Research Statement

Maxim Likhachev
maximl@seas.upenn.edu

Autonomous robots rely heavily on their ability to plan actions quickly, under massive uncertainty and with a large number of factors that affect execution. Graph searches such as Dijkstra's search, Breadth-First Search and A* [13] are highly popular means of finding least-cost plans due to their generality, solid theoretical ground and simplicity in the implementation. Unfortunately, these methods can rarely be used to solve complex planning problems on real robotic systems because they don't provide real-time performance guarantees, are limited to small (i.e., low-dimensional) problems and can not deal with problems that involve uncertainty. My research is driven by developing planning methods that are general, easy-to-use, theoretically well-grounded but at the same time meet the requirements of real robotic systems. In particular, I develop novel graph search algorithms that solve complex and/or large-scale problems in robotics in real-time while maintaining all the positive properties of graph searches. I use these algorithms to build planning modules for real robotic systems that perform challenging tasks.

For example, together with my colleagues, I have developed a series of novel versions of A* search that are designed specifically for planning in partially-known and dynamic environments under severe time constraints. For instance, ARA* [7] - an anytime version of A* - finds an initial, possibly highly suboptimal, solution quickly, and then improves it as time allows. Given enough time, it converges to a provably optimal solution. ARA* is highly efficient because it runs a series of suboptimal A* searches where each subsequent search reuses previous search efforts as much as possible. For each returned solution, it provides a suboptimality bound which allows its users to judge the quality of the solution and decide when to stop the improvement. LPA* [2], and its extension to a moving agent, D* Lite [1], are incremental versions of A* search. These algorithms avoid planning from scratch every time the agent needs to re-plan, which is done frequently in partially-known and dynamic environments. Instead, they solve these repeated planning problems much faster by reusing as much of the search results from previous search efforts as possible. Compared with A*, these algorithms are equipped with the same optimality guarantees and similar worst-case computational complexity bounds, but can be orders of magnitude faster. Finally, Anytime D* [5, 6], is a version of A* that both plans in anytime fashion and reuses as much of its previous planning efforts as possible. To the best of my knowledge, currently this is the only heuristic search that is both anytime and incremental.

These algorithms were used to solve a wide range of challenging planning problems including motion planning for a high-dimensional robot kinematic arm (Figure 1(b)) and generating dynamically constrained trajectories for various outdoor vehicles (Figure 1(d-f)) including a full-size SUV that has won the DARPA Urban Challenge race in 2007 [4]. One of the most recent algorithms we have developed, R* search [10], has been used to perform 8-dimensional planning for a quadrupled robot (Figure 1(c)). The algorithm was able to scale to the high-dimensionality of the problem by introducing randomization into A* search. Partially due to the success, wide applicability and strong theoretical ground of our algorithms, my students and I have recently been given funding to create a standardized open-source library of some of these planners [3]. In addition, we have also been given funding to develop new search-based planners that could be used effectively for the motion planning of high degree-of-freedom household robots (Figure 1(g))[1]. Similarly, my students and I have been recently funded to develop a search-based motion planner for unmanned

[1]funded by Willow Garage

helicopters flying autonomously through urban environments[2].

To deal with uncertainty, together with my colleagues, I have developed several planning under uncertainty algorithms [9, 8, 12]. In general, planning under uncertainty corresponds to solving POMDPs (Partially Observable Markov Decision Processes) and is known to be intractable. However, many of these problems exhibit special properties that we can capitalize on. For example, in some of these problems, one can clearly name in advance the best (preferred) outcomes of actions that involve uncertainty. In [9] we use this property to show how these seemingly very difficult and high-dimensional planning under uncertainty problems can be solved by running a series of simple and fast low-dimensional A*-like searches. The developed algorithm, called PPCP (Probabilistic Planning with Clear Preferences), is fast, scales to very large problems, and can guarantee optimality under certain conditions. The problems it solves contain billions of states; this is several magnitudes more than what current preeminent POMDP solvers can handle.

Our algorithms for planning under uncertainty were used to solve large scale problems such as traversing large environments with possible adversaries [11] (Figure 1(a)). My students and I are now starting a new project that will address the problem of autonomous landing in unmanned helicopters[3]. On the planning side, the problem is how to utilize various sensors, where to fly and what to sense in order to guarantee a safe (without crashing) landing while minimizing the time and cost of flight. This problem translates into a challenging POMDP planning problem that also exhibits clear preferences and can therefore be solved with PPCP.

