Research Proposal

Towards Generalizable Robot Task Planning in Open Worlds with Uncertainty

Zhongxuan Li

Abstract

An enduring goal of AI and robotics has been to deploy robots in unstructured and interaction-rich environments. Robot task planning remains a crucial but challenging problem by the fact that a planner only has access to a description of robot skills, partially observable dynamic state-space and the goal to reach. From these it computes action sequences for arbitrary situations. This proposal reviews prevalent methods in the realm of robot task planning. We then outlines the authors' previous experience in (1) robot swarm control and (2) optical network planning with constraints, which summed up to our future research plan to the Ph.D. program.

Keywords: Task and Motion planning, Formal Methods, Neural Networks

1 Introduction

> "If you look at the field of robotics today, you can say robots have been in the deepest oceans, they've been to Mars, you know? They've been all these places, but they're just now starting to come into your living room. Your living room is the final frontier for robots."
>
> -Cynthia Breazeal

Today robots are already good at automating simple, repetitive tasks (e.g. grasping) that don't require much high-level intelligence. However, robots are still incapable of longer horizon and increasingly sophisticated tasks that require open-ended learning and planning. As an example, imagine a robot that is switched on in an open world (e.g. an unfamiliar building) without a map, and given the task of finding a particular object. We must pay attention to how variable, dynamic, observable, and predictable the environment is, and what the robot knows and perceives about it while acting. The factors we must consider includes the following.

* *Dynamics* of the environment. Notably that the robot must plan in discrete domains with very large state spaces (e.g. is-holding-water, is-cooked). Hence it is critical to use a functional representation to reveal states incrementally [1]. But a purely discrete model abstracts away continuous lower-level movement parameters (e.g. object positions, joint poses) that also need to be modeled.
* *Time and concurrency* while executing. Some actions requires critical time constraints which must be modeled explicitly for the purpose of deadlines and synchronizations [2]. These constraints imposed further complexity in solving the task planning problem.
* *Generalizability* of a planner. When a robot switch to an unfamiliar environment, how much portion of its pre-learned skills and policy can be generalized to the new world.
* *Observability* of the environment. It is seldom the case that the information of surrounding environment is permanently known to the robot. Some factor may become (partially) observable if specific actions are performed. The robot must keeps a reasonable belief about state variables [3].
* *Uncertainty* in knowledge and predictions. No agent is omniscient. It may or may not be able to reason about the uncertainty (e.g. a moving obstacle) regarding the current state of the world and the predicted future [4]. Furthermore, the low-level continuous movement parameters can sometimes intervene the high-level decisions in terms of collision avoidance and kinematic constraints, leading to task failures.

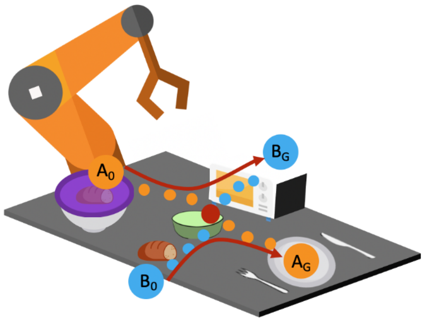

Figure 1: Example of a robot task planning problem: The arm needs to move the bread in the white bowl A_0 to the plate A_G using the green bowl, which also requires it to open the purple lid first. In addition, the arm needs to move a second bread in B_0 to the oven B_G using the green bowl which leads to conflicts with the previous task.

To brief conclude the above factors. Dynamics is about the intrinsic property of the environment, and how the planner defines the state-action space. Time, Concurrency and Generalizability are related to planner ability of orchestrating the action sequence. Observability and Uncertainty are about the noise involved during planning. In later context, we refer them collectively as uncertainties. Solving robot task planning with the above constraints could be hard, but is essential and critical to achieving long-term autonomy in the research of robotics [5]. The following questions need to be answered before a robot can be formally classified as a long-term autonomous agent.

