Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Non-Classical Planning for Robotic Applications

Scott Kiesel
University of New Hampshire
skiesel@cs.unh.edu

Abstract

For my dissertation I am focusing on non-classical planning for robotic applications. Much classical planning research relies on assumptions that do not hold in real world robotics applications. In many cases the entire world state is not known in advance and the events that occur in the future can not be known with certainty. Robots operating in the real world also need to be responsive and react to dynamic obstacles and events.

I am currently a sixth year PhD student at the University of New Hampshire. I am focusing my research on non-classical planning for robotics applications. A lot of my research has been driven from the applications side of things and highlighted some of the shortcomings of traditional planning techniques. Specifically there many assumption of classical planning do not hold when performing planning for robotics. Very rarely is the initial state fully specified or known with much certainty. Events very often do happen outside of the control of the planning agent. Actions can fail and do not always take the same amount of time for repeated execution.

Robots that take long periods to plan an entire solution on the order of minutes are much less practical in an environment with dynamic obstacles. Traditionally planning algorithms find a full solution from the current state all the way to the goal, but in many cases this can take too long. Real-Time Search is a promising area that aims to limit the amount of planning that is done before actions are executed. By doing this, obstacles that are moving toward the agent can be avoided.

Lastly, providing heuristics to guide task and motion planning is another area that I am doing research in to help speed up solving times and improve the overall quality of solutions.

This summary is criminally scarce on citations and I apologize. I hope that it is okay to refer the reader to my cited works for related work.

Open Worlds and Temporal Uncertainty

In this portion of my dissertation I have investigated two interesting classes of problems. The first revolves around the idea of Open World Planning. This is a type of planning where the entire world state is not known a priori, but instead is slowly discovered over time. As the agent moves around the world and interacts with it through its sensor data it is able to develop a more accurate picture of the world state.

In the 2013 ICAPS PlanRob paper (Kiesel et al. 2013), we examined a few domains that exhibit open worlds, but the most significant was the Search and Rescue domain. In this domain, a robot is tasked with finding injured victims inside of a building. To make this more difficult, the robot does not know the layout of the building and it also does not know the locations of any victims.

The robot must balance exploration of the building and discovery of victims against an approaching temporal deadline to return to its home base. In this work we wanted to show that while overly complicated techniques are the norm, simpler techniques also can perform quite well.

We implemented a form of Hindsight Optimization for our robot to use to search the building looking for victims. In our implementation of Hindsight Optimization, we would generate possible worlds that could exist given our current knowledge of the world. For example, we would have some concrete knowledge of the building layout given the history of our sensor data and given a very rough idea of what any building might look like we could generate random topological building layouts. Inside that topological building layout we would randomly distribute victims, or if an expected distribution was known, we would bias toward that.

Given these possible worlds, we then used a very simple domain specific solver to try to maximize reward in each world. From that, we could then rank the next action the robot could execute based on the expectation of reward that would follow. The action would then be executed, new sensor data would arrive, new world samples would be generated, planning would happen again and finally a new next action would be selected.

To show how versatile this approach is, in the 2014 ICAPS PlanRob paper (Kiesel and Ruml 2014), we shifted the focus from Open Worlds to Temporal Uncertainty. In this work we considered a simple robot assistant which could be given a variety of pick and place tasks around the house. The difficulty in this problem, was however, that there was uncertainty in the objects' locations, the duration of each action executed, the success of an action execution and also exogenous events that occurred at uncertain time points.

Again, for this problem we were able to generate a set of world samples that matched our current world knowledge, maximize reward in those samples using a simple domain specific solver and then finally choose actions based on expected reward from executing each action.

Using this simple approach we were able to demonstrate an agent with very interesting behavior. The agent was able to serialize goals based on expectation about temporal information, choose different actions based on their expectation to fail and even meet at rendezvous points with exogenous agents with uncertain arrival times.

Real-Time Search

In this portion of my dissertation I examined some new techniques in the area of Real-Time Heuristc Seach. Real-Time Search is an area that is motivated by the goal of agents that are able to react and interact in real world situations. Traditional planning algorithms search for a complete path from the current state to the goal. In many cases this can take a very long time to find and in a real world application, the result is a very expensive robot that looks like it is "frozen" or not doing anything.

