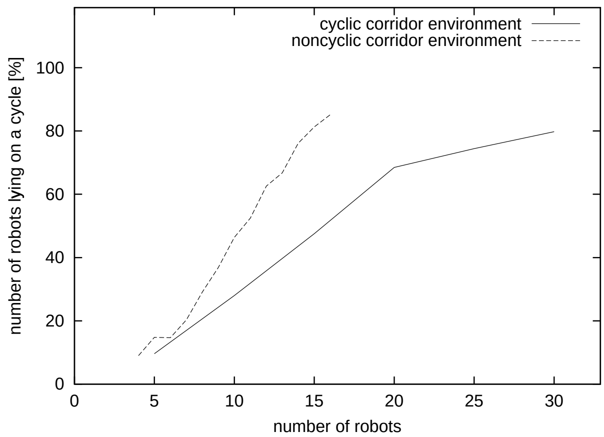

Fig. 10. Number of robots lying on a cycle in the constraint graph.

Please note that a further advantage of the constrained search compared to the unconstrained search is that for the robots in R_1 motion planning needs to be performed only once since their priorities are never changed.

4.4 Influence on the Overall Path Length

A further important question in the context of path planning is the minimization of the overall move costs. We also applied the randomized search technique described in this paper to minimize the length of the trajectories for a team of robots [4]. In this case, the randomized search was combined with a hill-climbing strategy. If additional time is available, the system performs several restarts and always keeps the best solution found so far. Figure 11 (left) shows the independently planned optimal paths for a team of 30 robots in an unstructured environment. By optimizing these paths over 100 iterations, we obtain the solution illustrated in Figure 11 (right). Figure 12 plots the evolution of the summed move costs of the best solution found so far over time and demonstrates the capabilities of this approach to reduce the overall path length. As can be seen from the figure, after 100 iterations the overall move costs are reduced by 15%.

5 Conclusions

In this paper we presented an approach to find solvable priority schemes for decoupled path planning methods for groups of mobile robots. Our approach is a randomized method which repeatedly reorders the robots to find a sequence for which a plan can be computed. To reduce the computation time necessary to find a solution certain constraints between the robots are extracted and exploited to focus the search. The approach has been implemented and tested

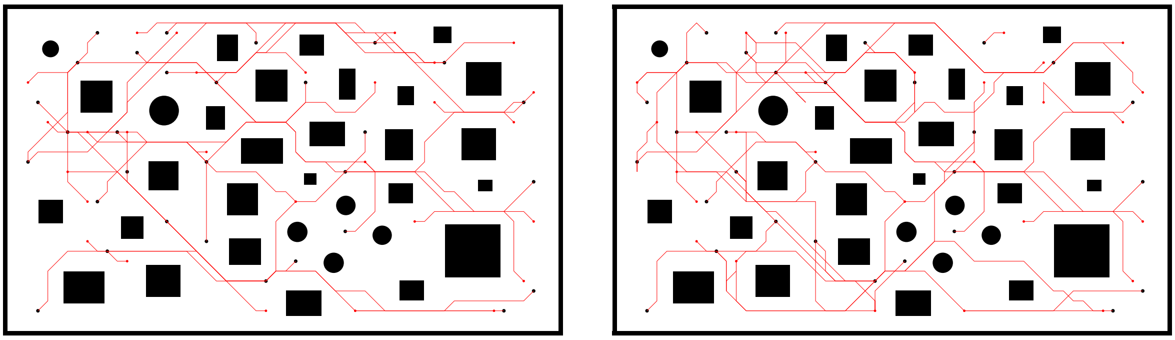

Fig. 11. Independently planned optimal paths for 30 robots (left) and the paths resulting after priority optimization (right).

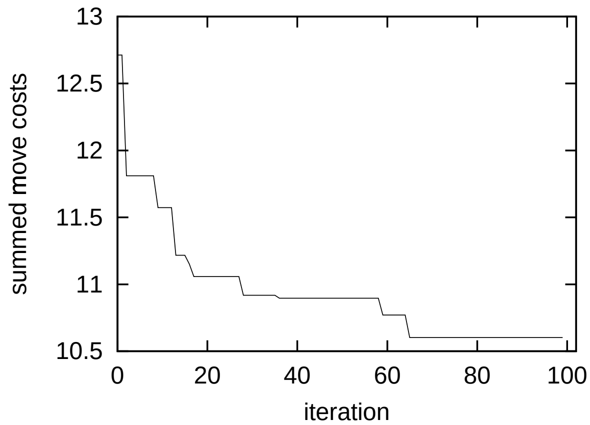

Fig. 12. Summed move costs plotted over time.

on real robots as well as in extensive simulation runs for two different decoupled path planning techniques. The experiments demonstrate that our technique significantly decreases the number of failures in which no solution is found for a given planning problem. Furthermore, our randomized search method can also be used to minimize the overall path length.

