Coordinated Exploration with Marsupial Teams of Robots using Temporal Symbolic Planning

Kai M. Wurm Christian Dornhege Patrick Eyerich Cyrill Stachniss Bernhard Nebel Wolfram Burgard

***Abstract*— The problem of autonomously exploring an environment with a team of robots received considerable attention in the past. However, there are relatively few approaches to coordinate teams of robots that are able to deploy and retrieve other robots. Efficiently coordinating the exploration with such marsupial robots requires advanced planning mechanisms that are able to consider symbolic deployment and retrieval actions. In this paper, we propose a novel approach for coordinating the exploration with marsupial robot teams. Our method integrates a temporal symbolic planner that explicitly considers deployment and retrieval actions with a traditional cost-based assignment procedure. Our approach has been implemented and evaluated in several simulated environments and with varying team sizes. The results demonstrate that our proposed method is able to coordinate marsupial teams of robots to efficiently explore unknown environments.**

I. INTRODUCTION

The problem of autonomously exploring an environment is one of the fundamental problems for autonomous mobile robots. There are several applications in which robots have been designed to explore their environment such as planetary exploration or in disaster missions. Using a coordinated team of robots instead of a single robot offers advantages such as fault tolerance or performance gains. The problem of multi-robot exploration with homogeneous robots is relatively well understood. Popular approaches to coordinate such teams estimate the cost and the expected information gain of exploring a target location to find optimal assignments of robots to targets [2, 20, 24].

In several exploration scenarios, however, one needs to consider heterogeneous teams of robots with different capabilities. For a task such as the autonomous exploration of lunar craters [6], one can imagine robots that approach the crater and then deploy a specialized robot which descents into the crater. Robots that are able to deploy and retrieve other robots have also been referred to as *marsupial robots* [17]. Such heterogeneous robots typically require to carefully plan deployment and retrieval actions and to properly take into account the different properties of the robots such as their sensor setup, their size and payload, their maximum traveling speed, or the type of terrain they are able to traverse.

This paper addresses the problem of coordinating a team of marsupial robots that explore an unknown environment.

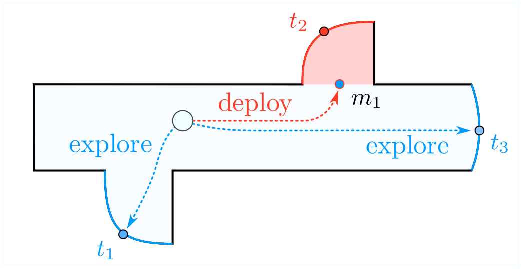

Fig. 1. An exploring robot (white circle) has to choose between three possible actions: explore target t_1, explore target t_3, or deploy a smaller robot at m_1 to let it explore t_2 in the red (dark) area.

In this scenario, a team of robots carries and deploys a number of smaller robots. These smaller robots are able to explore parts of the environment that the bigger robots cannot enter. From a conceptual point of view, the ability to deploy and retrieve robots using other autonomous robots introduces corresponding symbolic actions. To reveal the full potential of such a heterogeneous system, these actions need to be considered during coordination (see Fig. 1 for an illustration). Unfortunately, it is not straightforward to map such actions to cost or utility measures such as those used by the popular target assignment approaches [2, 16, 19, 24].

The problem of planning and executing actions such as deployment and retrieval in an exploration scenario has previously been approached using manually designed strategies [3, 17, 18]. Such hand-crafted strategies, however, are specific to a certain type of environment and it is unclear whether they are able to efficiently coordinate large teams of robots. The contribution of this paper is a novel coordination approach for multi-robot exploration that assigns robots to exploration targets and additionally plans symbolic actions such as deployment and retrieval actions. To achieve this, we integrate a temporal symbolic planner and a traditional path planner for coordinated exploration. In this way, we obtain a more general robot coordination approach that is able to efficiently solve the exploration task.

II. RELATED WORK

Several previous approaches consider the task of coordinating the actions of a team of equally equipped (homogeneous) robots exploring an unknown environment. In this setting, the coordination task is often formulated as an assignment problem where the robots are assigned to exploration targets according to a cost measure. Different methods have been presented to determine such an assignment. Burgard *et al.* [2] present an iterative assignment

All authors are with the University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany.

