Domain Specific Situated Planning – Dissertation Abstract

Devin Thomas

University of New Hampshire, Advisor: Dr. Wheeler Ruml

Abstract

There has been much recent work on finding paths in grid maps among moving obstacles. However, in addition to assuming complete omniscience regarding the map and the obstacles' trajectories, previous work has also assumed that time stands still while the agent plans. My dissertation addresses situated pathfinding, in which time passes and the obstacles continue to move while the agent plans. I will study situated planning in three domains: Grid pathfinding among moving obstacles, orienteering and opportunistic science.

Introduction

Traditionally when planning we assume that we receive the problem instance as input, then formulate a plan, then the clock begins and the agent executes the plan. Sometimes we would prefer to consider time passing as we plan, which is situated planning. I am working on situated planning in three domains, Grid pathfinding among moving obstacles, orienteering and opportunistic science.

When planning, our aim is to find a quick and safe path for our agent to reach its goal. While realtime planning algorithms address this by setting a fixed time bound for the agent to return an incremental plan. In contrast a situated agent plans "as the clock ticks" (Cashmore et al. 2018a) time passes, whether the agent is moving, thinking or waiting. Situated planning is less rigid than realtime planning, the agent may choose to spend additional time planning if it believes it will benefit from doing so. This "metareasoning" about when and for how long to plan maybe be important for some situated agents.

In my dissertation we will explore situated planning in several domains. First in grid based path planning, where the environment includes both static and moving obstacles and the agent seeks to minimize it's goal achievement time while avoiding collision with any obstacles. Second in the orienteering problem, where the agent is given a graph containing nodes with varying values, and has a time limit to visit some portion of those nodes and accumulate as much total value as it can. Thirdly the problem of opportunistic science, where the agent may have a window of time to amend its plan to take advantage a transient measurement. In the remainder of this abstract I will give an overview of the background of situated planning, and each of the domains we intend to work on.

Situated Planning

(Russell and Wefald 1991) argue that computations are actions, and the utility of such an action should be derived from its effect on the agents choice of actions. This utility can be estimated from statistical knowledge of the utility of previous computation actions. More recently the problem of situated planning was posed by Cashmore et al. (2018a), where the planner understands that execution is waiting on planning. This allows the planner to prune partial plans where the planning would likely finish too late to execute. In domain independent planning with absolute deadlines this showed empirical improvements over a baseline planner which had a set planning time, and moved all the deadlines earlier by that same planning time. (Shperberg et al. 2019) formalize the metareasoning problem allocating planning effort when actions expire (AE2) where the situated agent must decide which search nodes to expend planning effort on, when they each have an expiration time, and expected completion effort in addition to the normal cost. They optimize to maximize the probability that at least one solution is found by the deadline. They develop a formal MDP solution for AE2, and empirically demonstrate a greedy scheme which was near optimal in solution quality, while also fast enough to be used in metareasoning. The delay-damage aware (DDA) greedy scheme presented by (Shperberg et al. 2021) provides an optimal pseudo-polynomial solution in the case of known deadlines and, a fast greedy scheme that shows improvements over previous schemes with unknown deadlines. Situated planning is an area of active work, and it remains unclear how much of the sophisticated theoretical work can be applied to concrete problems. This dissertation focuses on three different domains where we can test the utility of these metareasoning algorithms.

Three Problem Domains

We have three domains which we will explore situated planning in. This first is a 2D grid with static and moving obstacles, this setting has been the focus of the work thus far. The others are orienteering and opportunistic science which we have discussed as potential domains to explore, but have not begun working in yet.

Grid Pathfinding Among Moving Obstacles

The problem statement for situated grid pathfinding among moving obstacles is: we have an agent that is situated on a 2D grid, with 8-way movement plus the ability to wait in place. The environment contains static and moving obstacles, the safety of the agent is binary. If the agent collides with an obstacle at any point in time it dies, otherwise it is safe. The agent has a starting location, and a goal location to reach as quickly as possible avoiding collision with any obstacles. The cost of the agent's actions is the time they take to complete, and the state space is $\langle x, y, t\rangle$ where the location is discrete and the temporal dimension continuous.

More formally, situated grid pathfinding among moving obstacles is a 6-tuple $\langle S, N, A, C, s_{start}, G\rangle$ where:

- S is a set of states, which are tuples $\langle x, y, t\rangle$ representing the agent's discrete grid cell and real-valued time.
- I is a function from x, y locations to a set of intervals $\{[t_i, t_j], \ldots, [t_k, t_l]\}$, representing safe times when the agent can occupy the grid point at $\langle x, y\rangle$ without colliding with a dynamic obstacle. Static obstacles correspond to grid cells with no safe intervals. Dynamic obstacles create unsafe intervals corresponding to the grid cells they occlude.
- A is a set of actions where each action $a \in A; a : S \rightarrow S$ has a duration t_a. We use 8-way motion augmented with a wait action.
- C is the mapping of actions to their durations. $C : A \rightarrow \mathbb{R}+$, we use $C(a \in A) = t_a$. $\{up, down, left, right\}$ have a duration of 1, diagonal motions have a duration of $\sqrt{2}$.
- G is the goal grid cell, $\langle x, y\rangle$.
- $s_{start} \in S$ is the starting state.

