A Framework for Multi-Robot Task and Motion Planning

Antonio Iannone and Antony Thomas and Fulvio Mastrogiovanni

***Abstract*—We present an approach for multi-robot integrated task and motion planning that allocates tasks to robots. Specifically, we focus on transportation-like tasks where the objective of a task correspond to transporting objects to a desired location. We demonstrate the performance under varying robot number and transportation tasks.**

***Index Terms*—Task planning, motion and path planning, multi-robot planning**

I. Introduction

Consider a robot performing a transportation-like task where the robot picks different objects from their locations and move them to their respective delivery locations while minimizing a cost metric. To this end, the robot has to reason about different objects and their properties— pick-up and delivery locations, to find an optimal solution. In such scenarios, a task plan (sequence of actions from a start state to a desired goal state) alone is not sufficient as appropriate motions to execute the sequence of actions need to synthesized. For example, consider a task-level action that takes the robot from one place to another. To achieve this task the robot needs to find a collision free motion plan. This presents the need for Integrated Task and Motion Planning (TMP), an area that has received much research interest [1]–[4].

Yet, most approaches focus on single-robot TMP systems. As such, they cannot be naturally extended to incorporate multi-robot semantics— task allocation or collision avoidance, and would have to treat the multi-robot system as a combined system, which becomes intractable as the number of robots increases. A multi-robot TMP approach in the context of transportation tasks is presented in [5]. They introduce Interaction Templates (IT) that enable handing over payloads from one robot to another. However, this method do not take into account the robot availability and assumes that there is always a robot available for such an handover. This can be catastrophic when many tasks are considered at a time. This assumption is relaxed in the work of Motes *et al.* [6]. The approach in [6] additionally considers task decomposition. A distributed approach is presented in [7]. They define task-level actions for a pair of robots and hence sub-optimal solutions are returned for an odd number of tasks.

We present an approach for multi-robot TMP wherein the objective of a task correspond to transporting objects to physical locations in the environment. The method allocates task to different robots, which may be accomplished simultaneously and the plan returned is task-level optimal. This means that the task plan cost returned by our approach is lower than any of the other possible task plan cost. Finally, unlike other multi-robot TMP approaches, by redefining task as locations to visit, our approach can be employed in multi-agent path finding.

The authors are with the Department of Informatics, Bioengineering, Robotics, and Systems Engineering, University of Genoa, Via All'Opera Pia 13, 16145 Genoa, Italy. email:anto.iannonemail@gmail.com, antony.thomas@dibris.unige.it, fulvio.mastrogiovanni@unige.it

II. Method

We present a multi-robot TMP approach that focus on transportation-like tasks and returns a task-level optimal plan. For actions that take the robot from one place to another, a motion plan is found that minimizes the robot path length.

TMP essentially involves combining discrete or high-level reasoning with continuous or low-level decision-making to facilitate efficient interaction between the two layers. TMP synthesizes a plan from a start state to a goal state by a concurrent or interleaved set of discrete actions and continuous collision-free motions. The de facto standard syntax for task planning is the *Planning Domain Definition Language* (PDDL) [8] and we resort to the same for task planning. For motion planning, any off-the-shelf motion planner may be employed. To achieve the interaction the discrete and continuous layers, it is imperative to obtain a mapping between. Specifically, this requires a mapping between (1) the discrete task state and the corresponding configuration (of robot and other objects) in the configuration space, and (2) discrete action and the corresponding motion plan. To realize this mapping we use semantic attachments that associate algorithms to functions and predicate symbols via external procedures. Thus, whenever the task planner needs to expand an action that require appropriate robot motions to be computed, the motion planner is called via the semantic attachments. The motion planner computes a plan respecting different constraints such as robot-object collision avoidance, robot-robot collision avoidance and returns the minimum path length cost to the task planner. To achieve multi-robot collision avoidance, each robot communicates it path to the robot. A fragment of the PDDL domain is shown in Fig. 1.

Proving task-level optimality is quite straightforward. Let the optimal plan π^* have a cost c^*. Suppose that there exists a plan π with cost c such that $c < c^*$. Let π and π^* have the same sequence of actions. The action costs are evaluated by the motion planner and since we use a sampling based motion planner, the motion cost returned is the optimal for each action. Thus, this is a contradiction and $c^* \leq c$. If π and π^* have different actions, the task planner ensures that the returned plan is optimal, and therefore $c^* \leq c$.

By redefining task as locations to visit, our approach can be employed in multi-agent path finding where each agent has

```
(:durative-action goto_region
 :parameters (?v - robot ?from ?to - region)
 :duration (= ?duration 100)
 :condition (and (at start (robot_in ?v ?from)))
 :effect (and (at start (not (robot_in ?v ?from)))
 (at start (increase (triggered ?from ?to ?v) 1))
 (at end (robot_in ?v ?to)) (at end (assign (triggered ?from ?to ?v) 0))
 (at end (visited ?to ?v)) (at end (increase (act-cost) (dummy))))
)

(:durative-action pick_up
 :parameters (?v - robot ?at - region ?o - object)
 :duration (= ?duration 10)
 :condition(and (at start (= (empty ?v) 0))
 (at start (robot_in ?v ?at)) (at start(= (holding ?v ?o) 0))
 (at start (= (object_in ?o ?at) 1)) (over all (robot_in ?v ?at)))
 :effect (and (at end (assign (empty ?v) 1)) (at end (assign (object_in ?o ?at) 0))
 (at end(assign (holding ?v ?o) 1)))
)

(:durative-action put_down
 :parameters (?v - robot ?at - region ?o - object)
 :duration (= ?duration 10)
 :condition (and (at start (robot_in ?v ?at)) (at start (= (empty ?v) 1))
 (at start (= (object_in ?o ?at) 0))
 (over all (robot_in ?v ?at)) (at start(= (holding ?v ?o) 1)) )
 :effect(and (at end (assign (empty ?v) 0)) (at end (assign (object_in ?o ?at) 1))
 (at end(assign (holding ?v ?o) 0)) )
)
```