This work has been supported by the German Research Foundation (DFG) under contract number SFB/TR-8 and by the EU as part of the Integrated Project CogX (FP7-ICT-2xo15181-CogX).

method based on the estimated cost of reaching a target and visibility constraints of robots in the team. Ko *et al.* [16] and Stachniss [19] present approaches that uses the Hungarian method to compute the assignments of frontier cells [23] to robots. Zlot and colleagues [24] propose an architecture in which the exploration is guided by a market economy. They consider sequences of potential target locations for each robot and trade tasks between the robots using single-item first-price sealed-bid auctions. Such auction-based techniques have also been applied by Berhault *et al.* [1] to assign robots to bundles of targets so that synergy effects between targets are taken into account. In a previous work, we present an approach that uses a segmentation of the environment [22]. By assigning robots to unexplored segments instead of frontier targets, a more balanced distribution of the robots over the environment is achieved and the overall exploration time is reduced.

An approach towards cooperation in heterogeneous robot systems is presented by Singh and Fujimura [18]. If a robot is too big to pass through a narrow passage, it informs other robots about this task. Howard *et al.* [14] present an incremental deployment approach that explicitly deals with obstructions, i.e., situations in which the path of one robot is blocked by another. A further heterogeneous system is presented by Grabowski and Navarro-Serment [10]. In this system, however, coordination is performed manually.

Whenever small robots with low traveling speeds or limited power resources are used in a heterogeneous robot team, it is favorable to have larger robots, the *marsupial robots*, transport the smaller ones to avoid a serious penalty in exploration time or power consumption [17]. Denner and Papanikolopoulos present a deployment method for such a marsupial team that explicitly takes power constraints into account [5]. Murphy *et al.* [17] present a physical implementation of a marsupial system and describe heuristics to deploy the micro-rover. Kadioglu and Papanikolopoulos [15] present a further physical implementation. In [3], a team of legged robots is deployed by a carrier robot in a rescue scenario. In all of the previously described exploration systems, deployment and retrieval in marsupial teams is determined by heuristics. In contrast to that, the approach presented in this paper explicitly takes these actions into account when coordinating the exploration.

Domain independent planning is a thoroughly investigated sub-field in artificial intelligence. A classical planning problem consists of a set of state-variables with finite domains, an initial state, a set of actions and a set of goal states. An action is defined by a precondition and its effects, which is a set of variable assignments. A solution for a classical planning problem is then a finite sequence of actions from the initial state to a goal state. There exist several efficient planning systems for classical planning problems [12, 13].

When temporal constraints are specified by admitting actions to have variable durations and to be executed concurrently, one refers to that as *temporal planning*. Several efficient approaches for temporal planning have been presented [7, 9]. The predominant approach of solving planning problems is forward search guided by a heuristic using A^* or similar algorithms. Most approaches to temporal planning allow the usage of numerical state variables. In contrast to binary and multi-valued state variables, numeric state variables have an infinite continuous value domain. While numeric state variables lead to undecidability even when used in a very limited form [11], they are considered to be of high importance when modeling real world domains.

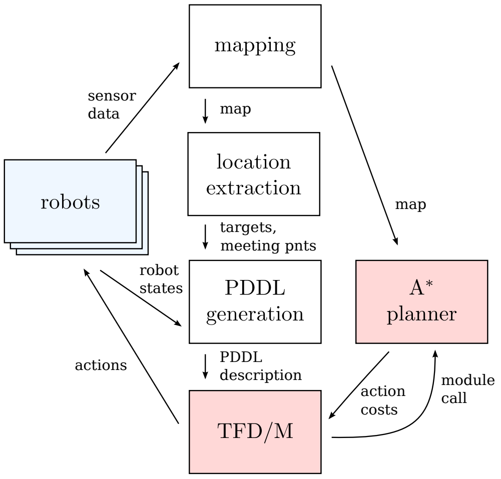

Fig. 2. System overview.

The work described in this paper builds upon TFD/M [4], a variant of the temporal fast downward planning system [7]. TFD/M is an efficient planning approach that searches directly in the space of time-stamped states. It additionally supports the use of external modules via state variables whose values are calculated by sub-processes during the planning process. By means of sub-processes, we combine temporal planning with path planners traditionally used for multi-robot coordination.

III. COORDINATED EXPLORATION WITH MARSUPIAL TEAMS

Throughout this paper, we assume global and unlimited communication between the robots and employ a centralized approach. Furthermore, all robots are assumed to have known relative positions. To achieve this in our experiments, the robots actually start from the same place in the environment.