* (**Q1**) How can the robot autonomously reason about the dynamic states of the world in certain forms of knowledge representations?
* (**Q2**) How can the robot automatically decompose its overall objective into simpler movement primitives and generate continuous motion parameters?
* (**Q3**) How can the robot handle uncertainties and react to various probabilistic contingencies?
* (**Q4**) How can the robot generalize its skills to different problem settings, in both object level and action level?

In the following sections, we lay out a research plan aimed at answering the aforementioned questions. The document is organized as follows: In section 2 we provide the fundamental background relevant for robot task planning. In section 3 we describe previous achievements and motivations, and in section 4 and 5 we discuss possible research directions and detailed timeline to implementation.

2 Related Prior Work

Defining a Planning Domain. Task planning requires a model of the world and descriptions of how state and actions that an agent interacts with the world. The language framework needs to be task-agnostic and domain independent. The representative symbolic languages includes Planning Domain Definition Language (PDDL) [6] and Answer Set Programming (ASP) [7]. A fundamental difference between them is on their (non-)monotonicity property, whereas the latter allows removal of previous achieved conclusions when new information is given [8]. For tasks with tight temporal constraints, Linear Temporal Logic (LTL) [9] reasons about the sequence of states that the planner might go through. However, all these symbolic languages encounter state explosion when the number of discrete objects and locations are large. To alleviate this problem, approaches in [10] uses abstraction of the state-space and plans in a multi-layer architecture, which is a still a pitfall by the fact that operating at a level of abstraction is not necessarily ideal for real-world tasks.

Generate Action and Motion. Provided with the state description, not only do we need to generate discrete state-transition actions achieve the overall goal, we also need to choose the appropriate continuous dynamic movement parameters. To address these concerns, two branches of work have emerged. To generate discrete actions, we can directly apply Breadth-First Search (BFS) or Depth-First Search (DFS) on the PDDL domain. Heuristic Search Planning (HSP) (e.g. Fast-Forward) achieves better time efficiency by solving relaxed versions of the task planning problem [11, 12]. Alternative search methods includes Planning Graphs [13] and Partial-Order Planning [14]. To generate continuous parameters, approaches can be classified into sampling-based methods and optimization based methods. The former includes Multi-Modal Motion Planning [15, 16], whereas the latter includes nonlinear programming [17, 18] and Continuous-Control Numerical Planning. Other approaches include [19] Gaussian movement primitives [20] and adaptive sampling for generating diverse motions trajectories. To best incorporate discrete task planning and continuous motion planning, Task and Motion Planning (TAMP) combines symbolic task planning with feasibility checking [21, 22]. The representative work is PDDLStream [23, 24] which has a hierarchical structure and treat motion samplers as black boxes.

React to Uncertainty. The physical nature of robot systems, as well as the uncertainty connected to robot behaviors (i.e. stochastic actions) and perception (i.e. partial observability) destroy many of the assumptions made by current methods for planning [25]. For planning in dynamic environment with moving obstacles, [26–28] replace sampling-based motion planer with optimization-based methods that jointly optimizes over all of the parameters and trajectories in a given abstract plan. Optimization methods can model the variables related to envrionmental uncertainty in an explicit way, thus well interpret the planning problem. But the problem formulation is normally highly non-convex. Mathematical techniques such as Sequential Quadratic Programming (SQP) can be used to accelerate the optimization process. For planning in occluded and even unknown environment, Partially Observable Markov Decision Processes (POMDP) [29, 30] present a set of possible states as belief, and updates the posterior estimation while executing tasks. Approaches include maximum likelihood observation [31] and other popular Bayesian-based methods [32–34]. However, these methods are still cumbersome today.

