Coordinated Navigation for Multi-Robot Systems with Additional Constraints

(Extended Abstract)

Bernd Brüggemann
Fraunhofer FKIE
Neuenahrer Str 20
Wachtberg, Germany
bernd.brueggemann@fkie.fraunhofer.de

Dirk Schulz
Fraunhofer FKIE
Neuenahrer Str 20
Wachtberg, Germany
dirk.schulz@fkie.fraunhofer.de

ABSTRACT

In this paper we present a method for navigating a multi-robot system through an environment while additionally maintaining a set of constraints. Our approach is based on graph structures that model movements and constraints separately, in order to cover different robots and a large class of possible constraints. Additionally, the partition of movement and constraint graph allows us to use known graph algorithms like Steiner trees to solve the problem of finding a target configuration for the robots. We construct so called separated distance graphs from the Steiner tree and the underlying roadmap graph, which allow to assemble valid navigation plans fast.

Categories and Subject Descriptors

I.2.9 [**Robotics**]: Autonomous vehicles

General Terms

Algorithms, Theory

Keywords

robot coordination, multi-robot systems, robot planning

1. INTRODUCTION

In this work, we look at the problem of planning of joint task execution for multi-robot systems. More specifically, we investigate the problem of coordinated multi-robot navigation, if additional constraints need to be fulfilled. These are constraints which are not directly related to the goal to be achieved, but have to hold throughout task execution. Such constraints are e.g. to ensure that each individual robot never departs from the group by more than a certain distance or to keep up the wireless communication within a complete group of robots. Here we propose a graph-based approach that allows to specify and maintain such constraints.

Cite as: Coordinated Navigation for Multi-Robot Systems with Additional Constraints (Extended Abstract), Bernd Brüggemann, and Dirk Schulz, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. . 1511-1512

2. BASICS AND DEFINITIONS

Our approach basically computes coordinated navigation plans on a roadmap graph $G_m = (E_m, V_m)$ within the free space of the environment. We assume that G_m forms a regular grid within free space. Each vertex $v \in V_m$ represents a position in the environment and the edge cost correspond to the path length between the corresponding positions in the environment. Our aim is to move a group of robots from a given start configuration S_0 to some target configuration S_t. Here, a *configuration* for a group of robots is a vector that assigns each robot to a vertex. A *motion action* transfers a configuration S of the multi-robot system into a new configuration S' by "moving" a robot along some edge of G_m. Searching a sequence of motion actions to reach S_t can be performed by using e.g the A*-algorithm. However the resulting solutions do not guarantee that the constraint holds during the whole motion sequence.

So we introduce a *constraint graph* G_c. The vertex set V_c of this graph coincides with V_m and the edges in E_c formalize binary constraints between two positions in the environment. Intuitively, such a constraint models a physical requirement in the real world. For example, if we want to ensure the wireless communication between two robots located at positions v and v', we have to make sure that the expected signal strength $L(v, v')$ between the two vertices exceeds a certain threshold L_{thresh}.

In our approach we use the constraint graph to construct complete configurations of the multi-robot system. We say a configuration S is *valid* if G_c restricted to the vertices in S is a single-component sub-graph of G_c. We say a motion action $a(S, S')$ is valid, if it transfers the valid configuration S into a valid configuration S'. The overall task can therefore be divided into two separate parts: First, for a given constraint graph G_c and a set of targets Z, find a valid target configuration S_t. Second, find a sequence of valid motion actions, that transfers S_0 into S_t via a sequence of valid intermediate configurations S_j.

3. CONSTRAINED MOTION COORDINATION FOR MULTI-ROBOT SYSTEM

The task of finding a valid target configuration for the multi-robot system involves placing robots at each mission target, and placing additional robots on vertices in such way, that the resulting configuration is valid. From the definition of a valid configuration one can obtain a valid target configuration by constructing its corresponding single-component constraint sub-graph which includes all mission targets. This task is closely related to the Steiner tree problem [2].

Considering the set of targets Z as terminals, the Steiner tree problem exactly matches the task of finding the target configuration. The Steiner tree for the constraint graph G_c is one possible valid target configuration S_t. In fact, this solution achieves optimality with respect to the number of robots required. Unfortunately, computing the Steiner tree is NP-complete, but several heuristics exist which compute near optimal Steiner trees in polynomial time [1].

With the achieved target configuration, all robots have to reach certain target vertices (either a Steiner vertex or a terminal) in the target configuration S_t and the global constraint between the robots has to be maintained throughout the navigation. While the robots' motions are planned based on the navigation roadmap, the constraint can only be checked based on the constraint graph. To plan valid paths, we therefore frequently have to map between constraint paths in G_c and motion paths in the navigation roadmap G_m. Additionally, to navigate a robot to its target vertex, further robots may be needed to build up relays to ensure the global constraint.

