From a Multi-robot Global Plan to Single-robot Actions

Bernd Brüggemann[1], Elmar Langetepe[2], Andreas Lenerz[2] and Dirk Schulz[1]
[1]*Research Group "Umanned Systems", Fraunhofer Institute for Communication, Information Processing and Ergonomics, Neuenahrer Str 20, Wachtberg, Germany*
[2]*Department of Computer Science I, University of Bonn, Bonn, Germany*

Keywords: Multi-robot Systems, Planning, Plan Execution, Coordination, Algorithmic Geometry.

Abstract: Planning for and coordinating robots in a multi-robot system (MRS) is crucial for optimizing the performance of the whole MRS. Thus a plan for the movement of the MRS must exist. If some centralized entity calculates the plan, it may result in one plan for the whole group. Such a global plan, which shows how the MRS can reach a goal state, has to be transformed into action guidelines for each robot. This task becomes harder if such a global plan includes dependencies caused by necessary cooperation of the robots. In this paper we present an approach to transform a global multi-robot plan into single-robot actions. We also provide a method to determine how many robots are needed to fulfil the global plan while obeying some constraints. Here we use plans generated by a coordinated navigation planner with spatial constraints, but the method could be expanded to a more general class of plans built from a centralized entity.

1 INTRODUCTION

Currently, decentralized planning for multi-robot systems is known as the more flexible and robust way of coordinating robots. But often this results in a more autonomous and a more "free" behaviour of the single robots (see, for example, (Alami et al., 1998)). As the robots have to coordinate themselves and figure out how to achieve the goal, their autonomous abilities have to be strengthened. Additionally, the behaviour of each robot cannot be predicted in advance very often. Thus, the operator has to accept that the robot shows unpredictable behaviour to some extend (for more information about decentralized MRS see, for example, (Parker, 2008)).

In most cases such an autonomous decentralized behaviour of the robots is desirable and benefiting, especially if the robustness of such approaches is needed. But there are certain situations imaginable where the behaviour of the robot group should be as predictable as possible. In such situations (like dangerous environments where the danger cannot be detected by the robots) a centralized coordination of the robots may provide advantages.

Another precondition for an advantageous use of MRS is a proper coordination of the robots. Such coordination often leads to constraints the robots have to obey. One of the most common constraints a MRS has to fulfil is to keep up the communication between the robots. This results in a MRS plan with spatial and temporal constraints: the robots have to be at a certain place on a certain time.

In this paper we present a method to determine how many robots are needed to execute the constrained multi-robot global plan and how to transform it into single actions for each robot. This multi-robot plan does not only consist of a planned path for the MRS but also of commands how to follow possible constraints resulting from the robots' coordination. The resulting approach can automatically transform a certain group of global plans into plans for each robot.

As there are numerous applications and algorithms for centralized multi-robot control, there are also several ways to deal with the transition of the general MRS goal into the actions of the single robots. In (Burgard et al., 2000) a multi-robot system has to explore an unknown environment. In their approach a central entity chooses target positions for each robot by estimating the information gain of those positions. In (Alami and da Costa Bothelho, 2002) a general planning framework for a multi-robot system is presented. Here centralized and decentralized planning algorithms are combined. The global plan is formulated in a chain of subtasks which can be understood by the corresponding robots. The tasks can be transformed into robot actions on the robot as they are predefined behaviours. Sanchez and Latombe show that some problems occur when transferring centralized

Brüeggemann B., Langetepe E., Lenerz A. and Schulz D..
From a Multi-robot Global Plan to Single-robot Actions.
DOI: 10.5220/0004016304190422
In *Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2012), pages 419-422
ISBN: 978-989-8565-22-8
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

to decoupled planning. In (Sanchez and Latombe, 2002) they show that sometimes a solution cannot be found because the decoupled planning process loses the completeness. Although it is not expected that the decentralized approach fails in general, you can expect that the solutions will be slightly worse than the centralized ones. Decoupled planning also often needs some mechanisms to detect and to solve deadlocks (like in (Ryan, 2010) or (Kaminka et al., 2010)).

