Task-Aware Waypoint Sampling for Robotic Planning (abridged version*)

Sarah Keren[1,2†], Gerard Canal[3] and Michael Cashmore[4]

[1]Harvard University, School of Engineering and Applied Sciences
[2]The Hebrew University of Jerusalem, School of Computer Science and Engineering
[3]Department of Informatics, King's College London
[4]University of Strathclyde

Abstract

To achieve a complex task, a robot often needs to navigate in a physical space to complete activities in different locations. For example, it may need to inspect several structures, making multiple observations of each structure from different perspectives. Typically, the positions from which these activities can be performed are represented as *waypoints* – discrete positions that are sampled from the continuous physical space and used to find a task plan. Existing approaches to waypoint selection either iteratively consider the entire space or use domain knowledge to consider each activity separately. This can lead to task planning problems that are more complex than is necessary or to plans of compromised quality. Moreover, all previous approaches only consider geometric constraints that can be imposed on the waypoint selection process.

We present Task-Aware Waypoint Sampling (TAWS), which offers two key novelties. First, it is an anytime approach that combines the benefits of random sampling with the use of domain knowledge in waypoint selection by performing a one-time computation of the connectivity graph from which waypoints are sampled. In addition, TAWS is the first approach that accounts for *performance preferences*, which are preferences a system operator may have about the generated task plan. These can account, for example, for areas near doorways where it is preferable that the robot does not stop to perform activities. We demonstrate the performance benefits of our approach on simulated automated manufacturing tasks.

Introduction

Robots are typically assigned complex missions that require performing various activities in different locations. To complete the overall mission, a mobile robotic agent must reason about a physical space and decide both which activities must be performed as well as how to navigate between the positions from which it can perform each activity. Since the physical space is continuous, task planning is typically performed using an abstraction of the space. A common abstraction approach is to use a finite set of discrete *waypoints* that represent configurations (positions) in the space. The waypoints represent nodes in a *probabilistic road map* (PRM) (Kavraki et al. 1996), in which the edges represent feasible paths between waypoints and their estimated navigation costs. The PRM is used by a *task planner* to find a sequences of activities and navigations between waypoints that accomplishes the assigned task.

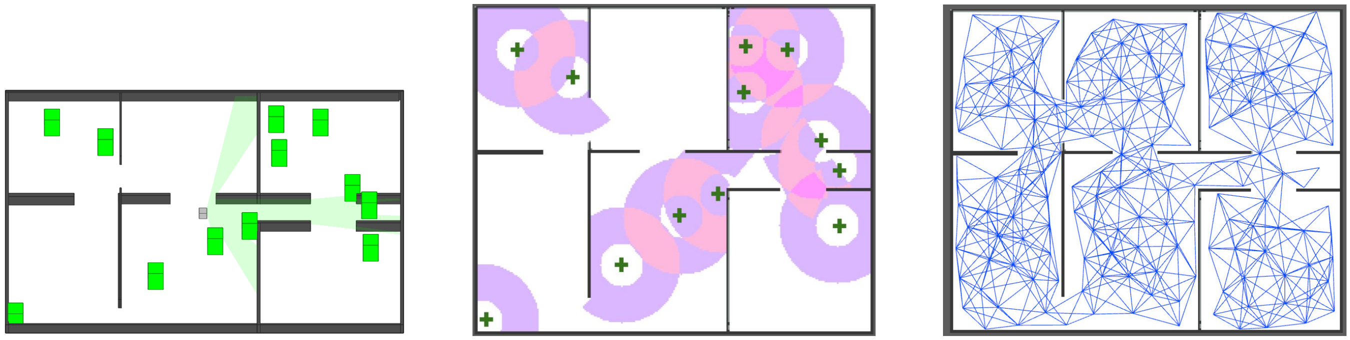
When generating waypoints there is a trade-off between the complexity and completeness of the resulting representation. Intuitively, a small set of waypoints is a coarse abstraction of the physical space that limits the positions that can be used to perform the task, potentially leading to lower quality plans or unsolvable problems. On the other hand, a larger set of waypoints will lead to a higher probability of finding a plan, but may exceed the computational capacity of the task planner.

