Coordinating the Motions of Multiple Mobile Robots Using a Probabilistic Model

Maren Bennewitz **Wolfram Burgard**

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

Abstract. This paper considers the problem of path planning for teams of mobile robots. It presents a decoupled and prioritized approach to coordinate the movements of the mobile robots in their environment. Our algorithm computes the paths for the individual robots in the configuration time-space. Thereby it trades off the distance to both to static objects as well as to other robots and the length of the path to be traveled. To estimate the risk of colliding with other robots it uses a probabilistic model of the robots motions. The approach has been implemented and tested on real robots as well as in extensive simulation runs. In different experiments we demonstrate that our approach is well suited to control the motions of a team of robots in various environments and illustrate its advantages over other techniques developed so far.

1 Introduction

Path planning is one of the fundamental problems in mobile robotics. As mentioned by Latombe [8], the capability of effectively planning its motions is "eminently necessary since, by definition, a robot accomplishes tasks by moving in the real world." Especially in the context of autonomous mobile robots, path planning techniques have to simultaneously solve two complementary tasks. On one hand, their task is to minimize the length of the trajectory from the starting position to the target location, and on the other hand they should maximize the distance to obstacles in order to minimize the risk of colliding with an object.

In this paper we consider the problem of motion planning for multiple mobile robots. This problem is significantly harder than the path planning problem for single robot systems, since the size of the joint state space of the robots grows exponentially in the number of robots. Therefore, the solutions known for single robot systems cannot directly be transferred to multi-robot systems.

The existing methods for solving the problem of motion planning for multiple robots can be divided into two categories [8]. In the *centralized* approach 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. In contrast to that, the *decoupled* approach first computes separate paths for the individual robots and then tries to 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 exponential in the dimension of the composite configuration space. In practice one is therefore forced to use heuristics for the exploration of the huge joint state space.

Many methods use potential field techniques [1, 2, 18] 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 robot to reduce the size of the search space. For example, [17, 9] 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 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. [5] consider coarse two-dimensional slices to represent the configuration time-space. [20] applies potential field techniques in the configuration time-space to resolve conflicts. All these techniques assign priorities to the individual robots and compute the paths in decreasing order starting with the robot with highest priority. Whenever a path is re-planned, these approaches try to resolve the conflicts with the previously determined paths. In this context, an important question is how to assign the priorities to the individual robots. In the approach presented in [4] higher priority is assigned to robots which can move on a straight line from the starting point to its target location. The approach described in [6] does not apply a priority scheme. Instead, it uses sets of alternative paths for the individual robots and determines a solution by applying heuristics to pick appropriate paths from the different sets.

An alternative approach to decoupled planning is the path coordination method which was first introduced in [15]. This method computes the paths of the individual robots independently and then applies scheduling techniques to deal with possible conflicts. 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 [3]). Coordinating the motions of multiple robots along fixed paths can be regarded as a special instance of the Job-Shop Scheduling Problem with the goal to minimize maximum completion time with unit processing time for each job. This problem is known to be NP-hard [13, 10]. Recently [11] presented a technique to separate the overall coordination problem into sub-problems thus allowing to deal with huge teams of robots.

[12] presented a reactive approach for decentralized real-time motion planning. Each robot plans its path towards its target dynamically based on its current position and sensory feedback. Since this method is similar to potential field approaches, it suffers from local minima and may also result in oscillations. Finally there are different techniques based on heuristics like traffic rules to resolve arising conflicts [7, 19].

The method described here is a decoupled and prioritized approach to coordinated path-planning for multiple robots. It applies the well-known A^* procedure to compute cost-optimal paths in the configuration time-space and uses the optimal move costs of the individual robots to determine their priorities. A general assumption of most of the existing planning techniques is that the execution of the navigation plans is deterministic, i.e. the robots perform all actions with absolute certainty. However, robots operating in real and populated environments often are faced with unforeseen obstacles and have to deviate from their previously planned paths. During path planning our approach therefore explicitely considers possible deviations of the robots from their planned trajectories. The parameters of the deviation-model have been learned in several experiments. Our approach has been implemented and tested on real robots and in extensive simulation runs. The experiments carried out in various environments illustrate that our technique is well suited to coordinate teams of mobile robots. They furthermore demonstrate that our technique outperforms the coordination approach described in [11, 15].