Life-long Generalization. Robots need the ability to be generalized to different domains (e.g. object shapes, dense clutter) and different behaviour sequences. To plan with different object semantics, deep learning model in [35] directly takes RGB-D input for estimating grasp affordance of various object shapes. For planning in clutter, Graph Neural Network (GNN) can be used in encoding the spatial relationship between objects [36]. In addition, non-symbolic methods in [37] and [38] construct a traversability graph which encode the reachability between grasping poses as edges. While it achieved higher success rate and shorter planning time, it has some strict underlying requirements such as objects' shape or their distance to each other. To generate versatile action sequences, it is unrealistic to employ only hard-coded symbolic planners. Multimodal imitation learning [39, 40] enables online acquiring of new skills during interactions with the outside world. Other learning based methods include Neural Task Programming [41] that generate action sequence end-to-end based on raw observations. Efforts have also been made on using learning-based methods to improve TAMP, including learning sampling guidance [42] and learning search guidance [43]. In general, machine learning methods can help with the process of acquiring models in nonideal domains as well as speeding computation. However, the theoretical impact of neural networks make it difficult for the policies to generalize outside of the training data.

Problem	Methods	Merits	Limitations
Programming Domain	PDDL	Domain independent	State explosion, need prior human knowledge
	ASP	Non-monotonicity	
	LTL	Strict temporal constraint	
Task Planning	BFS/DFS	Easy to implement	Probabilistic complete, not optimal
	Heuristic Search Planning	Faster	
	Graphplan	Faster	
	Partial-Order Planning	Avoid doing work that might have to be undone later	High computational power required
	Reachability Graph	Faster and higher success rate in clutter	Limited in specific task
	GNN Planning	Fit in certain topology	Need training, may overfit
	Learning search guidance	Faster and better optimality	Need training
Motion Planning	Sampling (PRM, RRT)	Faster	Need additional pruning, expensive collision checking
	Nonlinear programming	Can reach global Optima, model uncertainty with variables	Computational intensive
	Multi-Modal Motion Planning	Can handle complex tasks	May not converge
	Learning sampling guidance	Faster and better optimality	Need training
TAMP	Hierarchical planning	Escape from exponential explosion	Cannot consider all operation instantiations
	PDDLStream	Use off-the-shelf planners, balances exploration vs. exploitation	Cannot generalize to complex scenes with uncertainty
	POMDP	Handle uncertainty in perception	Computational Expensive
Robot Learning	Reinforcement Learning	Perform good in high-dimensional space	Trial and error, hard to converge
	Imitation Learning	Learn from expert planners	Hard to learn sub-goals and to generalize
	Neural Task Programming	End-to-end training	Intractable and not explainable

Table 1: Comparsion between different robot task (and motion) planning methods.

3 Achievements and Motivations

3.1 Achievements in Decision Making with Uncertainties

A major part of my job at Huawei is about decision making in large-scale optical transport networks. An exemplary work is route and wavelength planning in Wavelength Division Multiplexing (WDM) network [1], which is inherently similar to robot task planning. A lightpath is established between a source node and a destination node whenever a connection request arrives. There are multiple hard constraints that the planned lightpath must satisfy (e.g. wavelength continuity, disjoint protection path) while minimizing the total wavelength utilization and maximizing the number of success planned paths. A major source of uncertainty comes from rerouting. When one physical paths has fault, the path planner need to find another available route within seconds or miliseconds. Although there exist a bunch of graph search heuristic algorithms of solving this issue, they are not complete or only probabilistic complete. Hence heuristics may lead to task failures in rerouting. Furthermore heuristics often have poor adaptability to various topology of networks, making parameter tuning a tedious handcraft.

Our approach define the Route and Wavelength Assignment (RWA) problem in an Integer Linear Program (ILP). As optimization-based methods can model all variables in an explainable way, thus prevent the solution from trapping in local optima. But the non-convexity and NP-hard nature of the problem hinders the deployment of ILP in engineering practice. We therefore have proposed several techniques to accelerate the computational efficiency by apply relaxation on some strict constraints. Our approach is based on the art of Benders' cut, we first decompose the origin problem into three relaxation sub-problems[2] which can be quickly solved by meta-heuristics and branch-and-bound. A simplified illustration can be found in figure 3, the final version of this algorithm is transformed into product code in Huawei Optical Network Planner.

