Iterative AND/OR Graphs for Task-Motion Planning

Antony Thomas, Hossein Karami and Fulvio Mastrogiovanni

***Abstract*—We present an approach for Task-Motion Planning (TMP) that compactly encodes the task-level abstractions within an AND/OR graph growing online. We consider a cluttered table-top scenario where a target object needs to be retrieved. Such scenarios are challenging, since the number of object re-arrangements to achieve the target grasp cannot be determined before-hand. However, conventional AND/OR graph based planning requires that the number of re-arrangements are known ahead of time. To address this challenge we propose an AND/OR graph iteratively growing online until the target is retrieved. This iterative graph facilitates faster computations with respect to traditional task planners. We validate our approach using a Baxter robot both in the real-world and in simulation.**

***Index Terms*—Task-Motion Planning, AND/OR graphs, Push-Grasping**

I. Introduction

Task-Motion Planning (TMP) combines discrete decision making with continuous geometric reasoning, and has been an active research area among the Robotics and the Artificial Intelligence community [1], [6], [9], [13], [15]. Task planning finds a discrete sequence of actions from an initial state to a goal state, whereas motion planning finds a collision-free path from an initial configuration to a specific goal configuration. Real-world scenarios often present such an interaction between high-level task actions and the low-level geometric motions to achieve the respective tasks.

Central to TMP is the interaction between the task planning and motion planning layers. The *de facto* standard used for task planning by the planning community is the *Planning Domain Definition Language* (PDDL) [11] and its variations, and most approaches resort to the same for task planning. For motion planning several off-the-shelf planners are available [14]. Currently there exist different methods for interaction between the task and motion layers. Semantic attachments [4], [5], [15] associate motion planning algorithms to functions and predicate symbols in PDDL via external procedures. In [13] a planner-independent interface layer performs this interaction, while in [1] abstraction and refinement are used to implement it. Caelan *et al.* [7] use streams within PDDL enabling procedures for sampling values of continuous variables. Apart from the work in [7], in most other approaches the robot configuration, grasp poses, etc., need to be pre-computed before-hand thus obtaining a finite motion space. The need for such a pre-discretization is relaxed in [6], [8], [16]. However, these approaches are domain specific in nature. Though off-the-shelf PDDL planners are available, one needs additional expertise to incorporate these task-motion interactions that comply with the state-space search of the planner. Moreover, integrating multi-agent (the two Baxter arms in our case) capabilities remains an added challenge since the interaction layer need to handle multi-agent coordination.

In this paper, we address these challenges by encoding the TMP problem efficiently and compactly within an AND/OR graph that grows iteratively till a goal state is reached. Specifically, we consider a cluttered table-top scenario wherein a target object is to be retrieved from clutter. Objects that hinder the target object grasp are either picked and placed or pushed to a different location. In general, an AND/OR graph needs to be constructed *offline*. However, since the number of object re-arrangements are scenario-dependent and not known beforehand, such an offline construction is impossible. We thus propose an iterative AND/OR graph that grows *online* until the target is retrieved.

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:antony.thomas@dibris.unige.it, hossein.karami@edu.unige.it, fulvio.mastrogiovanni@unige.it

II. Approach

We present an integrated task-motion planning framework for retrieving a goal object by re-arranging (pick and place or push) the objects hindering the grasps.

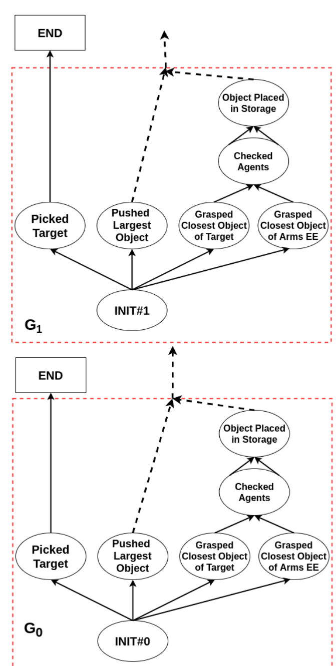


Fig. 1: Two AND/OR graphs growing online.

The AND/OR graph representation provides a framework for the planning and scheduling of task sequences [12], with a suitable motion planner used to check the feasibility of such tasks [10]. A background on the use of AND/OR graphs for human-robot collaboration, and a related example can be found in [2]. Moreover, an AND/OR graph inherently requires fewer nodes than the corresponding complete state transition graph, reducing the search complexity of the AND/OR space. Yet, such a representation requires all possible object re-arrangements to be computed *offline*. For the scenario considered here, this representation thus seems incompatible as we do not know before-hand the number of objects to be re-arranged. Moreover, the AND/OR graph is not compliant to failures in motion execution nor grasping arising due to actuation and other hardware failures.

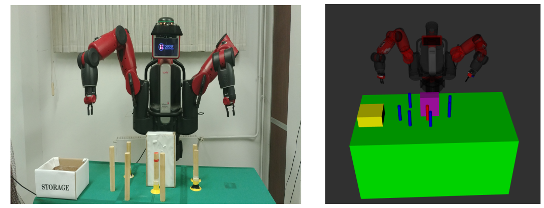
We recall here that the nodes of an AND/OR graph represent a high-level state and the hyper-arcs represent an action or a sequence of actions achieving the desired states. We note that by defining a suitable abstraction for our clutter scenario, the nodes and hyper-arcs of the graph remain the same irrespective of the number of object re-arrangements. Thus such an abstraction allows us to have a graph that grows/expands itself online until the target is retrieved. This is seen in Fig. 1, where an initial graph G_0 is expanded to a new graph G_1. The nodes in the graph represent different states. We note that in the two graphs apart from the nodes INIT#0 and INIT#1, all the other abstractions (nodes and hyper-arcs) remain the same. After each re-arrangement, the work-space configuration changes and the INIT# nodes represent such changed configuration, for example, the whole G_0 graph is embedded within INIT#1. This can be thought of as a virtual node augmented to the graph.