My plan for the future is to push forward the use of graph search-based techniques for solving challenging planning problems in robotics. These techniques are general, theoretically well-grounded and easy-to-use. These properties are important, especially in robotics, where systems are often very hard to analyze and involve a large number of modules. For example, in addition to all of the above-mentioned projects that we are currently working on, my students are also working on developing decentralized graph search techniques (Figure 1(i))[4]. We hope that in addition to the efficiency of these searches, we will be able to provide them with the rigorous guarantees on the suboptimality of the plans they generate for teams of tightly coupled robots.

In addition to graph search-based planning, I am interested in investigating a number of other directions of research in planning. For example, I believe that current use of learning in planning is by far underexplored. Similarly, to provide a robust operation of robotic systems, we need a tighter coupling between planning and perception modules, which nowadays are almost always completely separated modules. A tighter coupling between the two can allow for the planner to produce more robust plans and for the perception module to schedule its resources more intelligently.

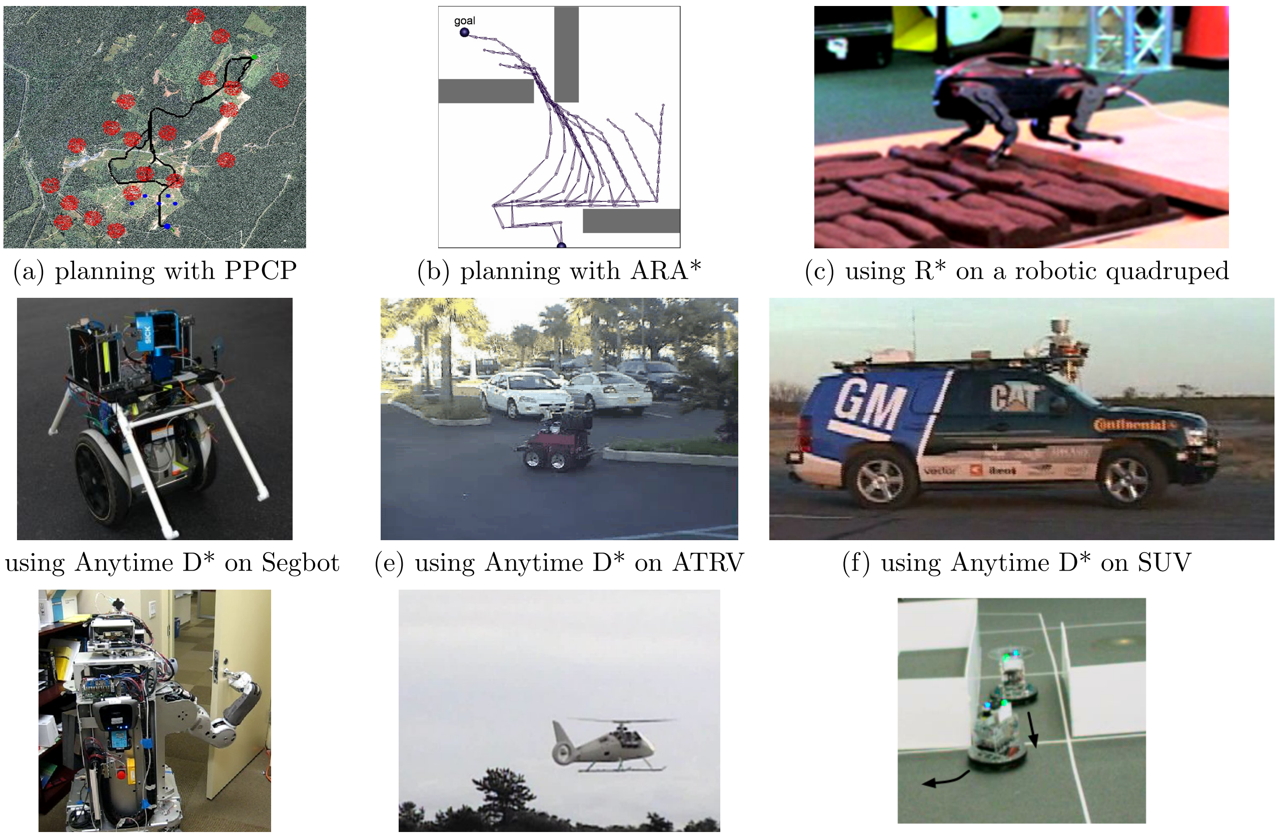
Robotics is an exciting area of research. On one hand, it presents a variety of very challenging problems. These problems can be formulated into well-defined abstract problems. The solutions to these problems can therefore generalize beyond robotics, advancing the field of Computer Science in general. On the other hand, it is always inspiring to see an algorithm running successfully on a real robot. While achieving this is often frustrating and highly time-consuming, the result always justifies the research well and motivates me to develop new algorithms that can solve other challenging problems.