We have found this problem tantamount to robot task and motion planning in dynamic environment with strict constraints. Many scholars have made contribution on utilizing optimization techniques in TAMP [26,27].

[1] Nine Key Challenges Facing Optical Communications in the Next 10 Years, Huawei

[2] i.e. pathfinding, optical multiplex section and lambda assignment, which are in essence graph search and boolean satisfiability problems.

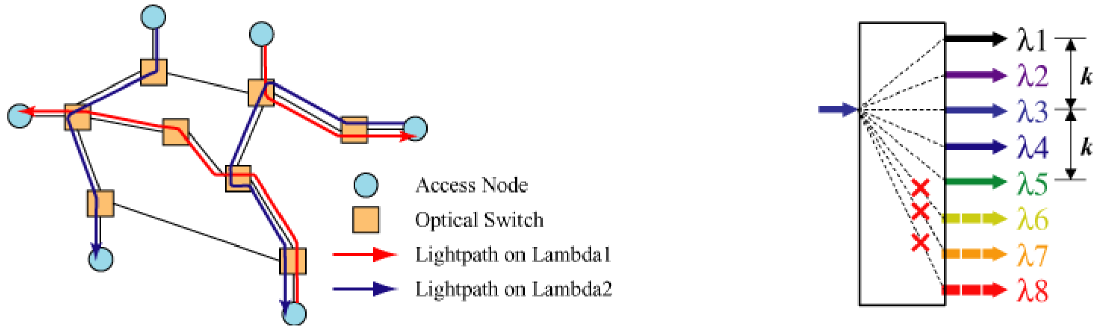

Figure 2: Left: An Example of RAW problem. The size of real world network often have hundreds of nodes. Right: it is not reasonable that all nodes have all wavelengths available, but the total lambda number per node could range to 96, result in large search space.

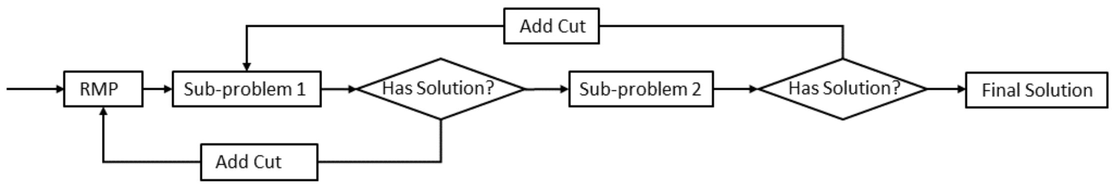

Figure 3: An simplified illustration of our approach. The origin problem is first relaxed to restricted master problem which is then solved by iteratively adding cuts from the two smaller sub-problems. Sub-problem 1 is a optical multiplex section problem solved by LKH algorithm. Sub-problem 2 is a lambda assignment problem solved by welch-powell algorithm,

Based on the intuition that optimization-based methods provide optimality guarantee, and high adaptability to different task settings. We can uncover more (integer) linear programming acceleration techniques related to mathematical nature of the robot task planning problem. In addition, hardware-based acceleration such as GPU/FPGA programming may help as well. Moreover, most planning problems make assumption of full observability, leaving how to deal with stochastic optimization and partial observability a promising topic. Several approaches for partially observable TAMP have been extended to handle these challenges [30,31,44,45]. Future work could involve how to use optimization approach to solve the belief state planning problem.

3.2 Achievements in Robot Control with Generalizability

One of the projects during my placement at Ocado Technology is about Multi-agent Path Finding (MAPF) in large-scale modern warehouse. Plans should prevent robots from colliding while still reaching their goal. Most state-of-the-art MAPF planners still rely on centralized planning and scale poorly past a few hundred agents. Common approaches include search methods like M-Star [46] and heuristics using congestion probabilities [47], path prospects [48] or conflict search [49]. Such planners often prove inefficient in cluttered factory where they can result in dead and livelocks [50]. Hence it is critical to develop a planner that is robust to all kinds of scenarios. Thanks to the discrete nature of warehouse grids, the continuous motion dynamic can be ignored, making this problem a perfect Markov Decision Process.