To ease this planning problem, we introduce a graph structure that we call Separated Distance Graph (SDG). A SDG $SDG(v) = (V_S(v), E_S(v), W)$, is constructed for a specific vertex v; its vertices $V_S(v)$ are v and all vertices directly connected to v in the constraint graph. Let $G_{\mathrm{m}}^{sub}(v)$ be the navigation roadmap restricted to vertices in $V_S(v)$. There is an edge between two vertices v and v_j in $SDG(v)$ if there exists a path between these vertices in $G_{\mathrm{m}}^{sub}(v)$.

Intuitively, an edge $(v_i, v_j) \in E_S(v)$ states the existence of a navigation path between these vertices that a robot can travel without violating the binary constraint with a robot located at the vertex v. The set of separated distance graphs for a given planning problem now allows us to derive a search algorithm for an abstract navigation plan that achieves the target configuration.

However, implementing this approach is not straight forward. Although two neighboring vertices in the Steiner tree always fulfill the constraint that has to be ensured, it is generally not possible to navigate a single robot on a path through G_m to this vertex, without violating the constraint at some intermediate position. This happens, e.g. if reaching a position requires lengthy detours due to obstacles. So in the general case, additional temporary relay-robots are required. To compute a valid navigation plan from a vertex a of the Steiner tree to a neighboring vertex b, we therefore perform a breadth-first search of valid motions through G_m with the help of the SDGs. By construction, the breadth-first search returns the path from a to b with the smallest number of intermediate relay-robots required.

On each temporary relay position, we leave one robot behind, in order to ensure the constraint. As soon as the remaining robots reach the next target location, the temporary relay-robots can catch up, without violating the constraint, e.g. by moving the farthest robot first. So the planning problem can be carried out by the following steps:

1. Given the targets Z, a navigation roadmap G_m for the environment and a symmetric binary relation that should hold between robots, we compute the corresponding constraint graph G_c for the environment.
2. Given G_m and G_c we compute SDGs for all vertices.

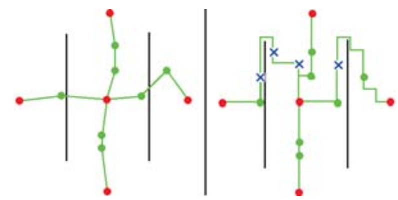

Figure 1: Distance constraint, bar-shaped obstacles. Dark (red) dots: terminals, light (green) dots: Steiner vertices, cross: temporary relays robot

3. Given Z and G_c, we compute a valid target configuration $S_t = (s_t^1, \ldots, s_t^K)$ by approximately solving the associated Steiner tree Problem.
4. Given s_0, S_t and the set of SDGs, we compute the temporary relay positions required to traverse the edges of the Steiner tree corresponding to S_t. This procedure also gives an upper bound on the number of robots required to execute the task.
5. The final plan now consists of the tree traversal of the Steiner tree with all robots, placing a robot at each target location, as well as placing and removing temporary relay-robots on the way to the next target location.

One example planing for a constraint concerning the distance between the robots can be seen in 1). The distance constraint ensure that no robot will be further away from the group than a certain threshold. In this example the world consist of two bar like obstacles. As the constraint is not affected by any obstacles, the Steiner tree, representing the target configuration, can have edges crossing obstacles. In this cases, it is not possible to follow some paths to target locations with a single robot without violating the distance constraint, so temporary relay-robots are needed.

4. CONCLUSION

In this work we presented a graph based approach to navigate a multi-robot systems while guaranteeing to fulfill a global group constraint during navigation. For this purpose, the possible movements of the robots and the constraint are represented by independent constraint and navigation roadmap graphs. We showed that the problem of finding valid target configurations can be solved using a approximate solution of the Steiner tree Problem. The Steiner tree can also be used as an abstract plan to reach the target configuration, but generally additional relay-robots are required to ensure the constraint during navigation. We introduced a new graph structure called Separated Distance Graph (SDG) that links the constraint graph with the navigation roadmap and allows to identify the additional relay positions using a breadth-first search technique.

5. REFERENCES

[1] C. Gröpl, S. Hougardy, T. Nierhoff, and H. J. Prömel. Lower bounds for approximation algorithms for the steiner tree problem. In *Proceedings of 27th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2001, Springer LNCS*, pages 217–228. Springer Verlag, 2001.

[2] F. K. Hwang and D. S. Richards. Steiner tree problems. *Networks*, 22(1):55–89, 1992.