In contrast to global planners which are planning directly on robot actions our approach introduces an intermediate step between global planning and local action execution. This enables the use of global plans which are better understandable for human operators.

The remainder of the paper is as follows: first, in section 2, constrained global plans as we are using them for multi-robot systems are defined and motivated. As such global plans do not directly report about the resources needed to execute them, we analyse the question how many robots are needed. This results in an algorithm which not only gives the needed number of robots but also a visiting sequence for the target positions. Section 3 shows the resulting approach to compute a plan for each robot. We close the paper with our conclusions and some words about future work.

2 GLOBAL PLANS

As stated in the introduction, we have to coordinate the robots to take advantage of the MRS. So in a MRS the robots can, for example, serve as communication relays. Thus, to keep up communication to robots far away from the operator additional robots have to be placed. The need of robots as relay station can be viewed in a more general way to allow additional constraints besides communication. A centralized planner for such constrained navigation problems can be found in (Brüggemann and Schulz, 2010). Such an application leads to a different kind of global plan, the *constrained multi-robot global plan*. Here the position of each robot is chosen so the constraint is not violated. Each *position* is either a target position for one robot or a relay position a robot has to take over. As such relay positions assure that the constraint is not violated, a robot is not allowed to move any more as soon as it is placed there.

As the constraint should not be violated during the execution of the plan, the order of arrival at the positions is crucial. Additionally, there are situations in which the constraint is obeyed between the relay positions but not at the path in between. In such cases additional temporary relays are needed. Such temporary relay robots are needed during the movement of other robots, but when all robots have reached the next relay position, the temporary relay robots are allowed to move away. An example of such an constrained multi-robot global plan can be seen in figure 1.

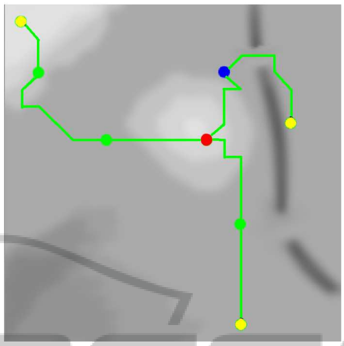

Figure 1: An example of a constrained multi-robot global plan. The red marker indicates the starting position. The green markers are relay positions. These are necessary to keep up the communication between the starting point and the target positions (yellow markers). The blue marker is a temporary relay position. To reach the rightmost target position a robot has to be placed there. If it is taken, a new task can be assigned to the robot on the temporary relay position.

3 TRANSFER GLOBAL PLANS TO LOCAL ACTIONS

To transform the constrained global multi-robot plan into local robot actions we have to determine at first how many robots are needed to fulfil the plan without violating the constraint. As normally the number of robots is limited within a MRS, we try to minimize the number of robots needed to execute the plan. This turns out to be a difficult problem due to the temporary relay robots. They are needed to get one or more robots over certain paths of the plan but then can be used somewhere else. Hence, if there are temporary relay robots needed ,there are situations in which they raise the total amount of needed robots and other situations in which they do not.

To get a more formal view of the problem we reformulate the global plan into so called navigation trees. Here the relay positions are the nodes. Two nodes are connected if there is a path between them in the global plan. Each edge in the navigation tree has a weight equal to the number of temporary relay positions on the corresponding path. A transformation of a global plan into a navigation tree can be seen in figure 2.

With the help of the navigation tree, the problem

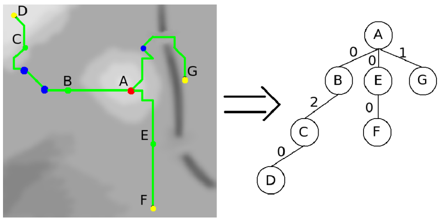

Figure 2: Transformation of a global plan into a navigation tree.

of finding the lowest number of robots needed for the global plan can be reformulated as multi-agent graph traversal problem: Find the lowest number of agents which are needed to visit every node of the navigation tree while obeying the following rules:

- If an agent visits an unvisited node, that agent has to stay there.
- To pass an edge the number of agents must be at least
  - equal to the edge weight if the following node is already visited, and
  - equal to 1 plus the edge weight if the following node is not yet visited.