We proposed an novel hybrid framework for decentralized MAPF that combines reinforcement learning. In this framework, agents learn to take into account the consequences of their position on other agents, in order to favor movements that will benefit the whole team and not only themselves. Using prevailing MARL algorithms such as COMA [51] and VDN [52], agents ultimately learn a decentralized policy where they still exhibit implicit coordination during online path planning without the need for explicit communication among agents. But RL policy is very hard to train due to huge samples of trial and error. Recently, there has emerged a trend to enhance learning-based methods with classical planners. An example would be using guided imitation learning (IL) to learn a policy

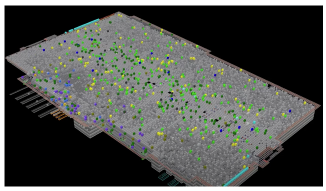

Figure 4: A multi-agent system modelled warehouse

guided by the expert MAPF planners [53]. Once learned, the resulting policy
can be copied onto any number of agents and naturally scales to different team sizes and world dimensions. However, the theoretical interpretability of NN-based methods is an open question. But we intend to further investigate the idea of applying learning-based methods to TAMP in the future.

Apart from the described two projects. My past experience [3] covers many aspects of hard-core robotics, including visual servo, motor control, and robotic system design. Overall, my past research experience grants me the adequate skills working as a Ph.D. candidate.

4 Methodology

4.1 Foundations and Insights

Recall the four questions listed in first section. **Q1** is partially answered by symbolic artificial intelligence which provide a holistic representation of predefined discrete task domain. **Q2** can be solved by the paradigm of TAMP that integrates symbolic planner with motion planning, that extends the planning down to continuous level. **Q3** cares about planning under state and observation uncertainty. Despite the achievements made by many researchers, we still encounter the problem of high computational complexity. On the other hand, variants of hierarchical TAMP (i.e. graph, optimization, Bayesian) are good at solving some particular problem scenes, but are limited to more complex scenes. **Q4** is pertinent to robot evolution w.r.t. environment changes. Neural-based methods (for both scene reasoning and task planning) help generalize to different motion compositions and object semantics, but it still lacks interpretability. In summary, the previous work fall into two categories of limitations : (1) the methods cannot find a optimal solution within short time interval. (2) the methods cannot generalize well at both task level and object level. We can hitherto propose our research direction based on the following foundations.

* (**F1**) Plan-skeleton and task abstraction leverage determinization in order to obtain major performance gains from off-the-shelf task planners.
* (**F2**) Uncertainty is central to all real-world robot applications, future TAMP methods should consider both future-state and present-state uncertainty.
* (**F3**) Further investigation is needed of strategies that combine computational efficient sampling and optimization approaches to TAMP that help the refinement of abstract task plans in dynamic state-space.
* (**F4**) Incorporating learning-based methods into planning will enable planners to reason with learned action models, requiring less human-provided domain knowledge (e.g. deformable objects, time, clutter).

4.2 Research Aims

F1 comes from empirical results shown in previous research [6, 23]. Symbolic task planning language enables us to best utilize the off-the-shelf task planners for generating abstract task sequence. **F2-3** is studied in [26,27]. While optimization-based methods has good optimality in various settings, computational efficiency is still an open question. Based on previous achievements, one approach is to explore mathematical techniques such as sequential convex optimization or Benders decomposition to accelerate the optimization-based planners. **F4** focuses on enabling the robots to evolve w.r.t. environment changes. We are likely to realize this by applying imitation learning or reinforcement learning alongside with classical planners. But how best exploit the benefits of each approaches is still an open question. Furthermore, extend existing solutions to cooperative multi-robot task and motion planning is probably useful and promising [54]. The potential contribution of this work can be summarized as follows.