2 Probabilistic Path Planning for Multiple Robots

The goal of path planning is to determine a trajectory with the optimal trade-off between the overall length and the distance to obstacles in the environment. To effectively plan the path of a mobile robot, path planning systems need a model of the environment. In our case, the map of the environment is given by an occupancy grid map [14]. The key idea of occupancy maps is to separate the environment into a grid of equally spaced cells. Each cell of such a grid contains the probability that this cell is occupied.

Given such a map our approach uses the well-known A^* procedure to determine the path from the current location to the target point. For each location $\langle x, y\rangle$ the A^* procedure simultaneously takes into account the cost of reaching $\langle x, y\rangle$ 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 approach the cost for traversing a cell $\langle x, y\rangle$ is proportional to its occupancy probability $P(occ_{x,y})$. The estimated cost for reaching the target location is approximated by the straight-line distance $|| \; \langle x, y\rangle - \langle x^*, y^*\rangle \; ||$ between $\langle x, y\rangle$ and $\langle x^*, y^*\rangle$. Accordingly, the minimum-cost path is computed using the following two steps.

1. **Initialization.** The grid cell that contains the robot location is initialized with 0, all others with ∞:

$$V_{x,y} \longleftarrow \begin{cases} 0, & \text{if } \langle x, y\rangle \text{ is the robot position} \\ \infty, & \text{otherwise} \end{cases}$$

2. **Update loop.** While the target location has not been reached do:

$$\langle x, y\rangle \longleftarrow \underset{\langle x', y'\rangle}{\operatorname{argmin}} \{V_{x',y'} \\ + c \cdot || \; \langle x', y'\rangle - \langle x^*, y^*\rangle \; || \; \}$$

   For each neighbor $\langle x', y'\rangle$ of $\langle x, y\rangle$ do

$$V_{x',y'} \longleftarrow \min \{V_{x',y'}, V_{x,y} \\ + || \; \langle x', y'\rangle - \langle x, y\rangle \; || \cdot P(occ_{x',y'})\}$$

In our approach, the constant c is chosen as the minimum occupancy probability $P(occ_{x,y})$, i.e.,

$$c = \min_{\langle x,y\rangle} P(occ_{x,y}).$$

This choice of c is necessary to ensure that A^* determines the cost-optimal path from the starting position to the target location. Figure 1 (left) shows a typical space explored by A^*. In this situation the robot starts in the corridor of our environment. Its target location is in the third room to the south. The figure also shows the accumulated costs of the cells considered by the

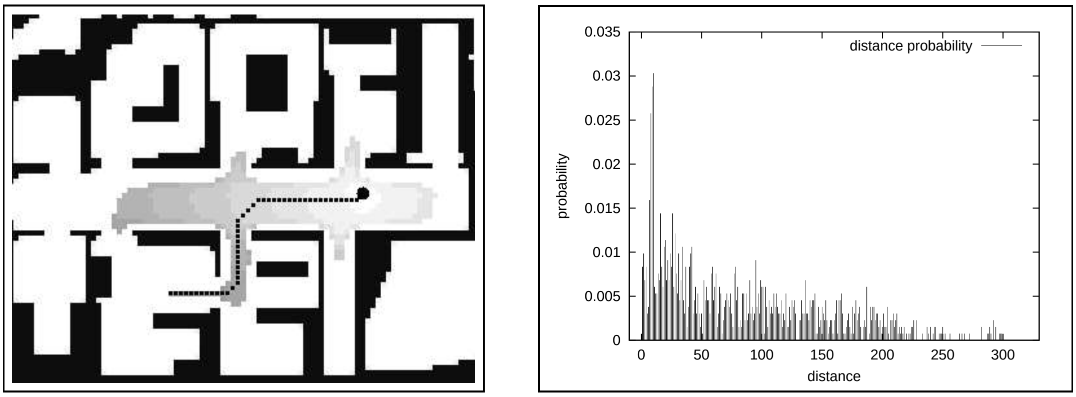


Fig. 1.: Result of a path planing process for a single robot using A^* (left) and the average deviation of a robot from the originally planned path during plan execution (right).