Generally, there are two common approaches to waypoint generation. With *Fixed Waypoint Generation* (FWPG), a PRM is generated by selecting a single waypoint for each possible activity (Edelkamp et al. 2018). FWPG provides a sufficient coverage of the planning space, but may yield representations that are too big for the planner to handle. On the other end, with a *Pure PRM* (PPRM) approach, a PRM (Kavraki et al. 1996) is created by randomly sampling waypoints. The size of the graph can be set to comply with the planner's capacity, but since the placement of waypoints is random, the coverage of the space may be insufficient, which requires iteratively generating a new PRM, until a solution is found.

In this work we suggest a novel approach to waypoint generation which bridges the gap between the two common approaches to sampling and provides good coverage of the space, while accounting for the planner's capacity. Our approach, Task-Aware Waypoint Sampling (TAWS), is an anytime approach that starts by generating a very *Dense PRM* (DPRM) that captures a fine representation of the reachability information of the space, and includes with very high probability a representation of a solution to the task. TAWS then launches an iterative planning process, sampling at each step waypoints from the DPRM according to probabilities induced by the task description. If a plan cannot be found using the current set of waypoints or if there is time to improve the quality of the current best solution, a larger set is re-sampled. In contrast, if the planner's capacity is exceeded, a smaller set is re-sampled.

*The full version of the paper can be found in the proceeding of the 31st International Conference on Automated Planning and Scheduling (ICAPS 2021)

†Contact Author

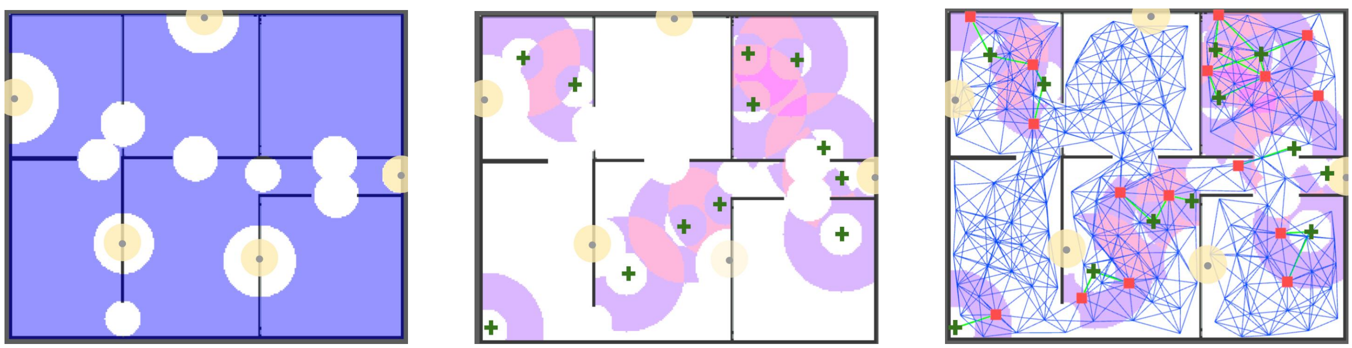


(a) 3D representation of an example factory environment, with the robot (gray square in the center), and the machines in green.

(b) A map of the physical space highlighted by the areas from which the robot can interact with each of 12 different machines (green crosses).

(c) The dense PRM imposed over the map. Each collision-free edge (blue lines) connects two waypoints (configurations).

Figure 1: An example setting from the RCLL domain



(a) Map of the physical space specifying first aid stations (yellow) and doorways (white), which represent performance preferences of areas where robots can never block and preferably should avoid, respectively.

(b) Map representing the combination of the performance preferences and the machine interaction areas.