Given a SSIPP problem presented to an agent at time t_0, a emergent solution plan is defined as a sequence of actions $(a_0, a_1, ..., a_i, ..., a_N)$ emitted by the agent, where

- The solution begins at S_{start}.
- Define $Succ(s, a) : S \times A \rightarrow S$ the successor function returning the result of applying action a at state s.
- We can then recursively define the states of the path $s_i = Succ(s_{i-1}, a_{i-1})$ with s_0 being the S_{start}.
- Each action is feasible, the states s_i are always within a safe interval, and thus never in collision with any obstacle.
- the agent ends at the goal cell after a finite sequence of actions.

The objective of the agent is to execute a plan that reaches a goal state as quickly as possible.

The offline equivalent to this problem is addressed by safe interval path planning(SIPP). The safe intervals are constructed by grouping all consecutive co-located states into a safe interval at that location, $i \in I$. This compresses the continuous time dimension into a compact discrete representation which can then be solved with optimally with A^* (Phillips and Likhachev 2011) (Yakovlev and Andreychuk 2017), or sub-optimally with variations of weighted A^* or focal search (Yakovlev, Andreychuk, and Stern 2020) or with anytime algorithms which run fast enough to be used in soft-realtime on a specific problem instance (Narayanan, Phillips, and Likhachev 2012).

SIPP and its variants take advantage of the property that when searching offline there is a built in dominance of states within an interval, with earlier arrival into an interval always being better. This is no longer the case when the agent is situated, as a hasty agent may miss out on opportunities that require more careful deliberation. Because of this our situated problem requires a more sophisticated handling of intervals to be correct. Our more sophisticated method, and how and when it is an improvement over simpler or incorrect methods is part of our upcoming paper.

In this domain we are also exploring how cutting edge methods from realtime search apply to situated planning. The situated agent must perform heuristic learning in order to escape local minima, the agent can back up information from the search frontier using methods like local search space real time A*(LSS-LRTA*) (Koenig and Sun 2009), potentially with separate learning of the static environment from the dynamic environment using partitioned learning realtime A*(PLRTA*) (Cannon, Rose, and Ruml 2014). Our agent must also have a strategy for picking which search nodes to expand first, in similar realtime settings it has been shown to be beneficial to use a $\hat{f}(n) = g(n) + \hat{h}(n)$ rather than $f(n) = g(n) + h(n)$ where $\hat{h}(n)$ is the expected cost to goal at search node n, rather than a heuristic cost to goal $h(n)$ (Kiesel, Burns, and Ruml 2015). The agent also may benefit from committing to more than one action at a time, thus increasing the amount of time it has to plan the next set of actions. One method would be to commit all the way to the search frontier, which then allows a situated planner to plan even further for the next set of actions thus dynamically increasing the size of partial plan committed to (Kiesel, Burns, and Ruml 2015). For realtime planning that scheme has been shown to be too aggressive, leading the agent to over-commit. (Cheng and Ruml 2019) found that a constrained dynamic scheme, where the size of set of actions committed to was allowed to grow, but only by a fixed amount per iteration was better.

This initial grid problem setting while simple, provides an arena to test how methods from realtime planning can be adapted to the situated setting. We have begun our experimentation using the same set of instances used by the bounded-suboptimal SIPP paper (Yakovlev, Andreychuk, and Stern 2020). Our initial results suggest that with appropriate choices of state space representation and learning algorithm, a situated agent can perform very well in these instances. As such it is not yet clear if this setting will be suitable, or sufficiently complex to explore our questions on how the agent should metareason.

The grid path planning domain has been the focus of our work so far, we are in the process of preparing a paper exploring the effects of choices in state space representation and heuristic learning on the success of a situated agent. Following the paper we plan to finish exploring the effects of other methods that have been found to be beneficial in realtime search in the situated setting, such as altering the choice of expansion algorithm and the commitment strategy. Additionally so far we have used problem instances from (Yakovlev, Andreychuk, and Stern 2020), we would like to construct instances of our own, for example instances of the game Frogger would be appropriate for this situated agent.

The Orienteering Problem

In a orienteering problem, the agent is given a starting location, and a set of checkpoints each with some score associated with it. The agent seeks to visit these checkpoints and return to the starting location, such that it maximizes the sum of the scores it accumulates while returning by some deadline (Vansteenwegen, Souffriau, and Oudheusden 2011). The orienteering problem with time windows, augments the orienteering problem with locations who's point value is only captured if the agent reaches them within a certain time interval. In this way it is similar, but distinct from the safe intervals mentioned before.

Orienteering is a natural problem to consider in a situated manner, as only limited information of the problem instance is available prior to starting the race. Additionally a situated agent may benefit from metareasoning on how long to spend planning, especially in the orienteering problem with time windows, when spending a substantial time at the start planning might mean missing out on quick deadlines. A situated agent in this setting must balance its short term goals to accumulate the most points, with its long term constraint that it must return by the time limit.