planning process (the darker the cell the higher the costs). As can be seen A^* only expands a small fraction of the overall state space and therefore is highly efficient. The disadvantage of the A^* procedure lies in the assumption that all actions are carried out with absolute certainty. To deal with the uncertainty in the robot's actions one in principle would have to use value iteration which generally is less efficient than A^*. To incorporate the uncertainty of the robots motions into the A^* approach, we convolve the grid map using a Gaussian kernel. This has a similar effect as generally observed when considering non-deterministic motions: It introduces a penalty for traversing narrow passages or staying close to obstacles. As a result, our robots generally prefer trajectories which stay away from obstacles.

As already mentioned above, our approach plans the trajectories of the robots in a decoupled fashion. First, we compute for each robot the cost-optimal path using the A^* procedure mentioned above. We then check for possible conflicts in the trajectories of the robots (in our current implementation we regard it as a conflict whenever two robots get closer than 1.2m). Whenever a conflict between some robots is detected, we use a priority scheme and determine new paths for the robots with lower priority. More precisely, suppose the k-th robot has a conflict with one or several of the $1, \ldots, k-1$ robots with higher priority. In this case we use A^* to re-plan the trajectory of this robot in its configuration time-space after including the constraints imposed by the $k-1$ robots with higher priority. According to that, the overall complexity of this planning technique is $O(nm \cdot \log(m))$ 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).

Please note, that although we only consider punctual and unconstrained motions in this paper, A^* is a general procedure which can deal with arbitrary state spaces. Thus, our technique can also be used for car-like robots by considering three-dimensional configuration spaces. It can furthermore be applied in the case of constrained spaces in which the constraints lead to a reduction of possible successor states. Therefore, our approach is a general method for planning the paths of teams of even non-circular and non-holonomic robots.

While planning in the configuration time-space we take into account possible deviations of the individual robots from their planned paths. For this purpose we use a probabilistic model which allows us to derive the probability that a robot will be at location $\langle x, y\rangle$ at time t given it is planned to be at location $\langle x', y'\rangle$ at that time. To estimate the parameters of this model we performed a series of 28 experiments with two robots in which we recorded the deviations of the robots from their pre-planned paths. In each run we constantly estimated for one robot the closest point on its planned trajectory and determined the distance of the second robot from the corresponding position of its path at the same point in time. As a result we obtained for a discrete set of distance ranges the number of times the second robot deviated from its originally planned path by that distance. The resulting probabilities are depicted in Figure 1 (right). In our current implementation this histogram is approximated by a set of linear functions in order to avoid over-fitting. Given these data, we can easily determine the probability $P_t^i(x, y)$ that robot i is at a location $\langle x, y\rangle$ at time t. This probability is then used to define a cost function which allows us to determine the cost for robot k of traversing cell $\langle x, y\rangle$ at time t:

$$C_t^k(x, y) = P(occ_{x,y}) + \sum_{i=1}^{k-1} P_t^i(x, y)$$

A typical application example of our planning technique is illustrated in Figure 2 (left). In this case, the robot depicted in light grey is supposed to move to the fourth room in the north. The second robot depicted in black starts in the corridor and

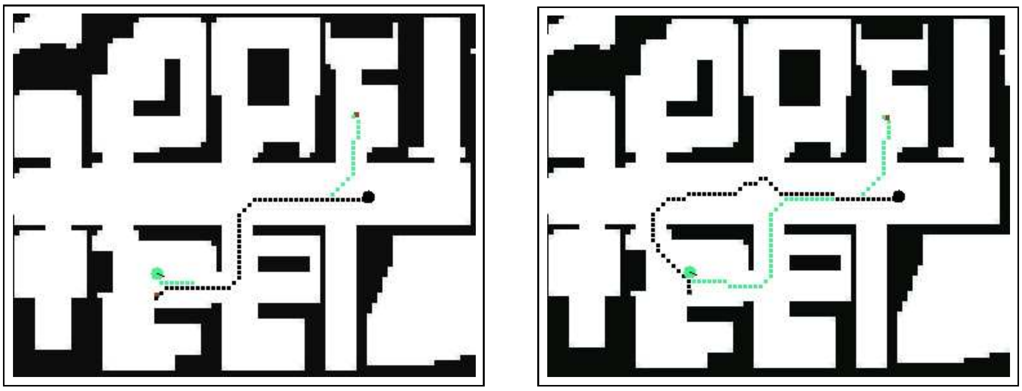


Fig. 2.: A conflict situation for two robots (left) and the resolved conflict by choosing a detour for the second robot (right).

