Infeasibility Proofs in Task and Motion Planning

Sihui Li Neil T. Dantam

I. Introduction

Robot task and motion planning (TMP) searches for solutions in a discrete task planning space and continuous motion planning space at the same time. Because of this combination, one of the key challenges is the integration between the two spaces, and completeness properties of the motion planner impacts the ability to transfer information between these spaces.

A complete motion planner returns a plan or reports the non-existence of plans in finite time. However, this completeness property is hard to achieve. A weaker notion of completeness is *probabilistic completeness*, which guarantees the return of a plan given a long enough time if a plan exists. Many sampling-based motion planners are probabilistically complete [1], [2]. If a plan does not exist, a probabilistically complete motion planner would run forever or until a timeout. In task and motion planning, this nontermination causes a dilemma between giving the motion planner more time to solve or generating a new task plan, since we do not know exactly whether the plan exists or not.

We discuss the requirements and application of motion planning infeasibility proofs in TMP, and the potential to strengthen the completeness properties of TMP. Recent work [3], [4] addresses infeasibility proofs in pure motion planning. Infeasibility proof construction runs in parallel with a sampling-based motion planner, and the resulting algorithm is *asymptotically complete* [5], a stronger notion than probabilistic completeness which guarantees the return of a plan or infeasibility proof in the limit. Applying a motion planner with this stronger definition of completeness to TMP problems offers the potential to resolve the dilemma between more motion planning time vs. alternative task plans and further to achieve stronger completeness guarantees in TMP.

II. Related Work

In this section, we give a brief summarization of TMP works emphasizing the three types of integration between the task space and motion space [6], [7]: satisfaction-first, sequence-first, and unified.

For satisfaction-first TMP strategies [8]–[10], continuous space searches happen first to ensure a set of satisfying configurations is available then goes into task planning. Sequence-first TMP strategies [11]–[17] first create a task plan, then they generate a motion plan for each action within the task plan. Unified strategies [18]–[21] alternate between task planning and motion planning or combine the two into a multi-modal space to search for solutions. Infeasibility proofs could help in all three types of integration by removing invalid search branches in task planning. Some previous work provides probabilistic completeness guarantees [8], [12]. Others seek to find asymptotically optimal TMP solutions [16], [19], [22]. However, previous work rarely discusses infeasibility in TMP.

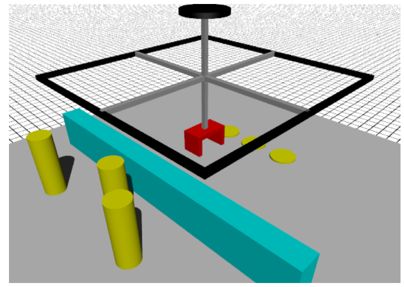

Fig. 1: Example task and motion planning problem where feasibility is a key issue. The gantry manipulator must move the three cylinders to the three stations. If the gripper is holding the cylinder, then there is not enough space above the block to pass. The task and motion plan must first remove the block. If instead the block is a fixed obstacle, then the TMP problem is infeasible.

III. Problem Definition

The inputs to the problem are: the task planning space as a discrete transition system; full continuous space information of the objects in the task space and the robot's description; the start state and the goal state.

IV. TMP with Infeasibility Proof

In this section, we outline how infeasibility proofs improve TMP's search results and completeness properties.

Figure 2 illustrates our strategy. We take the sequence first strategy similar to [12]. First, the task planner produces a sequence of discrete actions. Then, for each action in the action sequence, we try to generate a motion plan. With an asymptotically complete motion planner, the result is either a motion plan or an infeasibility proof. If an infeasibility proof is found, then this information transfers back to the task planner in the form of additional constraints in the search problem associated with the task plan. Eventually, if no task plan exists because of all the constraints added, we may conclude the TMP problem itself is infeasible.

Generally, there may be additional considerations that require further attention. One challenge involves the abstraction of infinitely many continuous states into a finite set of task planning states. Another issue is encoding or incrementally changing the configuration space when the task space experiences discrete state changes (such as when grasping an object).

Figure 1 gives an example of a TMP problem that would benefit from using infeasibility proof. In this scene, the

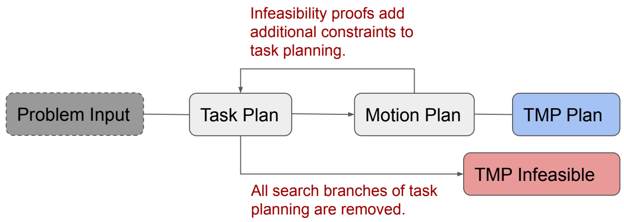

Fig. 2: Block diagram of a sequence first TMP strategies with motion planning infeasibility proof. The infeasibility information from the motion planning is encoded back to the task planning side.

robot has three prismatic joints. The X-axis and the Y-axis move horizontally within the black square region, and the Z-axis moves vertically. The goal is to move the three yellow cylinders to the three round yellow stations. There is a blue block in the middle.