A marsupial team consists of two types of robots that explore the environment. We consider n *carrier* robots. Each carrier initially carries m smaller robots called *rovers* which can be deployed and retrieved by the carriers. The key challenge is to generate exploration targets, to plan trajectories for the carriers and rovers, and to schedule deployment and retrieval actions at meeting points in the environment. Especially for efficient retrieval, one needs to consider the time the individual robots need to carry out their actions. This together with the fact that the robots operate in parallel makes the problem a *temporal* planning problem.

A. Overview of the Exploration Framework

The architecture of our exploration system is displayed in Fig. 2. The robots provide the sensor data and states

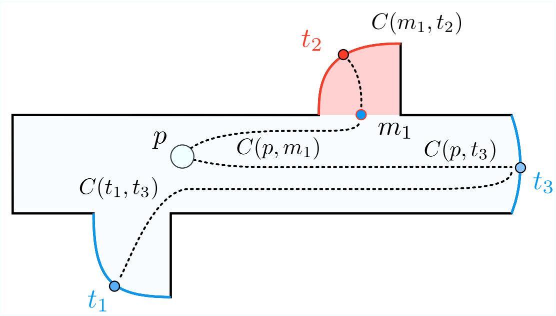

Fig. 3. Example of the costs that have to be considered. Dotted lines illustrate the estimated path costs between the robot pose p and the different target positions t_i, the costs between meeting points m_i and robot pose or targets positions, and costs between target positions. For the sake of better visibility we did not display all costs in this figure.

of the platforms (such as their positions, whether they are docked, etc.) to the centralized coordination system. We use the sensor measurements of the robots to build a grid map distinguishing free, occupied, and unexplored areas. Based on this map, we extract relevant locations in the environment, for example frontier cells [23] or locations for deployment. We then use this information to generate a problem description in the *Planning Domain Definition Language* (PDDL) [8], which serves as input for the temporal planner. In conjunction with a regular path planner such as A*, the planner computes action sequences for the robots that are send to the individual vehicles. The loop depicted in Fig. 2 is constantly executed. Whenever new information about the environment arrives, e.g., new sensor data is perceived or the robots moved, we replan.

B. Target Locations and Travel Cost

In this work, we model the fact that different robots may have different navigation capabilities and that certain areas of the environment can only be explored by the rovers and others only by carriers. We furthermore assume that the robots are able to determine based on their sensor observations which areas are traversable by which robot, for example based on techniques developed in our previous work [21].

To identify potential target locations, a set of exploration targets T is generated from the partially explored grid map. In addition to this, a set of meeting points M is determined. These meeting points are situated between those parts of the environment that can only be traversed by the rovers and the parts that can only be traversed by the carriers. They are used for deployment and retrieval of the rovers (see Fig. 3 for an illustration). To determine the meeting points a frontier extraction algorithm is used.

There are two basic types of actions a carrier can perform: exploring a target or visiting a meeting point to deploy or retrieve a rover (see Fig. 1). While deployment and retrieval are assumed to have constant cost c_{dep}, the cost of traveling between two locations in the environment is defined as the estimated path cost (i.e., travel time). This cost depends on the path length as well as on the traversability constraints and travel speed of the corresponding robot. Let *type* be a robot type (here: carrier or rover), x a location in the environment and $t \in \{T \cup M\}$ a target. We define the cost for reaching t as:

$$C_{type}(x, t) \tag{1}$$

$$= \begin{cases} \text{est. path cost}(x, t) \text{ , } & \text{if robots of type } \textit{type} \\ & \text{can reach } t \text{ from } x \\ \infty & \text{, otherwise.} \end{cases}$$

Finally, the exploration task is assumed to be completed as soon as the set of exploration targets T is empty.

C. Formulating the Exploration Problem as a Temporal Planning Problem

A wide range of problem types can be modeled as a general planning problem, ranging from transportation problems and single-player games to combinatorial problems. In recent years, the *Planning Problem Definition Language* (PDDL) [8] has been established as the prevalent planning language. In this paper we use PDDL/M [4], an extension to PDDL allowing for the definition of external modules.

The problem of multi-robot exploration with marsupial robots is a temporal planning problem. The reasons for this are mainly the facts that the actions of the individual robots have an individual duration and that the problem is inherently highly parallel. Especially for the efficient retrieval of rovers, the time the individual robots need to carry out their actions needs to be considered.