---

[2]funded by Dragonfly Pictures Inc., U.S. Army's Telemedicine and Advanced Technology Research Center (TATRC), and DARPA

[3]funded by Dragonfly Pictures Inc. and U.S. Army's Telemedicine and Advanced Technology Research Center (TATRC)

[4]funded under MAST program (ARL)



(a) planning with PPCP (b) planning with ARA* (c) using R* on a robotic quadruped

(d) using Anytime D* on Segbot (e) using Anytime D* on ATRV (f) using Anytime D* on SUV

(g) search-based motion planning for mobile manipulation (h) search-based motion planning for unmanned helicopters (i) multi-agent control with decentralized and time-bounded graph searches

Figure 1: Some of the robotic systems I worked on in the past (a-f) and my students and I are working on now (g-i). (a) PPCP is used to construct a policy (shown in black) that minimizes the expected time of getting to the goal location for the main robot in the environment that contains a number of possible adversary locations (red circles). The policy dictates where to go for the main robot and where to go and what adversary locations to sense for the scout robots (shown as small blue dots). (b) ARA* is used to plan a motion for a 20 DOF robotic arm that gets its end-effector to the goal location. (c) R* is used to find 8-dimensional plans for the footholds of the quadrupedal robot taking into account its kinematic constraints. (d-e) Anytime D* is used to plan paths that take into account the dynamics constraints of the Segbot and ATRV robots. (f) Anytime D* is used to plan complex parking maneuvers for the CMU SUV that has won the DARPA Urban Challenge competition in 2007. (g) As part of the project currently funded by Willow Garage, my students develop search-based motion planners for a mobile manipulator. (h) As part of the project currently funded by Dragonfly Pictures/Army/DARPA, my students develop search-based motion planners for unmanned helicopters. (i) As part of the MAST program, my students develop decentralized and time-bounded graph searches used to control multi-agent robotic systems.

References

[1] S. Koenig and M. Likhachev. D* Lite. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI)*, 2002.

[2] S. Koenig, M. Likhachev, and D. Furcy. Lifelong planning A*. *Artificial Intelligence Journal (AIJ)*, 2004.

[3] M. Likhachev. *SBPL graph search library.* http://www.cis.upenn.edu/ maximl/software.html.

[4] M. Likhachev and D. Ferguson. Planning long dynamically-feasible maneuvers for autonomous vehicles. *International Journal of Robotics Research (IJRR)*, 2009.

[5] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun. Anytime dynamic A*: An anytime, replanning algorithm. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 262–271, 2005.

[6] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun. Anytime search in dynamic graphs. *Artificial Intelligence Journal (AIJ)*, 172(14), 2008.

[7] M. Likhachev, G. Gordon, and S. Thrun. ARA*: Anytime A* with provable bounds on sub-optimality. In *Advances in Neural Information Processing Systems (NIPS) 16.* Cambridge, MA: MIT Press, 2003.

[8] M. Likhachev, G. Gordon, and S. Thrun. Planning for markov decision processes with sparse stochasticity. In *Advances in Neural Information Processing Systems (NIPS) 17.* Cambridge, MA: MIT Press, 2004.

[9] M. Likhachev and A. Stentz. PPCP: Efficient probabilistic planning with clear preferences in partially-known environments. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 860–868, 2006.

[10] M. Likhachev and A. Stentz. R* search. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2008.

[11] M. Likhachev and A. Stentz. Path clearance. *IEEE Robotics and Automation Magazine, Special Issue on Cooperative Control of Multiple Heterogeneous Unmanned Aerial Vehicles for Coverage and Surveillance*, 2009.

[12] M. Likhachev and A. Stentz. Probabilistic planning with clear preferences on missing information. *Artificial Intelligence Journal (AIJ)*, 173(5-6):696–721, 2009.

[13] N. Nilsson. *Problem-Solving Methods in Artificial Intelligence.* McGraw-Hill, 1971.