(c) Waypoints (red squares) are sampled from the DPRM for task planning. Machines that can be accessed from each waypoint are highlighted by green edges.

Figure 2: Integrating Performance preferences into the sampling process

As with PPRM, TAWS relies on sampled waypoints for task planning. However, it avoids the need to reconstruct a PRM at every planning iteration. As with FWPG, TAWS incorporates domain knowledge into the waypoint selection process, but instead of using it to fix a set of waypoints, it uses it to set the probabilities according to which waypoints are sampled from the DPRM. This can be used to increase efficiency by, for example, prioritizing waypoints from which more than one activity can be performed. Moreover, it makes it possible to account for *performance preferences*, arbitrary user-defined preferences over positions from which the robot can perform its activities but that cannot be directly represented in the map used by the robot for navigation. Such preferences can reflect, for example, social norms (e.g., areas where some social event is taking place and should be avoided by noisy robots), safety and efficiency constraints (e.g., a carpeted area that is hard for robots to traverse, or an area near a first aid kit robots shouldn't block), and areas where performance is enhanced (e.g., it is preferable for a robot to operate near a charging station since it will be able to recharge and recover if its battery is unexpectedly depleted). TAWS can account for these arbitrarily defined preferences by changing the probability of sampling certain positions according to the specified preferences.

Example 1 *Consider the scenario used for the Robocup Logistics League (RCLL) (Niemueller, Lakemeyer, and Ferrein 2015) and depicted in Figure 1a in which robots must navigate in a factory in order to collect items from a set of machines and deliver them to their destinations. In such scenarios, if the environment is fixed and known, FWPG can be used to prescribe a finite set of waypoints, including a waypoint for each activity robots may need to perform, such as picking up an item from a machine. The result may include many redundant waypoints or waypoints that cannot be connected, or lead to inefficient plans since each activity is considered separately. Also, FWPG does not allow for iterations if the planner's capacity is exceeded. On the other hand, the PPRM approach might require many iterations to* *solve problems of realistic size or produce inefficient plans since the size of the PRM and the accuracy of its cost estimations is limited by the planner's capacity.*

TAWS takes a hybrid approach by first producing a single dense PRM (DPRM) that is used throughout the search for a plan. This is likely to lead to plans that are more efficient at execution since the DPRM provides more accurate navigation cost estimates. Moreover, it can be used to reduce the probability that robots block doorways or make sure that robots never block access to first aid stations.

We offer two key contributions. First, we suggest performing a one-time computation of a connectivity graph in a given environment, thus decoupling between the connectivity analysis and the task planning process. This allows us to use high quality cost estimations throughout the planning process, regardless of the size of the current representation of the task planning problem or the capacity of the task planner. Secondly, we support performance preferences that induce the waypoint sampling probabilities and yield plans that comply with both hard constraints and user-defined preferences regarding the way a robot accomplishes its task.

To demonstrate the performance benefits of TAWS we use a set of simulated manufacturing tasks in an automated factory. First, we show that the use of a DPRM instead of an iterative or fixed generation of a PRM yields shorter plans that are computed more efficiently. In addition we show that TAWS produces plans that maximize compliance with the specified performance preferences when compared to current approaches, without compromising their quality in terms of plan duration.

Related Work

Typical robotic control systems must determine which activities must be performed, and how to navigate between those activities. Common approaches to planning for robots combine *motion planning* and *task planning* (Gravot, Cambon, and Alami 2005; Cambon, Alami, and Gravot 2009; Kaelbling and Lozano-Pérez 2011; Dornhege, Hertle, and Nebel 2013; McMahon and Plaku 2014; Srivastava et al. 2014; Toussaint 2015; Fernández-González, Karpas, and Williams 2017; Canal et al. 2018; Garrett et al. 2020). Motion planning is the process of finding a plan to perform a basic activity, such as picking up an item or moving between two adjacent locations. Task planning is the search for a sequence of activities that is predicted to achieve the goal, while minimising duration and other costs such as energy use.