It should be noted that our algorithm is not limited to the two different baseline path-planning techniques considered in this paper. In contrast, it can be used to find and optimize paths generated with arbitrary prioritized path-planning methods. This also includes planning techniques without the assumptions of the methods considered in this paper, like accurate global models of the environment and like deterministic execution of the planned movements. Additionally, we would like to mention that our method is equally suited to more complex coordination problems, in which the robots have large degrees of freedom.

Apart from the promising results presented in this paper, there are different aspects for future research. First, the technique described here provides no means to react to possible failures during the execution of the motion plans. For example, if one robot is delayed because unforeseen objects block its path, alternative plans for the robots might be more efficient. In such situations it is desirable that the system can quickly revise the current plan. Second, the delay of a single robot may result in a dead-lock during the plan execution. Accordingly, our systems requires techniques to detect such dead-locks and to resolve them appropriately. Finally, the current approach reduces the search to a fixed subset of the robots. Since this restriction reduces the number of possible solutions, we will investigate whether the performance can be increased further by extending the search appropriately.

References

1. K. Azarm and G. Schmidt. A decentralized approach for the conflict-free motion of multiple mobile robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1667–1674, 1996.
2. J. Barraquand, B. Langois, and J. C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Robotics and Automation, Man and Cybernetics*, 22(2):224–241, 1992.
3. J. Barraquand and J. C. Latombe. A monte-carlo algorithm for path planning with many degrees of freedom. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1990.
4. M. Bennewitz, W. Burgard, and S. Thrun. Optimizing schedules for prioritized path planning of multi-robot systems. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001. to appear.
5. Z. Bien and J. Lee. A minimum-time trajectory planning method for two robots. *IEEE Transactions on Robotics and Automation*, 8(3):414–418, 1992.
6. S. J. Buckley. Fast motion planning for multiple moving robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1989.
7. H. Chu and H. A. EiMaraghy. Real-time multi-robot path planner based on a heuristic approach. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1992.
8. M. Erdmann and T. Lozano-Perez. On multiple moving objects. *Algorithmica*, 2:477–521, 1987.
9. C. Ferrari, E. Pagello, J. Ota, and T. Arai. Multirobot motion coordination in space and time. *Robotics and Autonomous Systems*, 25:219–229, 1998.
10. L. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars. Probabilistic road maps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, pages 566–580, 1996.
11. J.C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA, 1991. ISBN 0-7923-9206-X.
12. S. M. LaValle and S. A. Hutchinson. Optimal motion planning for multiple robots having independent goals. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1996.
13. E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. Sequencing and scheduling: Algorithms and complexity. Technical report, Centre for Mathematics and Computer Science, 1989.
14. S. Leroy, J. P. Laumond, and T. Simeon. Multiple path coordination for mobile robots: A geometric algorithm. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.
15. Y. H. Liu, S. Kuroda, T. Naniwa, H. Noborio, and S. Arimoto. A practical algorithm for planning collision-free coordinated motion of multiple mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 1427–1432, 1989.
16. P. Martin and D.B. Shmoys. A new approach to computing optimal schedules for the job-shop scheduling problem. In *Proc. of the 5th International IPCO Conference*, pages 389–403, 1996.
17. H.P. Moravec and A.E. Elfes. High resolution maps from wide angle sonar. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 116–121, 1985.
18. P. A. O'Donnell and T. Lozano-Perez. Deadlock-free and collision-free coordination of two robot manipulators. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1989.
19. J. T. Schwartz, M. Scharir, and J. Hopcroft. *Planning, Geometry and Complexity of Robot Motion*. Ablex Publishing Corporation, Norwood, NJ, 1987.
20. B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard instances of satisfiability. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1992.
21. K. Souccar and A. G. Roderic. Distributed motion control for multiple robotic manipulators. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1996.
22. P. Sveska and M. Overmars. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1995.
23. P. Tournassoud. A strategy for obstacle avoidence and its application to multi-robot systems. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 1224–1229, 1986.
24. C. Warren. Multiple robot path coordination using artificial potential fields. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 500–505, 1990.

This article was processed using
the TEX macro package with SIRS2001 style