has its target location close to the starting point of the first robot. Since both paths are planned independently, they impose a conflict between the two robots. After applying the A^* procedure in the configuration time-space for the second robot, the conflict is resolved. The planner decides that the black robot has to avoid the conflict with the grey robot by moving to the north just at the door where the first robot enters the corridor. After this collision avoidance action, the path through the next doorway appears to have less costs, so that it takes a completely different trajectory. The resulting trajectories are depicted in Figure 2 (right).

3 Experimental Results

The approach described above has been implemented and evaluated on real robots as well as in simulation runs. The current implementation is quite efficient, although there still is a potential for improvements. For the 19×15 m^2 large environment in which we carried out the experiments described here, our system is able to plan a collision-free path in the configuration time-space in less than 6 seconds and in less than 1.5 seconds for a cell size of 40×40 cm^2. Please note that this computation time will not increase in the number of robots, since our approach uses lookup-tables to store the costs introduced by the previously planned robots. The time needed for single robot path planning in the two-dimensional configuration space is generally less than 0.01 seconds. These performance measures were taken on a 500MHz Intel Pentium III running Linux.

3.1 Application Example with Real Robots

The system has been evaluated using our robots Albert and Ludwig which are depicted in Figure 3. Whereas Albert is an RWI B21 robot, Ludwig is a Pioneer I system. Both robots are equipped with a laser-range finder to reactively avoid obstacles. Figure 3 (middle) shows one situation, in which both robots have a conflict. While Ludwig starts at the left end of the corridor of our lab and has to move to right end, Albert has to traverse the corridor in the opposite direction. Because of the uncertainty of Albert's actions, Ludwig decides to move into a doorway in order to let Albert pass by. The trajectory of Ludwig is depicted by a dashed line, and Albert's trajectory is indicated by a solid line. The position where Ludwig waited for Albert is indicated by the label "wait".

3.2 Competitive Ratio to the Optimal Strategy

In addition to the experiments using Albert and Ludwig, we performed a series of simulation runs in order to evaluate the applicability of the overall approach. An additional goal of these experiments is to demonstrate that our planner outperforms a prioritized variant of the coordination technique described in [15, 11]. To deal with larger groups of robots our current system uses the optimal move costs of the individual robots to determine the priorities. Since the complexity of the coordination technique grows exponentially in the number of robots, we apply the same priority scheme for this approach. For the following experiments we used the B21 simulator [16] which performs real-time simulations of the robot's actions and of its sensors. To get close to the behavior of a real robot, it adds noise to the simulated sensor information.

Figure 4 (left) shows the trajectories carried out by two robots in the situation depicted in Figure 2. As can bee seen in the Figure, the resulting trajectories in this example are quite close to the planned paths. Figure 4 (right) shows the corresponding

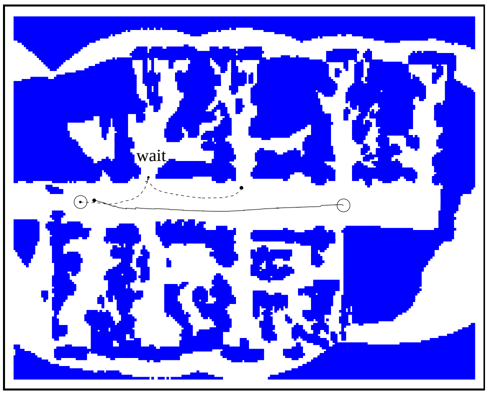
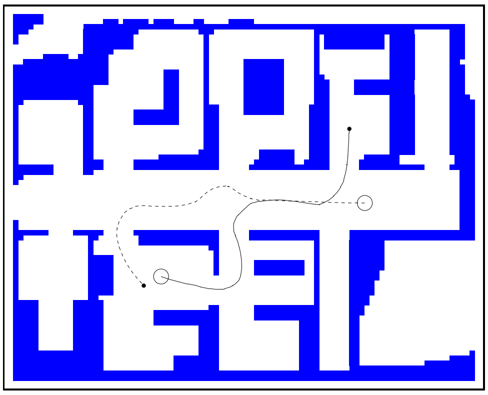