When planning in complex scenarios, task planning typically uses an abstraction of the space. One way to abstract the space is by using geometric computations that help the high-level planner make appropriate choices. For example, (Kaelbling and Lozano-Pérez 2011) handle the integration of continuous geometric planning with task planning by using geometric "suggesters", which construct configurations dynamically during an "aggressively" hierarchical planning process. Another approach integrates the motion planner's geometric search for positions into the symbolic forward-search of a task planner. For example, (Cambon, Alami, and Gravot 2009) devise an integrated task and motion planner that reasons about geometric constraints that describe the positions from which it is possible to accomplish some action as sub-manifolds of the configuration space of the robot. These sub-manifolds are mapped within the solver to high level symbols.

Another approach to abstraction uses *waypoints* that represent discrete positions (Cashmore et al. 2014; McMahon and Plaku 2014; Edelkamp et al. 2018). This reduces the complexity of the problem, making it possible to focus on the task-planning aspect of the problem, i.e., selecting and scheduling activities, while using heuristic approximations to estimate navigation and motion costs. Once a high-level task plan is produced, motion planning is delegated to a low-level motion planner. One of the benefits of this approach is that it is typically agnostic to the specific task and motion planners used.

In this paper we focus on waypoint-based approaches and and on the selection of waypoints for task planning. Waypoints can be selected randomly, for example using a PRM (Kavraki et al. 1996), or can be generated using knowledge of the space and task (Plaku and Hager 2010; Edelkamp et al. 2018). The disadvantage of the random approach is that in order to ensure coverage of all interesting areas, a large number of waypoints might be required. For simple problems, such as inspection missions (Cashmore et al. 2014), this can be feasible. However, in a more complex task this will result in problems that are too hard to solve within a reasonable time.

On the other hand, generating fixed waypoints means that for each affordance in the physical space (corresponding to a non-navigation action that the robot might make) a set of waypoints of fixed size is generated. To demonstrate, in the RCLL setting in Example 1, the approach by (Edelkamp et al. 2018) generates a separate waypoint for each item pickup activity by randomly sampling a position around the machine the item is positioned at, even if the machine has more than one item. These waypoints are connected together using a PRM, adding additional waypoints to cover the space, if needed. The resulting representation is guaranteed to include a solution. However as the number of activities increases, the size of the resulting task planning problem may unjustifiably exceed the planner's capacity, containing many redundant waypoints.

Another limitation of the fixed waypoint selection approach is that it completely relies on domain knowledge to select a waypoint for each activity. In some cases such domain knowledge may not be always available. In domains with complex configuration spaces, it may not be possible to explicitly prescribe in advance the region from which an activity can be performed, making it necessary to sample waypoints and determine whether an activity is achievable from them. For example, consider a mobile base carrying an arm with 5-degrees of freedom, performing a picking task in a cluttered scene. Due to the clutter, it is not possible to describe in advance a region for the base from which it is guaranteed that the arm can reach the target. However it is possible instead to sample a position and orientation for the base and use a motion planner to determine if there is a collision-free path for the arm to the target.

We suggest a new approach to waypoint sampling that combines the benefits of random sampling with the use of domain knowledge. In contrast to a fixed approach to waypoint selection, TAWS is in an anytime approach that iteratively improves solution quality. In contrast to a random sampling approach, it decouples the connectivity analysis of the domain by creating a single dense PRM (DPRM), and uses domain knowledge to induce the probabilities according to which waypoints are iteratilvey sampled from the DPRM, and sent to a task planner that is chosen by the user. This means that the quality of the cost estimations of navigating between waypoints does not depend on the current number of waypoints that are used to represent the task. Most notably, all approaches mentioned above only consider geometric constraints that can be imposed on the waypoint selection process. TAWS is the first approach that also accounts for arbitrary performance preferences, thus making it possible to prioritize or discourage specific behaviors.