* (**G1**) To propose a computational efficient, optimization-based TAMP framework that can better refine the task abstractions under uncertainties.
* (**G2**) To devise a online learning TAMP framework the combines RL/IL to enhance the adaptability to unseen scenarios.

[3] https://dieselmarble.github.io/

5 Timescale

Time Period	Milestones
2023.09 - 2024.08	1. Get used to life in Hong Kong; 2. Attend lectures related to AI and Robotics; 3. Confirm the topic of thesis proposal; 4. Start literature review;
2024.09 - 2025.02	1. Test prevailing benchmarks and baselines, and conduct comparison study; 2. Follow state-of-the-art robot task planning algorithm to get insights; 3. Summarize the work and accomplish one conference paper;
2025.03 - 2026.02	1. Propose the authors' own task planning algorithms; 2. Perform experiments on simulation and physical robot; 3. Summarize the work and publish one journal paper;
2026.03 - 2026.11	1. Conclude previous work and organize data; 2. Start to write the dissertation outline and first several chapters.
2026.12 - 2027.03	1. Finish the dissertation and prepare for the Ph.D. defense.

Table 2: Detailed timeline for the proposed Ph.D. study.

Successfully conduct the research plan requires both theoretical study and physical experiment/simulation. We divide the whole plan into three main stages.

In the first step, a robust, computational efficient motion planner will be designed to integrate with the traditional TAMP planner. Detailed mathematical proof and induction is a must. We plan to evaluate the proposed method on variations of manipulation tasks both in simulation environment and on physical robots. We will evaluate the result based on success rate in dynamic environment (e.g. with moving obstracle/target).

In the second step, we will investigate into how can learning-based methods help with current TAMP framework. Possible directions include learning sampling guidance and learning search guidance. Furthermore, we will particular look into imitation learning/reinforcement learning that help automated long-horizon task decomposition and sub-goal discovery. The generalization to new target domains could be tested with different object starting locations or varies the object's coefficient of friction. We can evaluate the results in multiple aspects, including the rate of convergence and the success rate of execution in both seen and unseen scenarios.

In the third step, we aim to research the robot task planning in various settings. For example, belief state planning, robot task planning with human collaboration and multi-robot assembly planning. Methods proposed in the previous two steps can be used to evaluate in these new problem settings.

6 Conclusion

Robot task planning has shown a promising future in many manufacturing, assembly lines, and household processes. Our work demonstrates the potential to take this intelligence one step further, targeting at building generalizable task planner under uncertainties. Specifically, an optimization-based motion planner will be integrated into the TAMP framework to leverage faster and robust planning with uncertainty. Then the learning algorithm to mitigate environment changes will be devised to improve the task planner performance. We intend to investigate these ideas further in future studies.

References

[1] E. Karpas and D. Magazzeni, "Automated planning for robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 417–439, 2020.

[2] R. Diankov, *Automated Construction of Robotic Manipulation Programs.* PhD thesis, USA, 2010. AAI3448143.

[3] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning and Acting.* Cambridge University Press, 2016.

[4] M. Stilman and J. Kuffner, "Navigation among movable obstacles: real-time reasoning in complex environments," in *4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, vol. 1, pp. 322–341 Vol. 1, 2004.

[5] L. Kunze, N. Hawes, T. Duckett, M. Hanheide, and T. Krajník, "Artificial intelligence for long-term robot autonomy: A survey," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4023–4030, 2018.

[6] M. Ghallab, C. Knoblock, D. Wilkins, A. Barrett, D. Christianson, M. Friedman, C. Kwok, K. Golden, S. Penberthy, D. Smith, Y. Sun, and D. Weld, "Pddl - the planning domain definition language," 08 1998.

[7] V. Lifschitz, "What is answer set programming?," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, p. 1594–1597, AAAI Press, 2008.

[8] Y. Jiang, S. Zhang, P. Khandelwal, and P. Stone, "An empirical comparison of pddl-based and asp-based task planners," *CoRR*, vol. abs/1804.08229, 2018.