To generate a PDDL task description, we need to define (i) the objects involved in the planning process, (ii) the predicates that define the state of the planner, (iii) actions that change the predicates, and (iv) start and goal states.

First, we define what type of objects are involved. In the exploration scenario, possible objects are robots that can be either rovers or carriers and locations that can be meeting points or exploration targets. Fig. 4 (left) illustrates the corresponding PDDL statements. Second, we specify the predicates that define the internal states. The major predicates we use to describe the exploration problem are

```
(at ?r - robot ?x - location)
```

which describes if the robot r is at position x.

```
(on ?e - rover ?c - carrier)
```

is used to determine if a rover e is docked to a carrier c. For each target $t \in T$, we also define if it has been explored

```
(explored ?t - target).
```

Additionally, we use a numeric fluent

```
(num_docked ?c - carrier)
```

that contains the number of rovers that are docked to a carrier c.

Third, the actions that change the predicates have to be provided. We need four actions in our setting, namely *dock*, *undock*, *move*, and *explore*. The actions *dock* and *undock* require that the carrier and the rover are at the same meeting point (see `at` predicate). For docking, the number of docked rovers has to be lower than the carrier's capacity and the action changes a rover's state from being at a meeting point to being on a carrier (see `on` predicate).

The other two actions *move* and *explore* model the possible motions of the robots. The *move* action moves a robot to a meeting point for deployment or retrieval while the *explore* action moves the robot to a target and explores it. For the *move* and *explore* actions, we utilize the module interface [4] of our planning approach to define the duration. Instead of specifying a constant duration or a fixed formula, we call an external module that determines the duration and the actual cost for taking the action. In our setting, the external module is realized by a traditional A^* path planner that plans the optimal trajectory of the robot to the given target location based on the current occupancy grid map constructed by the robots so far. Fig. 4 (middle) depicts the PDDL statements that describe the action *explore*. The term `[pathCost ?r ?s ?g]` represents the call of the external module.

Finally, the initial state of the current planning procedure and the goal state need to be specified. For the situation depicted in Fig. 1, this is exemplified in Fig. 4 (right).

D. The TFD/M planning system

The PDDL description forms the input to the TFD/M planner. Based on this description, the temporal planner computes concurrent action sequences for the robots. TFD/M is a domain-independent progression search planner built on top of the planning system *Fast Downward* [12]. It extends the original system to support durative actions, numeric and object fluents, and external modules.

TFD/M solves a planning problem in three phases: First, the PDDL planning task is translated from its binary encoding into a more concise representation using finite-domain variables. This enables the use of heuristics employing hierarchical dependencies between state variables which leads to an increased search performance. In the second step, efficient internal data structures for the heuristic and the search component are generated. The most important ones are domain transition graphs for each variable that encode how state variables can change their values and the causal graph that represents the hierarchical dependencies between different state variables. Finally, a best-first progression search is performed, guided by a numeric temporal variant of the context-enhanced additive heuristic.

In contrast to many other temporal planning systems, TFD/M does not split the search in an action selection and a scheduling phase but searches directly in the space of time-stamped states. This typically leads to plans of significantly higher quality [7]. Note, however, that due to the inadmissibility of the heuristic evaluation function, the first plan that is generated is not necessarily optimal.

TFD/M does not terminate after a solution was generated, but is implemented as an anytime algorithm. By producing a potentially non-optimal solution quickly, the search space can be pruned to those time-stamped states which can potentially be extended to solutions of a lower overall execution time than the best solution found so far. If all states in the resulting state space are expanded, the produced solution is guaranteed to be optimal.

TFD/M features semantic attachments that are a means of evaluating components of the planning task externally. TFD/M implements this as a module interface for predicates, numerical effects, and durations. In our case, durations of actions are specified as a module call in the planning task description. When expanding actions in the search phase the planner detects these module calls and executes the dynamic library associated with the module call which in turn will retrieve the real costs computed by the A* path planner.

For further details on TFD/M, we refer the reader to our previous work [4, 7].

IV. EVALUATION

The approach described above has been implemented and evaluated thoroughly using a multi-robot simulation system. The experiments have been designed to show that explicitly planning symbolic action sequences leads to a significantly more efficient coordination approach than using a heuristic extension of previous coordination approaches.

A. Simulation System

To quantitatively evaluate our coordination approach, we developed a simulation system that is able to simulate large teams of marsupial robots. In our current system, we also simulate laser range sensors. Sensor and odometry noise are not considered since we focus on the coordination aspects of the problems. The environment is modeled by a grid map with additional traversability information. The maximum sensor range and traveling speed of carriers and rovers can be specified.