An alternative to this would be Real-Time Search, which guarantees that at least one new action to execute will be provided every t time units. The most important thing to note about this is that as a result, Real-Time Search algorithms are not typically capable of finding full solutions for the common values of t (which is usually on the order of fractions of a second).

These searches are usually doing some sort of bounded computation version of the A* search algorithm. A* examines nodes in increasing f order, where f is simply the sum of the "cost so far" and the estimated "cost to go". To guarantee completeness, when there are no dead-ends, algorithms also employ a basic learning technique to update the "cost to go" estimates based on what the agent has seen so far. Several proofs exist to show that this learning can converge to the true "cost to go" values given enough trials (having the agent move from the start to the goal, then teleporting them back to the start and having them go again).

In the 2013 SoCS paper (Burns, Kiesel, and Ruml 2013) we argue that in real world applications and video games, the metric that really matters is how quickly the agent can get from their starting location to the goal location, a metric we called Goal Achievement Time. With this in mind we adopted a technique from the Suboptimal Heuristic Search literature where the error in the estimate of the "cost to go" is learned and corrected online during search.

We also make the observation that the main motivation behind Real-Time Search is to provide responsive agents. Traditionally, a search algorithm will do a bounded amount of computation, emit a series of actions for execution, and then do another bounded amount of computation and emit more actions. What we point out and leverage to great benefit is that by emitting a series of actions, an algorithm actually allocates itself extra planning time. Consider a real-time bound of 1 millisecond. If 10 actions are emitted after the first iteration, technically, the planner has 10 milliseconds to plan instead of just 1 millisecond. Using this idea, we create dynamic time bounds online based on how long until the next deadline and greatly improve the performance of our algorithms.

Research that has not yet been done but is planned for this area of my dissertation is to combine these ideas with a previous Real-Time algorithm called Partitioned Learning Real-Time A* (PLRTA*). The motivation behind PLRTA* was to be an online real-time algorithm that was able to avoid dynamic obstacles which other previous algorithms had trouble doing in many situations. Combining our dynamic bounding and online heuristic correction with PLRTA* would provide a very powerful algorithm that was capable of using its time effectively, operating in dynamic environments and minimizing goal achievement time.

Heuristics For Motion Planning

The last portion of my dissertation is on heuristics for motion planning. Motion planning finds itself as a module in almost every robotic system. As such, it is important that motion planners are able to spend their time focusing on the important pieces of the search space expected to contain good solutions.

In the 2012 SoCS extended abstract (Kiesel, Burns, and Ruml 2012) I used Heuristic Search as an abstraction to bias sampling based motion planners' exploration. By identifying large areas of the search space to focus sampling and other areas to (mostly) ignore, we were able to get much better performance compared to uniform random sampling and existing simple sampling biases such as goal biasing.

There is interesting research to be done in this area focused on combining other Heuristic Search techniques with sampling based motion planning. Some areas of promise are using learning techniques from Real-Time Search to help update edge costs in the abstract space. Also, using Anytime Heuristic Search to focus the initial search effort on finding a solution path very quickly and then refining it over time to approach the optimal solution.

References

Burns, E.; Kiesel, S.; and Ruml, W. 2013. Experimental real-time heuristic search results in a video game. In *Sixth Annual Symposium on Combinatorial Search (SoCS-13)*.

Kiesel, S., and Ruml, W. 2014. Planning under temporal uncertainty using hindsight optimization. In *24th International Conference on Automated Planning and Scheduling: Planning and Robotics Workshop*.

Kiesel, S.; Burns, E.; Ruml, W.; Benton, J.; and Kreimendahl, F. 2013. Open world planning via hindsight optimization. In *23rd International Conference on Automated Planning and Scheduling: Planning and Robotics Workshop*.

Kiesel, S.; Burns, E.; and Ruml, W. 2012. Abstraction-guided sampling for motion planning (extended abstract). In *Fifth Annual Symposium on Combinatorial Search (SoCS-12)*.