[9] G. De Giacomo and M. Y. Vardi, "Linear temporal logic and linear dynamic logic on finite traces," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, p. 854–860, AAAI Press, 2013.

[10] K. He, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "Towards manipulation planning with temporal logic specifications," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 346–352, 2015.

[11] B. Bonet and H. Geffner, "Planning as heuristic search," *Artificial Intelligence*, vol. 129, no. 1, pp. 5–33, 2001.

[12] J. Hoffmann, "Ff: The fast-forward planning system," *AI Magazine*, vol. 22, p. 57, Sep. 2001.

[13] A. Blum and M. L. Furst, "Fast planning through planning graph analysis," in *IJCAI*, 1995.

[14] A. Coles, A. Coles, M. Fox, and D. Long, "Forward-chaining partial-order planning," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 20, pp. 42–49, May 2021.

[15] K. Hauser and J.-C. Latombe, "Multi-modal motion planning in non-expansive spaces," *The International Journal of Robotics Research*, vol. 29, no. 7, pp. 897–915, 2010.

[16] K. Hauser and V. Ng-Thow-Hing, "Randomized multi-modal motion planning for a humanoid robot manipulation task," *The International Journal of Robotics Research*, vol. 30, no. 6, pp. 678–698, 2011.

[17] M. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning - extended abtract," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 6231–6235, International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[18] M. Toussaint, "Logic-geometric programming: An optimization-based approach to combined task and motion planning," p. 1930–1936, 2015.

[19] Z. Wang, C. R. Garrett, L. P. Kaelbling, and T. Lozano-Pérez, "Learning compositional models of robot skills for task and motion planning," *CoRR*, vol. abs/2006.06444, 2020.

[20] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[21] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *CoRR*, vol. abs/2010.01083, 2020.

[22] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 639–646, 2014.

[23] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Pddlstream: Integrating symbolic planners and blackbox samplers," *CoRR*, vol. abs/1802.08705, 2018.

[24] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "STRIPS planning in infinite domains," *CoRR*, vol. abs/1701.00287, 2017.

[25] M. Weser, D. Off, and J. Zhang, "Htn robot planning in partially observable dynamic environments," in *2010 IEEE International Conference on Robotics and Automation*, pp. 1505–1510, 2010.

[26] T. Migimatsu and J. Bohg, "Object-centric task and motion planning in dynamic environments," *CoRR*, vol. abs/1911.04679, 2019.

[27] D. Hadfield-Menell, C. Lin, R. Chitnis, S. Russell, and P. Abbeel, "Sequential quadratic programming for task plan optimization," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5040–5047, 2016.

[28] E. Fernández-González, E. Karpas, and B. Williams, "Mixed discrete-continuous planning with convex optimization," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, p. 4574–4580, AAAI Press, 2017.

[29] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[30] L. P. Kaelbling and T. Lozano-Pérez, "Integrated task and motion planning in belief space," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1194–1227, 2013.

[31] D. Hadfield-Menell, E. Groshev, R. Chitnis, and P. Abbeel, "Modular task and motion planning in belief space," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4991–4998, 2015.

[32] M. Hanheide, M. Göbelbecker, G. S. Horn, A. Pronobis, K. Sjöö, A. Aydemir, P. Jensfelt, C. Gretton, R. Dearden, M. Janicek, H. Zender, G.-J. Kruijff, N. Hawes, and J. L. Wyatt, "Robot task planning and explanation in open and uncertain worlds," *Artificial Intelligence*, vol. 247, pp. 119–150, 2017. Special Issue on AI and Robotics.

[33] H. Kurniawati, Y. Du, D. Hsu, and W. S. Lee, "Motion planning under uncertainty for robotic tasks with long time horizons," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 308–323, 2011.

[34] N. Ye, A. Somani, D. Hsu, and W. S. Lee, "DESPOT: online POMDP planning with regularization," *CoRR*, vol. abs/1609.03250, 2016.