B. Baseline Approach

The baseline approach that we compare our algorithm against is a heuristic extension of a method that assigns robots to target locations based on cost estimates [19]. The carriers are assigned to exploration targets independent of whether they are accessible to them or not. The selection is solely based on the estimated travel cost. The rovers are then deployed heuristically: Whenever a carrier is assigned to a target that it cannot explore itself it will move to the nearest connecting meeting point and deploy a rover there. This rover will then explore the targets reachable from the meeting point. As soon as it has finished exploring them, it will return to the meeting point. As mentioned above, we assume a limited number of rovers per carrier. As soon as a carrier needs to deploy a rover but has none available, our heuristic requires the carrier to first retrieve a rover.

C. Comparison of Baseline Solution With Our Approach

We evaluated robot teams of varying sizes and different environments have been used in the simulation. In the simulation, carrier robots are twice as fast as rovers and their maximum sensor range is also twice as far.

Two of the environments we used to evaluate our approach can be seen in Fig. 5. The office environment resembles a typical office building with two corridors and a number of rooms. Some of the rooms can only be explored by

```
(:types
  robot
  carrier rover - robot
  location
  target meeting - location )
(:predicates
  (at ?r - robot ?x - location)
  (on ?e - rover ?c - carrier)
  (explored ?t - target)
  (can_explore ?r - robot ?t - target) )
```

```
(:durative-action explore
  :parameters (?r - robot
     ?s - location ?g - target)
  :duration (= ?duration
     [pathCost ?r ?s ?g])
  :condition (and (at start (at ?r ?s))
     (at start (not (explored ?g)))
     (at start (can_explore ?r ?g)) ... )
  :effect
    (and
      (at start (not (at ?r ?s)))
      (at end (at ?r ?g))
      (at start (explored ?g))
      ... ))
```

```
(:init
  (at robot0 p)
  (on robot1 robot0)
  (can_explore robot0 t1)
  (can_explore robot1 t2)
  (can_explore robot0 t3)
)
(:goal (and
  (explored t1)
  (explored t2)
  (explored t3)
))
```

Fig. 4. Examples for PDDL definitions. Left: definition of the required types and predicates. Middle: definition of the explore action. Right: Example that shows how to specify the current state of the world for the TFD/M planner (see scene shown in Fig. 1).

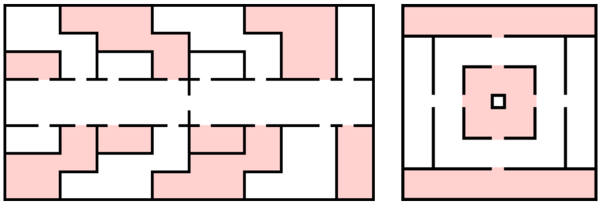

Fig. 5. Our simulated experiments: office (left) and maze (right). White areas can only be traversed by carriers while red (dark) areas can only be explored by rovers.

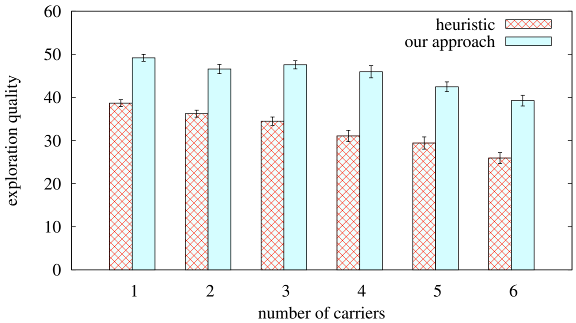

Fig. 6. Exploration quality in the office environment over 30 runs using two rovers per carrier. The higher the value, the better the coordination. The error bars indicate the 95% confidence intervals. Note that similar results were obtained for the maze environment.

rovers. The maze environment features a central area that can only be explored by rovers but in contrast to the office environment has multiple meeting points that can be used for deployment.

In both environments, we simulated 30 exploration runs using our approach and the baseline method with random initial robot positions. Exploration targets were determined using the frontiers approach and neighboring exploration targets were clustered using visibility constraints similar to the approach proposed by Burgard *at al.* [2].