Fig. 3.: Our robots Albert (left), Ludwig (right) and a situation in which Ludwig moves away in order to let Albert pass by.

paths obtained with the coordination diagram technique. Please note that in this situation our technique is significantly better than the coordination technique. Since the coordination technique does not change the trajectories and restricts the robots to stay on their pre-planned paths, the robot starting in the corridor has to wait until the other robot passed by. Therefore, the time to arrive at its target location is almost twice as long as it would be without any conflict. In contrast to that, the two robots arrive almost at the same time using our technique.



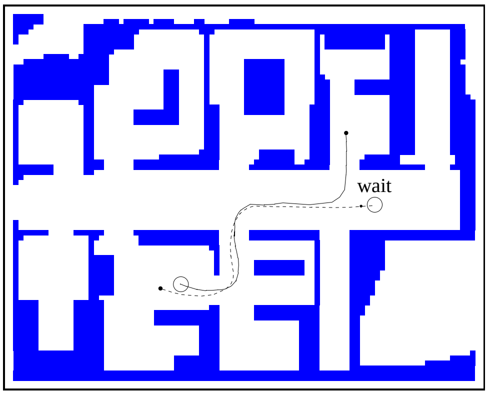


Fig. 4.: Simulation run with the resulting trajectories for the planned paths shown in Figure 2 (left) and the trajectories obtained using the coordination technique (right).

Since the coordination method restricts the robots to stay on their independently planned paths, it does not find a solution in many situations in which our technique is able to determine collision-free trajectories. A typical example is shown in Figure 5 (left). Here two robots have to pass each other in a corridor. Whereas the coordination method cannot resolve this conflict, our planner directs one robot to leave its optimal trajectory and to enter a doorway in order to let the other robot pass by.

To get a quantitative assessment of the performance of our method compared to the optimal strategy and compared to the coordination technique we performed extensive experiments with our simulator.

The first series is designed to compare our probabilistic planning technique to the optimal solution and to the coordination technique. We performed 10 different simulation runs using the environment shown in Figure 2. In each experiment we started two robots at different places and defined target locations for which there is a conflict which can be resolved by the coordination technique. Since our approach is more general than the coordination technique, all three methods were able to compute a solution in these situations. For each solution provided by the individual planners we recorded the sum of the lengths of the two paths, i.e. the number of cells traversed in the map plus the number of time steps each robot waited. In order to be able to compute the optimal solution we had to reduce the resolution of the grid maps to 60×60 cm^2. Figure 5 (right) shows the resulting path lengths for the different runs and the individual planning techniques. Whereas the comparative ratio of our technique relative to the optimal solution was 1.02, the coordination technique needed 1.24 as many steps as the

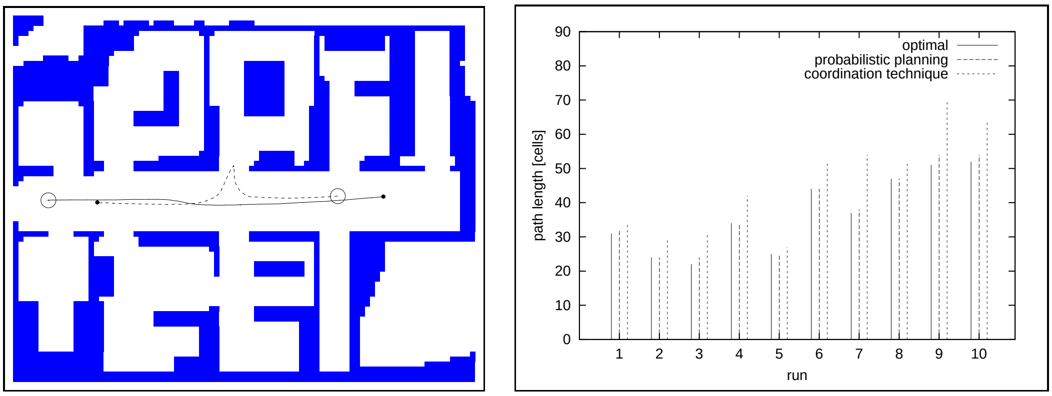


Fig. 5.: Solution generated by our probabilistic planning technique in a situation in which the coordination method does not find a solution (left) and the performance comparison to the optimal solution and to the coordination technique (right).