Task Aware Waypoint Sampling (TAWS)

The input to the *Task Aware Waypoint Sampling* (TAWS) problem is a tuple $p = \langle M, A, F \rangle$, where

- M is the set of configurations $m \in \mathbb{R}^n$, where n represents the dimensions of the space,
- A is a set of non-navigation activities that can be performed, and
- F is a set of performance preferences. Each preference $f \in F$ is a score function $f : M \to \mathbb{R}_{\geq 0}$.

Each sampled waypoint corresponds to a configuration $m \in \mathbb{R}^n$. Each activity $a \in A$ is associated with a function $\omega_a : M \to [0, 1]$ specifying the probability of successfully executing a from configuration m. Typically, these probability functions are generated using prescribed templates for each activity type the robot can perform. Preferences $f \in F$ are used to describe areas from which it is (un)desirable that the robot operates.

In Example 1, a robot navigates the factory floor and can interact with a number of stationary machines. For simplicity, we ignore the orientation of the robot, and describe the configuration space as a 2-dimensional map (the floorplan) i.e., $m \in \mathbb{R}^2$. The activity set represents the possible interactions of the robot with each machine (e.g., picking up an item from a machine). The function $\omega_a : \mathbb{R}^2 \to \{0, 1\}$ of each machine is defined by a prescribed template that defines the probability of successfully completing the activity in a given configuration, taking into account adjacent obstacles (e.g., walls) and the extent of the robot's arms. Figure 1a shows an example setting with 12 stationary machines. In this setting, each activity is deterministically mapped to configurations from which it can be achieved. The areas from which it is possible to pickup objects from a machine are depicted by rings around each machine (Figure 1b). The areas in pink are those from which more than one activity can be performed. The performance preferences can prioritize sampling from these areas. This can yield shorter plans in settings in which more than one object needs to be collected from a single machine or settings in which it is possible for a robot to reach more than one machine from a single waypoint without the need to move. The performance preferences can also be used to increase the probability of sampling waypoints that are near charging stations or at doorways. They can also be used to guarantee no waypoint is sampled near first aid stations. Previous approaches that consider each activity separately do not account for such task level considerations when selecting the waypoints that are sent to the task planner.

Sampling Procedure

The TAWS approach decouples the connectivity analysis of a domain, which provides estimations of navigation costs within the physical space, and the task planning process, which finds a sequence of activities that accomplish the assigned task. First, it generates a *Dense PRM* (DPRM) over the configuration space. The process starts from the robot's initial position. The PRM is constructed by iteratively selecting a waypoint from the existing PRM for expansion. A set of new waypoints is cast from the chosen waypoint. Waypoints that are not in collision with any obstacle in the map (and that are traversable by the robot), are added to the graph. The coordinates of each node and the length of each straight edge are stored so that they can be used to estimate the cost of traveling between the waypoints during the task planning process. The accuracy of the estimate is correlated with the resolution of the DPRM. A DPRM for the factory domain in Example 1 is shown in Figure 1c.

After completing the generation of the DPRM, the iterative task planning stage begins. At each iteration, a number of waypoints are sampled from the DPRM and sent to the task planner to find a sequence of reachable activities that accomplishes the task.

TAWS is an anytime approach; even if it finds a solution, it will continue to search for better solutions until it is halted. If a plan is found, it is recorded, and the number of waypoints is incremented in order to find a more efficient solution. If the planner is unable to solve the problem within a time bound, the number of waypoints is decremented. If the planner claims that the problem is unsolvable, the number is incremented. This process is repeated iteratively until timeout is reached.

The selection of waypoints at each iteration is done according to the following procedure:

1. A sampling probability is assigned to each waypoint in the DPRM, using a task specific score which is induced by the activities and the performance preferences $f \in F$ and discussed in detail in the next section.
2. A waypoint is sampled from the DPRM and added to the task plan's model. The distance between the new waypoint and all existing waypoints is calculated by finding the shortest path through the edges of the DPRM. This value is added to the planning model as an estimate of the path's cost. In addition, the planing model is updated with information about all the activities that can be performed from the new waypoint.