A sequence S which obeys the rules above and needs n robots provides a base for the translation of global maps into local actions. Additionally, we want to minimize n. This results in the following algorithms:

Algorithm 1 computes the number of robots needed to visit every node in the sub-tree of v_m and return to v_m. The *dominating edges* are those edges which have a higher weight than any other edge in their sub-tree. A *cluster* is the sub-tree of a dominating edge. Algorithm 2 computes the minimal number of robots needed to traverse the navigation tree. The traversal of the robots can be expressed as a sequence of leafs visited. Unfortunately, trying out every possible sequence would result in exponential computing time. However, it can be shown that all leafs, except one leaf the robot group stops at, have to be visited in the descending order of their corresponding dominating edges' weights. Thus, algorithm 2 counts how many robots are needed for each leaf taken as the leaf the robots have to stop at. This not only gives the minimal number of robots needed to execute the global plan but also the visiting order of all target positions.

With this visiting order and the following notation we are able to transform the global plan into local robot actions:

- $\rightarrow A$: Go to position A
- A': Wait at position A until the robot which stays at A arrives
- $||x,y$: Wait until the robots with number x and y have reached the next relay node

Algorithm 1: Count number of robots when returning to node v_m.

procedure TRAVERSE1(node v_m)
  Be v_r the node the group of agents is at the moment
  for every node $v' \neq v_m$ on the path $\pi = v_m \ldots v_r$ **do**
    if v' is not visited before **then**
      mark v' as visited
      change m (total robots) and u (robot available) corresponding to the need of v'
    end if
  end for
  Let cluster C be the first unvisited cluster in v_m
  if $u <$ # nodes in $C+$ weight of e_m **then**
    $m = m +$ (# nodes in $C+$ weight of $e_m - u$)
    $u =$ # nodes in $C+$ weight of e_m
  end if
  mark the dominating edge $e_m(C)$ as visited
  $u = u$ - # nodes in $C+$ weight of e_m
end procedure

Algorithm 2: Find minimal number of robots needed and corresponding visiting sequence.

procedure TRAVERSE2(node v_m)
  for all clusters C_v in v_m **do**
    mark C_v as visited
    TRAVERSE1(v_m)
    mark C_v as unvisited
    TRAVERSE2(v)
    if $k_{opt} >$ counter **then**
      $k_{opt} =$ counter
    end if
    mark all clusters as unvisited
  end for
end procedure

The two wait commands are necessary to ensure that no robot violates the constraint given by the constrained global map as well as to gather enough robots to pass the edges. At first we assume that we move all robots as one group. Then we present an optimization of the robot actions which results in a parallel execution of those actions.

To find an optimal visiting sequence, we build the action guideline of each robot step by step. Each robot gets its own plan. The path is split into sub-paths consisting only of positions and temporary nodes. Whenever a robot receives the command "move to X" "$\rightarrow X'$" is added to the action guideline. Please notice that we always add the command "wait until robot for that node is there (')". This guarantees that the con-

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Action guidelines:
$0 :\to A$
$1 :\to A' \to B$
$2 :\to A' \to B' \to tR1||5,6,7,8 \to tR2' \to tR3' \to D$
$3 :\to A' \to B' \to tR1' \to tR2||5,6,7,8,2 \to tR2' \to D' \to tR4||4,5,6,7,8 \to v_{l,2}$
$4 :\to A' \to B' \to tR1' \to tR2' \to tR3||5,6,7,8,2,3 \to D' \to tR4' \to v'_{l,2} \to tR4||5,6,7,8 \to D' \to v_{l,1}$
$5 :\to A' \to B' \to tR1' \to tR2' \to tR3' \to D' \to tR4' \to v_{l,2} \to tR4' \to D' \to v'_{l,1} \to D' \to tR3||8 \to tR2' \to tR1' \to B' \to tR5||6,7,8 \to v_{l,3}$
$6 :\to A' \to B' \to tR1' \to tR2' \to tR3' \to D' \to tR4' \to v'_{l,2} \to tR4' \to D' \to v'_{l,1} \to D' \to tR3' \to tR2||8,5 \to tR1' \to B' \to tR5' \to v'_{l,3} \to tR5||7,8 \to B' \to A' \to tR6||7,8 \to C$
$7 :\to A' \to B' \to tR1' \to tR2' \to tR3' \to D' \to tR4' \to v'_{l,2} \to tR4' \to D' \to v'_{l,1} \to D' \to tR3' \to tR2' \to tR1||8,5,6 \to B' \to tR5' \to v'_{l,3} \to tR5' \to B' \to A' \to tR1' \to C' \to tR7||8 \to E$
$8 :\to A' \to B' \to tR1' \to tR2' \to tR3' \to D' \to tR4' \to v'_{l,2} \to tR4' \to D' \to v'_{l,1} \to D' \to tR3' \to tR2' \to tR1' \to B' \to tR5' \to v'_{l,3} \to tR5' \to B' \to A' \to tR6' \to C' \to tR7' \to E' \to v_{l,4}$