Fig. 1: A fragment of the PDDL domain. The PDDL keyword *increase* is overloaded to refer to an encapsulated object and invokes the motion planner if the PDDL action to be expanded has an effect with *increase*.

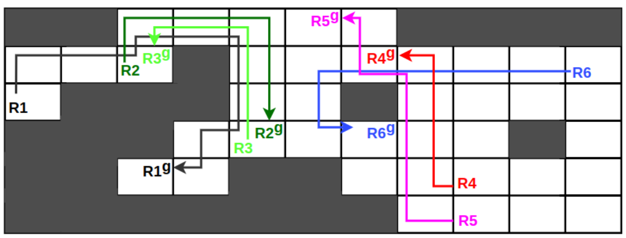

Fig. 2: Our TMP approach reduced to a multi-agent path finding problem. Robots $R1, \ldots, R6$ and their respective path to goal are shown.

a unique start state and a unique goal state. An example with six agents is shown in Fig. 2. At each instant, an agent can perform a move action or a wait action to stay idle at its current location. This collision avoidance strategy is implemented by the motion planner when called upon (semantic attachment). Note that for this variant of the problem only the *goto_region* action of PDDL is required.

III. Results

We evaluate the performance of our approach using varying number of robots and tasks. Fig. 3 shows a representative example of transportation tasks. Robots are allocated transportation tasks from Ti to goal G. Table I shows different statistics for varying robot number and tasks. For a fixed number of robots, the planning time increases linearly with increasing number of tasks. Given a fixed set of tasks, the planning time grows steadily with rise in robot numbers since the number of conflicts with respect to each robot increases.

IV. Conclusion

This paper presents an approach for multi-robot TMP that focus on transportation-like tasks— the objective of a task correspond to transporting objects to physical locations in the environment. The method allocates tasks to robots ensuring task-level optimality. Currently, the tasks have a single goal,

Robots	#Transportation tasks	Average planning time (s)
6	4	7.22
6	6	8.83
6	8	10.01
6	10	11.06
6	12	13.14
2	8	2.51
4	8	5.65
6	8	10.28
8	8	16.78
10	8	25.05

TABLE I: Experiment results for varying robot number and transportation tasks. The tasks are randomly selected and the average planning time is reported.

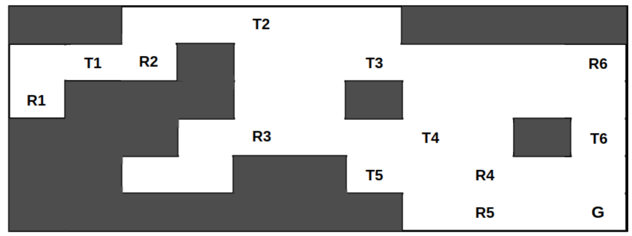

Fig. 3: Robots $R1, \ldots, R6$ have to carry out 6 transportation tasks $T1, \ldots, T6$. The planner allocates tasks to each robot such that the overall path length is minimized as the robots reach the goal G.

that is, each transportation task ends at the same goal. Also, the current framework do not support interaction between different robots— for example, a scenario where robots can transfer payloads. Immediate future work involves addressing these two limitations and evaluating the approach with other challenging environments.

References

[1] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32(9-10):1194–1227, 2013.

[2] Siddharth Srivastava, Eugene Fang, Lorenzo Riano, Rohan Chitnis, Stuart Russell, and Pieter Abbeel. Combined task and motion planning through an extensible planner-independent interface layer. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 639–646. IEEE, 2014.

[3] Neil T. Dantam, Zachary K. Kingston, Swarat Chaudhuri, and Lydia E. Kavraki. Incremental task and motion planning: A constraint-based approach. In *Robotics: Science and Systems*, 2016.

[4] Antony Thomas, Fulvio Mastrogiovanni, and Marco Baglietto. MPTP: Motion-planning-aware task planning for navigation in belief space. *Robotics and Autonomous Systems*, 141:103786, 2021.

[5] James Motes, Read Sandström, Will Adams, Tobi Ogunyale, Shawna Thomas, and Nancy M Amato. Interaction templates for multi-robot systems. *IEEE Robotics and Automation Letters*, 4(3):2926–2933, 2019.

[6] James Motes, Read Sandström, Hannah Lee, Shawna Thomas, and Nancy M Amato. Multi-robot task and motion planning with subtask dependencies. *IEEE Robotics and Automation Letters*, 5(2):3338–3345, 2020.

[7] Antony Thomas, Fulvio Mastrogiovanni, and Marco Baglietto. Towards Multi-Robot Task-Motion Planning for Navigation in Belief Space. In *European Starting AI Researchers' Symposium*. CEUR, 2020.

[8] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. PDDL- The Planning Domain Definition Language. In *AIPS-98 Planning Competition Committee*, 1998.