3. The score function is updated to reduce the probability of sampling more waypoints near the one sampled. The radius for reducing the probability near a sampled waypoint is defined by the user.

After selecting the required number of waypoints, the resulting model is sent to a task planner that is chosen by the user.

Note that as opposed to previous approaches to waypoint sampling that were mentioned in the previous section, TAWS iteratively selects waypoints from the DPRM, and not from the underlying map. This means that even when the number of waypoints that are sent to the planner decreases, the quality of the estimations of the costs of navigating between the waypoints is not compromised. This is due to the fact that these estimations are evaluated according to the DPRM, which contains connectivity information about the entire space, regardless of the number of waypoints in the model that is sent to the task planner.

Combined Score

In order to account for both the probability of successfully accomplishing an activity from a given configuration as well as the performance preferences, TAWS uses a single *combined score* (CS) that associates a score to each waypoint (corresponding to a sampled configuration) in the DPRM. This score, that can be defined arbitrarily to account for different settings, is normalized over the waypoints in the DPRM, and is used to specify the probability of sampling each waypoint for inclusion in the task planning problem.

Specifically, the CS we suggest in Equation 1 can be used for the type of settings we consider here. The score uses a weighted sum over the different activities, while considering preferences multiplicatively.

$$CS(m) = \prod_{f \in F} f(m) \sum_{a \in A} \omega_a(m) \tag{1}$$

The above scoring approach increases the score (and corresponding sampling probability) of waypoint m from which multiple actions can be achieved by summing the probabilities $\omega_a(m)$ of successfully completing each activity from m. The application specific performance preferences can be used to account for anything from breaking ties between otherwise equally probable waypoints, to imposing hard constraints that prevent sampling in certain regions. The former case can be achieved by setting $f(m)$ to vary between $1 - \epsilon$ and 1 for some small value ϵ, so that the score of a waypoint is scaled down by up to $1 - \epsilon$ in areas where it is preferred not to sample a waypoint. This may be relevant, for example, in settings where it is preferred that a noisy robot avoids getting close to a working station of a human worker. The latter case, hard constraints such as for ensuring that a robot never blocks access to a first-aid station, can be enforced by setting $f(m)$ to 0 in the critical area. This ensures that no waypoint can be sampled in that area, as its combined score and therefore sampling probability will be 0.

In Figure 2a, critical areas represent doorways and first aid stations. The combined score is assigned according to CS, the cost function in Equation 1, that considers both the performance preferences and the activity information (Figure 2b). The score is normalized and used to set the probability of sampling each waypoint from the DPRM. Once a once a waypoint is sampled, the probability of sampling nearby waypoints is reduced. The sampled set is used to search for a plan for the task (Figure 2c).

Evaluation

Due to the workshop's space constraints, we omit here the empirical evaluation. The full version of the paper, including the complete description of our empirical evaluation, appears in the proceeding of the 31st International Conference on Automated Planning and Scheduling (ICAPS 2021) - https://icaps21.icaps-conference.org/. Videos of our simulated scenario are available at https://vimeo.com/user129497320. The source code and experimental setup including all problem files can be found at *https://github.com/sarah-keren/ROB-IS*.

Conclusion

We presented Task-Aware Waypoint Sampling (TAWS) as a new approach to selecting waypoints for task planning. TAWS's novelty is in the way it decouples the connectivity analysis of a domain from the task planning process, and its ability to account for user defined performance preferences.

The connectivity information is captured through a Dense PRM (DPRM), which is generated once and reused through the anytime iterative planning process to provide high quality estimations of navigation costs. The task planning problem at each iteration is constructed by sampling waypoints from the DPRM according to probabilities that are defined by both the activities that are part of the domain description and the performance preferences.

Our empirical evaluation on a set of automated factory problems shows that TAWS finds solutions that maximize compliance with the specified preferences, without compromising computation time and plan duration.