Given the initial work-space configuration, an AND/OR graph denoted by G_0 is first constructed. The graph G_0 represents the fact that if a feasible grasping trajectory exists then the Picked Target task is to be performed and the graph terminates (END). Otherwise, an object is to be identified to be either pushed (Pushed Largest Object) or grasped (Grasp Closest Object to Target or Grasp Closest Object to Arms EE), where we use EE for "end-effector". The grasped object is then placed in a storage are (Object Placed in Storage). Since the target object has not been grasped, G_0 expands to a new graph G_1 with updated work-space configuration via the node INIT#1. The process repeats iteratively until a graph G_n is expanded, which terminates when the target is retrieved.

We note here that G_i can also fail due to actuation errors. In such a scenario a new graph expands with the same work-space configuration thus making our approach robust to execution failures. To decide which objects to re-arrange, [3] search for objects that penetrate the planned trajectory space of each action by computing their volumes. In contrast, we first compute the motion plan by considering the object of interest, neglecting other objects in the work-space. For this planned trajectory, we then compute the collision probabilities with each object and the ones with collision probabilities greater than an ϵ bound are chosen as the objects to be re-arranged. Different heuristics are then employed to decide the order of removal.

III. Results

To demonstrate the capabilities of our approach, we consider a cluttered scenario in which a Baxter robot has to retrieve an object of interest. We restrict the pick-ups to only side grasps to make the scenario more challenging. To perceive objects in the environment, an RGB-D camera is mounted on the robot head. In Fig. 2(a), seven cylinders are placed on a table and the robot is required to pick up the object (target) with red tape; scene planning is visualized in simulation in Fig. 2(b). The white box has to be pushed since it occludes the target. However, few objects around the white box need to be removed before pushing. The experiment was conducted eight times and in each experiment the table configuration is randomly re-arranged. The overall planning and execution times for task and motion planning segments are shown in the first four rows of Table I. AND/OR time is the time for traversing the initial AND/OR graph G_0 and the online graph search time is the time for traversing the AND/OR graphs that grows online. An arc of an AND/OR graph may encode a sequence of actions and the graph search component of the task planner enumerates them to examine the feasibility. As seen in the last two rows of Table I, the graph was iteratively grown about 15 times on the average in each experiment with around 130 motion planning attempts. We note that the increased depth and motion planning attempts are due to (1) the failure of the motion planner in finding feasible plans, (2) grasping failures.



(a) (b)

Fig. 2: (a) Experiment setup. (b) Scene planning in simulation.

AND/OR Time	0.057±0.001 [s]
Online Graph Search Time	0.092±0.02 [s]
Motion Planner Time	6.2±0.7 [s]
Execution Time	119.85±4.57 [s]
Online Graph Depth Growth	15±3
Motion Plan Attempts	130±23

TABLE I: Quantitative results.

IV. Conclusion

We have presented a novel approach for TMP planning using iterative AND/OR graphs that grow online till the goal state is reached. Such a representation addresses the challenge of not knowing the number of object re-arrangements beforehand and allows for fast computation as compared to traditional task planners. We have validated our approach with a Baxter robot. Current work includes validation on other TMP benchmarks and extension to human-robot collaboration.

References

[1] 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.

[2] Kourosh Darvish, Francesco Wanderlingh, Barbara Bruno, Enrico Simetti, Fulvio Mastrogiovanni, and Giuseppe Casalino. Flexible human–robot cooperation models for assisted shop-floor tasks. *Mechatronics*, 51:97–114, 2018.

[3] Mehmet Dogar and Siddhartha Srinivasa. A framework for push-grasping in clutter. In *Robotics: Science and systems VII*, 2020.

[4] Christian Dornhege, Patrick Eyerich, Thomas Keller, Sebastian Trüg, Michael Brenner, and Bernhard Nebel. Semantic Attachments for Domain-Independent Planning Systems. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 114–121, Thessaloniki, Greece, September 2009.

[5] Christian Dornhege, Marc Gissler, Matthias Teschner, and Bernhard Nebel. Integrating symbolic and geometric planning for mobile manipulation. In *Safety, Security & Rescue Robotics (SSRR), IEEE International Workshop on*, pages 1–6. IEEE, 2009.

[6] Caelan Reed Garrett, Tomas Lozano-Perez, and Leslie Pack Kaelbling. FFRob: Leveraging symbolic planning for efficient task and motion planning. *The International Journal of Robotics Research*, 37(1):104–136, 2018.

[7] Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 440–448, 2020.

[8] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 1470–1477. IEEE, 2011.

[9] 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.

[10] Hossein Karami, Kourosh Darvish, and Fulvio Mastrogiovanni. A Task Allocation Approach for Human-Robot Collaboration in Product Defects Inspection Scenarios. In *29th IEEE International Conference on Robot and Human Interactive Communication*, 2020.

[11] 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.

[12] AC Sanderson, MA Peshkin, and LS Homen de Mello. Task planning for robotic manipulation in space applications. *IEEE transactions on aerospace and electronic systems*, 24(5):619–629, 1988.

[13] 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.

[14] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. https://ompl.kavrakilab.org.

[15] Antony Thomas, Fulvio Mastrogiovanni, and Marco Baglietto. Task-Motion Planning for Navigation in Belief Space. In *The International Symposium on Robotics Research*, 2019.

[16] Marc Toussaint. Logic-geometric programming: An optimization-based approach to combined task and motion planning. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.