An overview of the results obtained in these environments is given in Fig. 7. It can be seen that our approach explores the environment significantly faster than the baseline method in all configurations. It can also be seen that using more than three carriers improves the overall exploration time only marginally. The number of rovers for which this effect occurs clearly depends on the structure of the environment and the number of areas that can be explored by rovers only.

As a further benchmark, we computed the exploration quality according to [24] as

$$\mathcal{Q} = \frac{1}{A} \sum_{i=1}^{n} d_i, \tag{2}$$

where A is the total area of the environment and d_i denotes the distance traveled by robot i. This measure can intuitively be understood as the area each robot explores per movement.

The results in Fig. 6 show that our approach reaches a significantly higher exploration quality. Especially larger teams of robots are coordinated more efficiently, so that unnecessary movement is avoided.

V. DISCUSSION

The experimental results demonstrate that our approach can effectively coordinate large teams of robots and significantly outperforms a handcrafted strategy. In addition to that, our planning framework adds a substantial degree of flexibility to our system. For example, additional constraints such as power constraints for individual robots can be specified by adding adequate predicates to the problem description. Furthermore, other temporal actions such as recharging batteries or deploying sensor nodes can be integrated in a straightforward way.

A. *Limitations of the Approach*

The planning system described in this paper generates sequences of actions for the robots to explore the environment given the current knowledge about the world. While the robots move, their state changes and new information about the environment may be perceived. Therefore, we execute the planning cycle (see Fig. 2) in a continuous loop and use the solution the anytime planner reports. If more than one solution is found we set the timeout to 30 s.

We analyzed our approach with up to 24 robots (6 carriers plus 18 rovers). However, for significantly larger teams, the planning problem becomes large so that the solution reported by the anytime planner after 30 s may be sub-optimal.

VI. CONCLUSION

In this paper, we presented a novel approach to coordinate autonomous exploration with marsupial robots. Our approach combines traditional approaches for homogeneous teams that coordinate rovers by solving an assignment problem that maximizes a given evaluation function with a temporal planner that explicitly deals with the deployment and

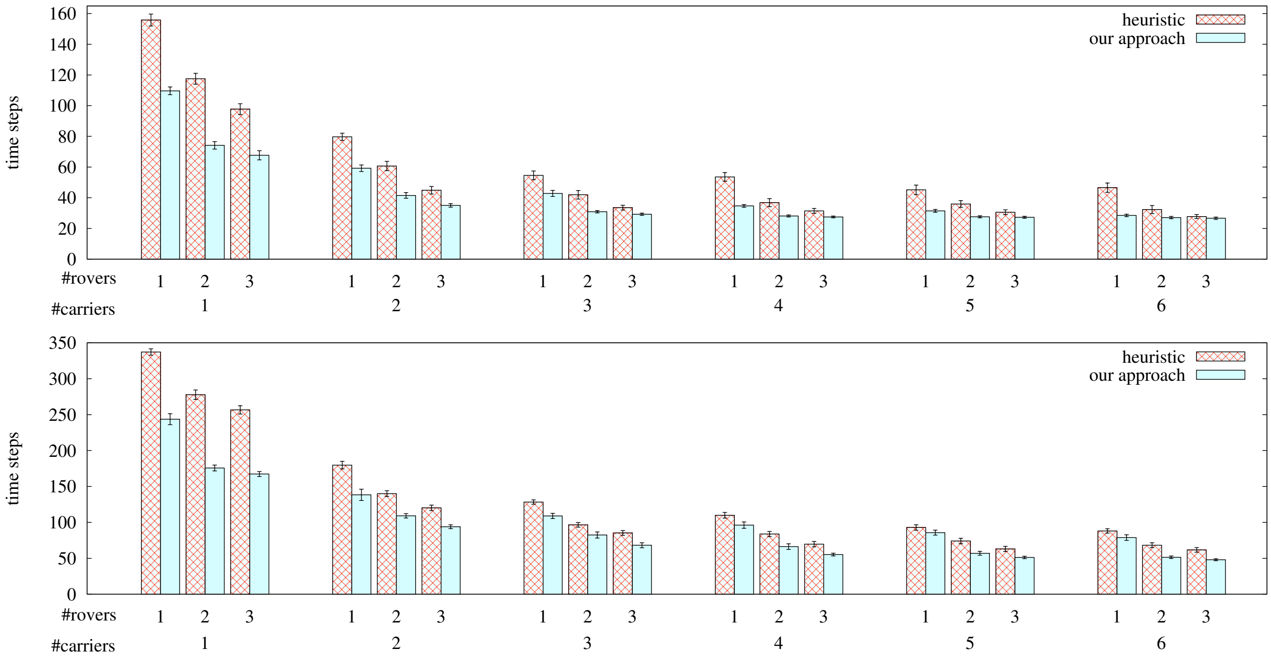

Fig. 7. Exploration time obtained with our approach compared to the heuristic in the maze environment (top) and in the office environment (bottom) for varying team sizes (number of carriers and number of rovers per carrier). The error bars indicate the 95% confidence intervals.