Opportunistic Science

The opportunistic science domain is an agent who is executing a long term plan which it must complete. During execution the agent is presented with an unexpected opportunity to achieve a large reward by expending some surplus resources it is holding in reserve, whether those resources are time, or battery charge or sampling capacity. These opportunities are transient, and rare or otherwise impractical to plan for as part of the long term plan. Similar to the orienteering problem, the agent must keep in mind the need to maintain adequate resources to complete its long term goals, while reacting to opportunities in such a way as to maximize its benefit.

There are numerous autonomous agents who may encounter transient scientific opportunities. This can be a martian rover who could observe a dust devil and record it, or use it to clean off it's solar panels (Lorenz and Reiss 2015). This has included a focus on on-board systems to adjust to unexpected surplus or deficits in rover resources (Gaines et al. 2006) (Rabideau et al. 2020). Similarly this problem has been explored with autonomous submersibles performing underwater maintenance (Cashmore et al. 2018b). Those agents represent systems with high latency to get assistance planning, in contrast there are systems like the CHIME radio telescope, which generates terabytes of data each second, and as such can only buffer the data for a matter of seconds while it decides what events are most worth recording (Amiri et al. 2018).

Conclusions

Situated planning in contrast with traditional offline planning has time progress while the agent plans. Prior work in situated planning has been mainly theoretical or domain independent. We aim to explore situated planning in three contrasting domains, giving us wide view of how the theory of situated planning can be applied. Our work so far has been focused on the grid path planning domain, which has shown some promising initial results.

References

Amiri, M.; Bandura, K.; Berger, P.; Bhardwaj, M.; Boyce, M.; Boyle, P.; Brar, C.; Burhanpurkar, M.; Chawla, P.; Chowdhury, J.; et al. 2018. The CHIME fast radio burst project: system overview. *The Astrophysical Journal*, 863(1): 48.

Cannon, J.; Rose, K.; and Ruml, W. 2014. Real-time Motion Planning with Dynamic Obstacles. *Ai Communications*, 27: 345–362.

Cashmore, M.; Coles, A.; Cserna, B.; Karpas, E.; Magazzeni, D.; and Ruml, W. 2018a. Temporal planning while the clock ticks. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28.

Cashmore, M.; Fox, M.; Long, D.; Magazzeni, D.; and Ridder, B. 2018b. Opportunistic Planning in Autonomous Underwater Missions. *IEEE Transactions on Automation Science and Engineering*, 15(2): 519–530.

Cheng, C. C.; and Ruml, W. 2019. Real-time Heuristic Search in Dynamic Environments. In *Twelfth Annual Symposium on Combinatorial Search*.

Gaines, D.; Estlin, T.; Chouinard, C.; Casta, R.; Casta, A.; Bornstein, B.; Anderson, R.; Judd, M.; Nesnas, I.; and Rabideau, G. 2006. Opportunistic Planning and Execution for Planetary Exploration.

Kiesel, S.; Burns, E.; and Ruml, W. 2015. Achieving goals quickly using real-time search: experimental results in video games. *Journal of Artificial Intelligence Research*, 54: 123–158.

Koenig, S.; and Sun, X. 2009. Comparing real-time and incremental heuristic search for real-time situated agents. *Autonomous Agents and Multi-Agent Systems*, 18(3): 313–341.

Lorenz, R. D.; and Reiss, D. 2015. Solar panel clearing events, dust devil tracks, and in-situ vortex detections on Mars. *Icarus*, 248: 162–164.

Narayanan, V.; Phillips, M.; and Likhachev, M. 2012. Anytime Safe Interval Path Planning for dynamic environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4708–4715.

Phillips, M.; and Likhachev, M. 2011. SIPP: Safe interval path planning for dynamic environments. 5628 – 5635.

Rabideau, G.; Wong, V.; Gaines, D.; Agrawal, J.; Chien, S.; Fosse, E.; and Biehl, J. 2020. Onboard automated scheduling for the mars 2020 rover.

Russell, S.; and Wefald, E. 1991. Principles of metareasoning. *Artificial intelligence*, 49(1-3): 361–395.

Shperberg, S. S.; Coles, A.; Cserna, B.; Karpas, E.; Ruml, W.; and Shimony, S. E. 2019. Allocating planning effort when actions expire. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2371–2378.

Shperberg, S. S.; Coles, A.; Karpas, E.; Ruml, W.; and Shimony, S. E. 2021. Situated Temporal Planning Using Deadline-aware Metareasoning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, 340–348.

Vansteenwegen, P.; Souffriau, W.; and Oudheusden, D. V. 2011. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1): 1–10.

Yakovlev, K.; and Andreychuk, A. 2017. Any-Angle Pathfinding for Multiple Agents Based on SIPP Algorithm. arXiv:1703.04159.

Yakovlev, K.; Andreychuk, A.; and Stern, R. 2020. Revisiting Bounded-Suboptimal Safe Interval Path Planning. In *Proceedings of the 30th International Conference on Automated Planning and Scheduling (ICAPS)*, 300–304. AAAI Press.