optimal solution. On the 95% confidence level our approach performed significantly better than the coordination technique. On average, the paths generated by the coordination method were 20% longer than the trajectories generated by our method.

3.3 Comparisons for Larger Numbers of Robots

Additionally, we performed series of simulation runs using two different environments to compare the performance of our probabilistic approach with the performance of the coordination technique for different numbers of robots. Figure 6 depicts the two environments used in the experiments. The first environment shown on the left side of Figure 6 is a typical office environment. The second situation is a rather unstructured environment (see right image of Figure 6) which offers many possibilities for the robots to change their routes. In 9000 experiments we evaluated the path planning techniques for 2 to 6 robots in both environments. The corresponding start and goal positions were randomly chosen from a set of predefined positions.

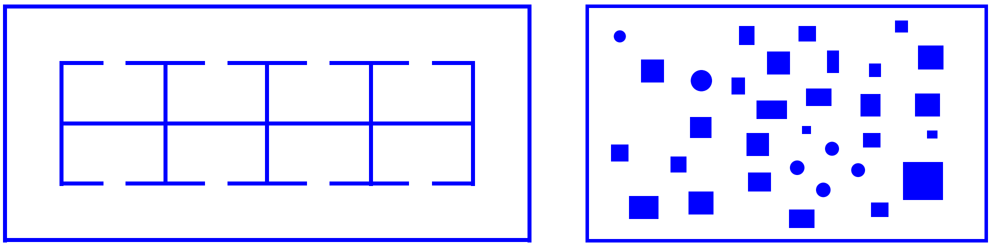


Fig. 6.: Two different environments used for simulation runs.

Figure 7 (left) shows for both environments the average number of conflicts each robot is involved in. Please note that we only evaluated situations in which there was at least one conflict between the robots. As can be seen this number is significantly higher in the office environment than in the unstructured environment because all robots have to travel along the corridor whereas they have a lot more possibilities to choose alternative routes in the unstructured world.

For each number of robots we evaluated 50 experiments in the structured and 100 experiments in the unstructured environment in which there was a conflict between the robots and in which both techniques were able to compute a solution. The priority scheme was to sort the robots according to the optimal move costs between their initial and their goal position. A typical example with four robots is shown in Figure 8 (left). The priorities of the robots and the trajectories computed with our probabilistic planning technique are shown in Figure 8 (right).

In each experiment we measured the sum of the move costs generated by our probabilistic technique and computed by the coordination technique. Since the optimal solutions were not known (and cannot be computed in a reasonable amount of time for more than two robots) we compared the results of the planning techniques with the sum of the optimal move costs for the

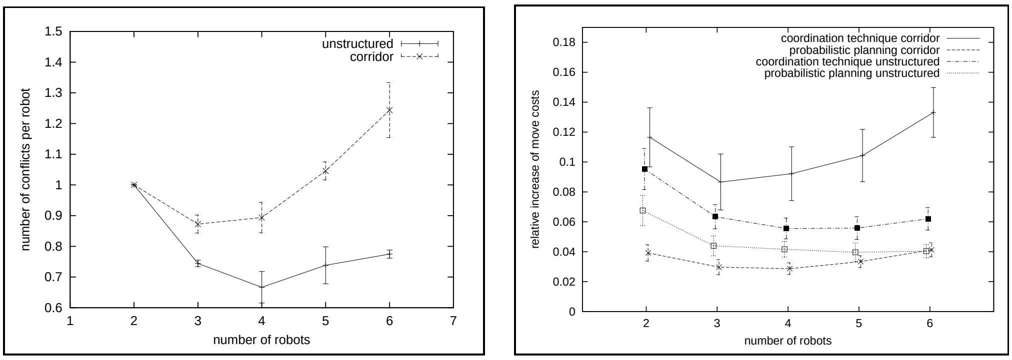


Fig. 7.: Average number of conflicts (left) and the comparison of the relative increase of move costs of the probabilistic technique and coordination technique (right).