Figure 3: Resulting action guidelines from the navigation tree in figure 4.

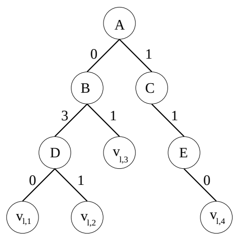

Figure 4: Example navigation tree.

straint is never violated. It also does no harm because, if the robot for that node is already there, then this command has no effect.

We also have to deal with temporary relay positions, which we name tR_x. While crossing edges with a weight, a defined number of robots is needed. They are placed on the tR_x nodes and have have to stay there until the whole group reaches the next non-tR_x node. After that the temporary robots can also go forward. The robot farthest away from the position of the group has to move first, then the second farthest, and so on.

Up to this point we have moved all robots from the beginning in one group. In most cases this is not necessary. Thus, we propose an optimizing step which can be done after calculating the action guidelines for the robot group. This optimization uses the fact that a robot which passes a node v_i in the direction of the leafs and back without getting at least one "||" command in between is not necessary. So all the commands between two appearances of v_i can be removed. However, this remove operation has some side effects: As the waiting commands "||" are always related to the whole robot group, a robot whose action guideline was partly removed has to be removed from the waiting lists of the other robots. A resulting plan transformation can be seen in figure 3.

4 CONCLUSIONS

In this paper we address the problem of transforming a constrained global multi-robot plan into plans for each single robot in that multi-robot system. Therefore, we introduce navigation trees to determine the minimal number or robots for a general global plan. This results not only in the minimal number of robots but also in a visiting sequence for the target positions. Although the combined approaches, building a global plan with our planning approach together with the automatic translation to robot action guidelines are tested on real robots (on a MRS with up to 6 robots) an exhaustive evaluation of the performance is necessary. Especially the online optimization via interchanging guidelines is a focus of future work.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

REFERENCES

Alami, R. and da Costa Bothelho, S. (2002). Plan-based multi-robot cooperation. In Beetz, M., Hertzberg, J., Ghallab, M., and Pollack, M., editors, *Advances in Plan-Based Control of Robotic Agents*, volume 2466 of *Lecture Notes in Computer Science*, pages 65–95. Springer Berlin / Heidelberg. 10.1007/3-540-37724-7_1.

Alami, R., Fleury, S., Herrb, M., Ingrand, F., and Robert, F. (1998). Multi-robot cooperation in the martha project. *Robotics Automation Magazine, IEEE*, 5(1):36–47.

Brüggemann, B. and Schulz, D. (2010). Coordinated navigation of multi-robot systems with binary constraints. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE Computer Society.

Burgard, W., Moors, M., Fox, D., Simmons, R., and Thrun, S. (2000). Collaborative multi-robot exploration. volume 1, pages 476–481 vol.1.

Kaminka, G., Erusalimchik, D., and Kraus, S. (2010). Adaptive multi-robot coordination: A game-theoretic perspective. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 328–334. IEEE.

Parker, L. E. (2008). Distributed intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, 2(1).

Ryan, M. (2010). Constraint-based multi-robot path planning. In *IEEE International Conference on Robotics and Automation*.

Sanchez, G. and Latombe, J.-C. (2002). Using a prm planner to compare centralized and decoupled planning for multi-robot systems. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 2112–2119 vol.2.