retrieval of small rovers. Our approach has been implemented and thoroughly tested in extensive simulation runs. The experimental results demonstrate that our approach can effectively coordinate large teams of robots and significantly outperforms a handcrafted strategy.

References

[1] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt. Robot exploration with combinatorial auctions. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1957–1962, 2003.

[2] W. Burgard, M. Moors, C. Stachniss, and F. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376–378, 2005.

[3] F. Dellaert, T. Balch, M. Kaess, R. Ravichandran, F. Alegre, M. Berhault, R. McGuire, E. Merrill, L. Moshkina, and D. Walker. The Georgia Tech yellow jackets: A marsupial team for urban search and rescue. In *AAAI Mobile Robot Competition Workshop*, 2002.

[4] C. Dornhege, P. Eyerich, T. Keller, S. Trüg, M. Brenner, and B. Nebel. Semantic attachments for domain-independent planning systems. In *Proc. of the Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pages 114–121, 2009.

[5] A. Drenner and N. Papanikolopoulos. A Framework for Large-Scale Multi-Robot Teams. *Modeling and Control of Complex Systems*, page 297, 2007.

[6] European Space Agency (ESA). ESA's lunar robotics challenge website. http://www.esa.int/esaCP/SEMGAASHKHF_index_0.html, 2008.

[7] P. Eyerich, R. Mattmüller, and G. Röger. Using the context-enhanced additive heuristic for temporal and numeric planning. In *Proc. of the Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pages 130–137, 2009.

[8] M. Fox and D. Long. PDDL2.1: an extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research (JAIR)*, 20(1):61–124, 2003.

[9] A. Gerevini, A. Saetti, and I. Serina. An approach to efficient planning with numerical fluents and multi-criteria plan quality. *Artificial Intelligence.*, 172(8-9):899–944, 2008.

[10] R. Grabowski, L.E. Navarro-Serment, C.J.J. Paredis, and P.K. Khosla. Heterogeneous teams of modular robots for mapping and exploration. *Autonomous Robots*, 8(3):293–308, 2000.

[11] M. Helmert. Decidability and undecidability results for planning with numerical state variables. In *Proc. of the Int. Conf. on Artificial Intelligence Planning and Scheduling*, pages 44–53, 2002.

[12] M. Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

[13] J. Hoffmann and B. Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.

[14] A. Howard, M.J. Matarić, and S. Sukhatme. An incremental deployment algorithm for mobile robot teams. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.

[15] E. Kadioglu and N. Papanikolopoulos. A method for transporting a team of miniature robots. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2297–2302, 2003.

[16] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. A practical, decision-theoretic approach to multi-robot mapping and exploration. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3232–3238, 2003.

[17] R.R. Murphy, M. Ausmus, M. Bugajska, T. Ellis, T. Johnson, N. Kelley, J. Kiefer, and L. Pollock. Marsupial-like mobile robot societies. In *Proc. of the Annual Conf. on Autonomous Agents*, page 365, 1999.

[18] K. Singh and K. Fujimura. Map making by cooperating mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 254–259, 1993.

[19] C. Stachniss. *Robotic Mapping and Exploration*, volume 55 of *STAR Springer tracts in advanced robotics*. Springer, 2009.

[20] C. Stachniss, O. Martinez Mozos, and W. Burgard. Efficient exploration of unknown indoor environments using a team of mobile robots. *Annals of Mathematics and Artificial Intelligence*, 52:205ff, 2009.

[21] K.M. Wurm, R. Kümmerle, C. Stachniss, and W. Burgard. Improving robot navigation in structured outdoor environments by identifying vegetation from laser data. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[22] K.M. Wurm, C. Stachniss, and W. Burgard. Coordinated multi-robot exploration using a segmentation of the environment. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008.

[23] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proc. of the Int. Conf. on Autonomous Agents*, pages 47–53, 1998.

[24] R. Zlot, A.T. Stenz, M.B. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2002.