individual robots if the paths are computed independently, i.e. in independent single robot problems. Thus, in the experiment described above we compared the resulting move costs of the robots (shown in Figure 8 (right)) with the corresponding costs obtained with the coordination technique both relative to the move costs of the paths in Figure 8 (left).

As can be seen in Figure 7 (right) our method significantly outperforms the coordination technique in both environments. Especially in the office environment the coordination technique frequently forces the robots to wait in a room for longer periods of time until another robot passed by. Since our probabilistic planning technique allows to robots to choose detours in the corridor, the reduction in the average move costs obtained with our probabilistic planning technique is much higher.

As already mentioned in the experiments described above we used the move costs to determine the priority of the individual robots. To evaluate an alternative priority schemes we performed the same experiments using the number of conflicts each robot was involved in to determine the priority of the robots. It turned out that the results obtained with this heuristic do not differ significantly to those obtained when the robots are sorted according to their move costs.

Another interesting aspect is the number of situations in which the different approaches were able to generate a solution. Figure 9 (left) shows for both methods the number of cases in percent in which a solution could be found in the unstructured environment. Obviously, the coordination technique quite often cannot find a solution as the number of robots rises. For example, for 6 robots only 55% of the planning problems could be solved by the coordination technique whereas our probabilistic technique was able to find a solution in 99.3% of the problems.

Figure 9 (right) depicts one of the two planning problems with 6 robots for which our prioritized planning method is not able to find a solution. Since robots 0 and 2 have higher priority their paths are computed first. As a result, robot 4 cannot "escape" so that no path can be found for this robot. Thus, given the fixed priority scheme there is no way to find a path for robot 4.

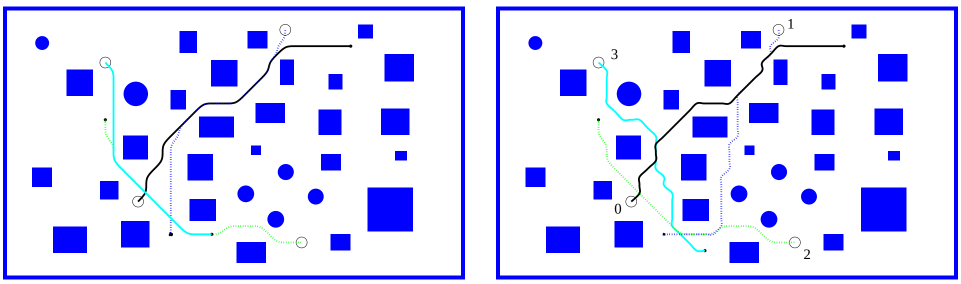
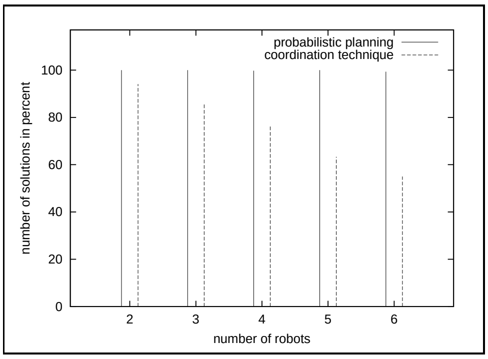


Fig. 8.: Typical experimental setup with four robots including their independently planned and optimal trajectories (left) and the priorities of the robots and paths computed by our probabilistic technique (right).