In the future, we intend to evaluate TAWS in other settings beyond the factory use-case. Specifically, we intend to investigate high-dimensional exploration scenarios, in which the sampling probability of TAWS can be used to specify areas in which a more meticulous search is desired. In addition, evaluation in this work focused on comparing the estimated plan duration of each approach. As a next step we intend to include in our evaluation the plan execution, comparing the execution time of the plans generated by each approach. Finally, in this work, desirable behaviors were induced by changing the way in which the model for task planning was built. Therefore, we will investigate how TAWS can be adapted for settings in which robots are treated as "black boxes" and their inner implementation cannot be modified. In such settings, their behavior can instead be influenced by changing the information that is provided to them, such as the map that is used for navigation.

References

Cambon, S.; Alami, R.; and Gravot, F. 2009. A Hybrid Approach to Intricate Motion, Manipulation and Task Planning.

The International Journal of Robotics Research 28 (1): 104–126.

Canal, G.; Pignat, E.; Alenyà, G.; Calinon, S.; and Torras, C. 2018. Joining high-level symbolic planning with low-level motion primitives in adaptive HRI: application to dressing assistance. In *IEEE International Conference on Robotics and Automation (ICRA)*, 3273–3278. ISSN 2577-087X. doi: 10.1109/ICRA.2018.8460606.

Cashmore, M.; Fox, M.; Larkworthy, T.; Long, D.; and Magazzeni, D. 2014. AUV mission control via temporal planning. In *Proceedings - IEEE International Conference on Robotics and Automation*. ISSN 10504729. doi: 10.1109/ICRA.2014.6907823.

Dornhege, C.; Hertle, A.; and Nebel, B. 2013. Lazy evaluation and subsumption caching for search-based integrated task and motion planning. In *IROS workshop on AI-based robotics*.

Edelkamp, S.; Lahijanian, M.; Magazzeni, D.; and Plaku, E. 2018. Integrating Temporal Reasoning and Sampling-Based Motion Planning for Multigoal Problems With Dynamics and Time Windows. *IEEE Robotics and Automation Letters* 3(4): 3473–3480.

Fernández-González, E.; Karpas, E.; and Williams, B. C. 2017. Mixed Discrete-Continuous Planning with Convex Optimization. In *AAAI*.

Garrett, C. R.; Chitnis, R.; Holladay, R.; Kim, B.; Silver, T.; Kaelbling, L. P.; and Lozano-Pérez, T. 2020. Integrated task and motion planning. *arXiv preprint arXiv:2010.01083* .

Gravot, F.; Cambon, S.; and Alami, R. 2005. aSyMov: A Planner That Deals with Intricate Symbolic and Geometric Problems. *Robotics Research. The Eleventh International Symposium. Springer Tracts in Advanced Robotics* 15: 100–110.

Kaelbling, L. P.; and Lozano-Pérez, T. 2011. Hierarchical task and motion planning in the now. In *2011 IEEE International Conference on Robotics and Automation*, 1470–1477.

Kavraki, L. E.; Svestka, P.; Latombe, J. .; and Overmars, M. H. 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* 12(4): 566–580.

McMahon, J.; and Plaku, E. 2014. Sampling-based tree search with discrete abstractions for motion planning with dynamics and temporal logic. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, 3726–3733.

Niemueller, T.; Lakemeyer, G.; and Ferrein, A. 2015. The RoboCup logistics league as a benchmark for planning in robotics. *Planning and Robotics (PlanRob-15)* 63.

Plaku, E.; and Hager, G. D. 2010. Sampling-Based Motion and Symbolic Action Planning with geometric and differential constraints. In *IEEE International Conference on Robotics and Automation*, 5002–5008.

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

Toussaint, M. 2015. Logic-Geometric Programming: An Optimization-Based Approach to Combined Task and Motion Planning. In *IJCAI*, 1930–1936.