A naïve task plan would be to move each of the cylinders one by one, that is *step 1: move*{*c1, s1*}, *move*{*c2, s2*}, *move*{*c3, s3*}. However, this task plan does not have a corresponding motion plan because when the robot is grasping the cylinder, it cannot pass through the opening on top of the blue block. In this case, the motion planner generates infeasibility proofs for the *move* action of each cylinder at the current steps. Transferring these infeasibility proofs to the SAT problem in task planning adds three constraints, that is *step 1: (not move*{*c1, s1*}*)), (not move*{*c2, s2*}*), (not move*{*c3, s3*}). Eventually, when all possible combinations of moving blocks to stations are constrained to be invalid, the task planner generates a new task plan, *step 1: move(b), step 2: move*{*c1, s1*}, *move*{*c2, s2*}, *move*{*c3, s3*}, which moves the blocks away first, then try to move the cylinders. Also, if the block is part of the non-movable obstacle, following similar steps, we cannot find a valid task plan, which means the TMP problem is infeasible.

It is also possible to use satisfaction-first strategies. Infeasibility proofs in configuration space create separations of free regions. Between separations, no motion plan exists. Using this information, we can generate samples for actions in the same region to guarantee the existence of motion plans.

V. Future Work

Infeasibility proofs offer the potential for asymptotic completeness guarantees in TMP. When infeasibility proofs in motion planning conclusively remove search branches in task planning, and when task planning exhausts all search directions and still cannot find a satisfying solution, it means no solution to the TMP problem exists. We can analyze the fixed points of the planning graph [23] to show that a goal is unreachable.

Currently, the infeasibility proof construction works well in 5-DoF and takes several minutes. The primary focus is to improve the overall runtime and scale the algorithm to higher dimensions since the practical use of infeasibility proof in TMP requires constant calls to the motion planning sub-routine. Incorporating infeasibility proofs in TMP also requires proper "mapping" between the continuous space and the discrete space, and open issues remain for feasibility determination when infinitely many configurations map to a discrete state. We need structured and generalized methods to encode configuration information to task space properly and to sample configurations given task-level constraints.

References

[1] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *T-RO*, vol. 12, no. 4, pp. 566–580, 1996.

[2] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *ICRA*, vol. 2, 2000, pp. 995–1001.

[3] S. Li and N. T. Dantam, "Learning proofs of motion planning infeasibility," in *RSS*, 2021.

[4] ——, "A sampling and learning framework to prove motion planning infeasibility," *IJRR*, 2023.

[5] ——, "Exponential convergence of infeasibility proofs for kinematic motion planning," in *WAFR*, 2022.

[6] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.

[7] N. T. Dantam, *Task and Motion Planning*. Springer Berlin Heidelberg, 2020.

[8] C. R. Garrett, T. Lozano-Perez, and L. P. Kaelbling, "Ffrob: Leveraging symbolic planning for efficient task and motion planning," *IJRR2018*, vol. 37, no. 1, pp. 104–136, 2018.

[9] A. Krontiris and K. E. Bekris, "Efficiently solving general rearrangement tasks: A fast extension primitive for an incremental sampling-based planner," in *ICRA*. IEEE, 2016, pp. 3924–3931.

[10] A. Akbari, J. Rosell, *et al.*, "Task planning using physics-based heuristics on manipulation actions," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2016, pp. 1–8.

[11] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, "Incremental task and motion planning: A constraint-based approach," in *RSS*, 2016.

[12] ——, "An incremental constraint-based framework for task and motion planning," *IJRR*, vol. 37, no. 10, pp. 1134–1151, 2018.

[13] T. Lozano-Pérez and L. P. Kaelbling, "A constraint-based method for solving sequential manipulation planning problems," in *IROS*, 2014, pp. 3684–3691.

[14] S.-Y. Lo, S. Zhang, and P. Stone, "Petlon: planning efficiently for task-level-optimal navigation," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 220–228.

[15] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Sampling-based methods for factored task and motion planning," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1796–1825, 2018.

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

[17] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 30, 2020, pp. 440–448.

[18] W. Thomason and R. A. Knepper, "A unified sampling-based approach to integrated task and motion planning," in *ISRR*, 2019.

[19] W. Thomason, M. P. Strub, and J. D. Gammell, "Task and motion informed trees (tmit*): Almost-surely asymptotically optimal integrated task and motion planning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 370–11 377, 2022.

[20] Z. Kingston, A. M. Wells, M. Moll, and L. E. Kavraki, "Informing multi-modal planning with synergistic discrete leads," in *ICRA*. IEEE, 2020, pp. 3199–3205.

[21] Y. Lee, W. Thomason, Z. Kingston, and L. E. Kavraki, "Object reconfiguration with simulation-derived feasible actions," *arXiv preprint arXiv:2302.14161*, 2023.

[22] W. Vega-Brown and N. Roy, "Asymptotically optimal planning under piecewise-analytic constraints," in *Algorithmic Foundations of Robotics XII*. Springer, 2020, pp. 528–543.

[23] S. J. Russell, *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.