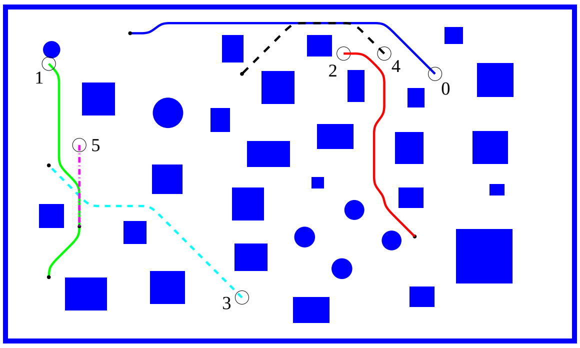


Fig. 9.: Number of cases in percent where a solution could be found in the unstructured environment (left) and a situation in which no solution can be found for robot 4.

4 Conclusions

In this paper we presented an approach to decoupled and prioritized path planning for groups of mobile robots. Our approach plans the paths for the individual robots independently. If a conflict between the paths of some robots is detected it uses a priority scheme to re-plan the path of the robots with lower priority in their configuration time-space. Thereby it considers the constraints imposed by the robots with higher priority. Our approach uses occupancy grid maps to plan the motions of the robots using A^*. Simultaneously it trades off the length of the trajectory and the distance to objects in the environment. It furthermore uses a probabilistic model to integrate possible deviations of the robots from their planned paths into the planning process. Therefore, the resulting trajectories are robust even in situations in which the actual trajectories of the robots differ from the pre-planned paths.

Our method has been implemented and tested on real robots. The independent planning of the paths for the individual robots is highly efficient and requires not more than 0.01 seconds. Additionally, the system can rather quickly resolve conflicts. For the examples in the map of our department the computation of a collision-free path in the configuration time-space generally requires less than 6 seconds using a spatial resolution of 20×20 cm^2 and less than 1.5 seconds for a cell size of 40×40 cm^2. Please note that this computation time will not significantly increase in the number of robots, since our approach uses lookup-tables to store the costs introduced by the previously planned robots.

In all experiments our approach showed a robust behavior. We performed a series of experiments with up to 6 robots to compare our technique to the coordination method [15, 11]. These experiments demonstrate that our approach significantly outperforms the coordination technique.

Apart from the promising results presented in this paper, there are different aspects for future research. First, in our current implementation we assume equal constant velocities of the robots. In practice, teams often are inhomogeneous and contain different types of robots with different average velocities which has to be taken into account when computing the probability that a robot is at a certain location. Our approach currently uses a fixed priority scheme. More flexible assignments of priorities to the individual robots will with high likelihood result in more efficient solutions. Furthermore, our system currently does not react to larger differences during the plan execution. For example, if one robot is delayed because unforeseen objects block its path, alternative plans for the other robots might be more efficient. In such situations it would be important to have means for detecting such opportunities and to re-plan dynamically. On the other hand, the delay of a single robot may result in a dead-lock during the plan execution. In this context, the system requires techniques for detecting dead-locks while the robots are moving and to resolve them appropriately.

References

1. J. Barraquand, B. Langois, and J. C. Latombe. Numerical potential field techniques for robot path planning. Technical Report STAN-CS-89-1285, Department of Computer Science, Stanford University, 1989.
2. 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.
3. 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.

4. S. J. Buckley. Fast motion planning for multiple moving robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1989.
5. M. Erdmann and T. Lozano-Perez. On multiple moving objects. *Algorithmica*, 2:477–521, 1987.
6. C. Ferrari, E. Pagello, J. Ota, and T. Arai. Multirobot motion coordination in space and time. *Robotics and Autonomous Systems*, 25:219–229, 1998.
7. D. Grossman. Traffic control of multiple robot vehicles. *IEEE Journal of Robotics and Automation*, 4:491–497, 1988.
8. J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
9. 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.
10. 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.
11. 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.
12. V. J. Lumelsky and K. R. Harinarayan. Decentralized motion planning for multiple mobile robots: The cocktail party model. *Journal of Autonoumous Robots*, 4:121–135, 1997.
13. 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.
14. 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.
15. 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.
16. D. Schulz, W. Burgard, and A.B. Cremers. Robust visualization of navigation experiments with mobile robots over the internet. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1999.
17. 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)*, pages 1631–1636, 1995.
18. 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.
19. J. Wang and S. Premvuti. Distributed traffic regulation and control for multiple autonomous mobile robots operating in discrete space. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 1619–1624, 1995.
20. 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 SIRS2000 style