[35] A. Curtis, X. Fang, L. P. Kaelbling, T. Lozano-Pérez, and C. R. Garrett, "Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances," *CoRR*, vol. abs/2108.04145, 2021.

[36] T. Silver, R. Chitnis, A. Curtis, J. B. Tenenbaum, T. Lozano-Pérez, and L. P. Kaelbling, "Planning with learned object importance in large problem instances using graph neural networks," *CoRR*, vol. abs/2009.05613, 2020.

[37] H. Tian, C. Song, C. Wang, X. Zhang, and J. Pan, "Sampling-based planning for retrieving near-cylindrical objects in cluttered scenes using hierarchical graphs," *IEEE Transactions on Robotics*, pp. 1–18, 2022.

[38] C. Nam, S. H. Cheong, J. Lee, D. H. Kim, and C. Kim, "Fast and resilient manipulation planning for object retrieval in cluttered and confined environments," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1539–1552, 2021.

[39] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei, "Learning to generalize across long-horizon tasks from human demonstrations," *CoRR*, vol. abs/2003.06085, 2020.

[40] K. Hausman, Y. Chebotar, S. Schaal, G. S. Sukhatme, and J. J. Lim, "Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets," *CoRR*, vol. abs/1705.10479, 2017.

[41] D. Xu, S. Nair, Y. Zhu, J. Gao, A. Garg, L. Fei-Fei, and S. Savarese, "Neural task programming: Learning to generalize across hierarchical tasks," *CoRR*, vol. abs/1710.01813, 2017.

[42] B. Kim, L. P. Kaelbling, and T. Lozano-Pérez, "Guiding the search in continuous state-action spaces by learning an action sampling distribution from off-target samples," *CoRR*, vol. abs/1711.01391, 2017.

[43] W. Shen, F. W. Trevizan, and S. Thiébaux, "Learning domain-independent planning heuristics with hypergraph networks," *CoRR*, vol. abs/1911.13101, 2019.

[44] C. R. Garrett, C. Paxton, T. Lozano-Pérez, L. P. Kaelbling, and D. Fox, "Online replanning in belief space for partially observable task and motion problems," *CoRR*, vol. abs/1911.04577, 2019.

[45] C. Phiquepal and M. Toussaint, "Combined task and motion planning under partial observability: An optimization-based approach," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9000–9006, 2019.

[46] G. Wagner and H. Choset, "M*: A complete multirobot path planning algorithm with performance bounds," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3260–3267, 2011.

[47] C. Street, B. Lacerda, M. Mühlig, and N. Hawes, "Multi-robot planning under uncertainty with congestion-aware models," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, (Richland, SC), p. 1314–1322, International Foundation for Autonomous Agents and Multiagent Systems, 2020.

[48] W. Wu, S. Bhattacharya, and A. Prorok, "Multi-robot path deconfliction through prioritization by path prospects," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9809–9815, 2020.

[49] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.

[50] H. Ma, S. Koenig, N. Ayanian, L. Cohen, W. Hönig, T. K. S. Kumar, T. Uras, H. Xu, C. A. Tovey, and G. Sharon, "Overview: Generalizations of multi-agent path finding to real-world scenarios," *CoRR*, vol. abs/1702.05515, 2017.

[51] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," *CoRR*, vol. abs/1705.08926, 2017.

[52] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. F. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel, "Value-decomposition networks for cooperative multi-agent learning," *CoRR*, vol. abs/1706.05296, 2017.

[53] M. J. McDonald and D. Hadfield-Menell, "Guided imitation of task and motion planning," *CoRR*, vol. abs/2112.03386, 2021.

[54] J. Chen, J. Li, Y. Huang, C. Garrett, D. Sun, C. Fan, A. Hofmann, C. Mueller, S. Koenig, and B. C. Williams, "Cooperative task and motion planning for multi-arm assembly systems," 2022.