Abstraction-guided Sampling for Motion Planning

Scott Kiesel and **Ethan Burns** and **Wheeler Ruml**
Department of Computer Science
University of New Hampshire
Durham, NH 03824 USA
`skiesel`, `eaburns`, `ruml` at `cs.unh.edu`

Abstract

Motion planning in continuous space is a fundamental robotics problem that has been approached from many perspectives. Rapidly-exploring Random Trees (RRTs) use sampling to efficiently traverse the continuous and high-dimensional state space. Heuristic graph search methods use lower bounds on solution cost to focus effort on portions of the space that are likely to be traversed by low-cost solutions. In this work, we bring these two ideas together in a technique called f-biasing: we use estimates of solution cost, computed as in heuristic search, to guide sparse sampling, as in RRTs. We see this new technique as strengthening the connections between motion planning in robotics and combinatorial search in artificial intelligence.

Introduction

The main contribution of this work is a new technique called f-biasing, named after the value f used by A* to order its search effort. Just as A* (Hart, Nilsson, and Raphael 1968) improves over Dijkstra's algorithm (Dijkstra 1959) by using a heuristic to focus search effort, f-biasing focuses exploration of RRT-based algorithms toward areas that are more likely to lead to the goal configuration via low cost trajectories. To use f-biasing, we first solve an abstract version of the problem. Then, using the cost estimates found in the abstract problem, we bias the location of samples in the RRT so that they are more likely to be drawn from portions of configuration space that are traversed by low cost solutions. Further details are presented by Kiesel, Burns, and Ruml (2012).

Previous Work

We begin with a discussion of related work in both heuristic search and robotics. A* (Hart, Nilsson, and Raphael 1968) is an optimal search algorithm for discrete graphs (Dechter and Pearl 1988). A* visits nodes in increasing order of estimated solution cost $f(n) = g(n) + h(n)$, where $g(n)$ is the cost of the path from the initial node to node n and $h(n)$ is the heuristic value of n, estimating the cost from n to a goal node. In this work, we are bringing the use of heuristics to the area of continuous motion planning.

Currently, some of the most powerful heuristics used by the search and AI planning communities are created using abstraction. Sturtevant and Geisberger (Sturtevant and Geisberger 2010) present an overview and a comparison of recent advances in the area of abstraction-based heuristics for grid pathfinding.

Rapidly-exploring random trees (RRTs) (LaValle 1998) grow a tree from the initial configuration toward random samples in configuration space. Each iteration of the RRT algorithm samples a random configuration, finds the nearest node in the tree, and then adds a new node to the tree by steering the nearest node toward the sample.

The RRT* algorithm (Karaman and Frazzoli 2011) is a simple modification to the RRT algorithm that allows it to find cheaper plans. Whenever a new node is added to the tree, nearby nodes are updated if they can be reached by a cheaper path via the new node. This is similar to A*, in which, whenever a cheaper path with a lower g value is found to a node, the cheaper path is kept and the other is discarded. Unlike A*, RRT* does not employ a heuristic.

Previous authors have also recognized that uniform exploration may not be efficient enough. There are a variety of previous proposals for biasing sample selection in an attempt to decrease time to first solution, improve navigation near obstacles, and increase exploration (LaValle 2006). Goal-biased sampling (Lavalle and Kuffner 2000) selects the goal configuration, or configurations near the goal, more often than uniform sampling in an attempt to grow the RRT more quickly toward the goal. This strategy can suffer in the presence of obstacles. While it was developed independently from our work, path-biasing (Vonásek et al. 2009; Krammer, Granzer, and Kastner 2011) is closely related because it can be seen as using the solution to a simplified representation of the motion planning problem such as a discrete grid or visibility graph (Nilsson 1969) whereas we use all solutions. This is a significant difference because if the simplified representation's solution does not contain a dynamically feasible solution or it is difficult to construct, the benefits are no longer apparent.

f-biased Sampling

f-biased sampling combines these three ideas: heuristic search, abstraction, and sample-based motion planning. The first step is to create an abstraction of the motion planning domain. Next, Dijkstra's algorithm is used to pre-compute the cost of the shortest path through each abstract node from the initial configuration to the goal in the abstract space. An abstract node's final values after search are its g and h-value in the abstract space. These are summed to form its associated f-value. Like a heuristic, these abstract solution costs allow RRTs to grow toward configurations that map to abstract states with low costs.

The abstraction is represented by a weighted directed graph that is small enough to be searched exhaustively with Dijkstra's algorithm. In our implementation, we use a simple uniform discretization of configuration space to create an n-dimensional grid. Each vertex in the abstract graph is a discrete configuration that represents all configurations in the continuous space that fall within its Voronoi hyper-rectangle. Adjacent vertices in the abstract graph are connected via an edge if neither vertex is obstructed by an obstacle. The weight of each edge reflects an estimate of the cost of the navigating between the two discrete configurations that it connects.

We proceed as in the standard RRT or RRT* algorithm, however, more samples are taken from configurations that correspond to low cost abstract nodes. Once an abstract node is selected, a sample from the concrete configuration space is drawn uniformly from its preimage.

Experimental Results

We evaluated the performance of f-biased RRTs experimentally on three different path planning domains; a 'straight-line vehicle', a Dubins vehicle and a hovercraft. These results are presented using the IPC Anytime Metric over 100 instances with 10 random seeds in Figure 1. Each algorithm-domain pairing was run with a specified timeout that is reported in each plot. Solution costs were collected during execution to provide the each anytime profile.

In all cases the benefits of using f-biasing dominates previous biasing techniques. In the most bottom right panel in Figure 1 it can even be seen that an f-biased RRT finds an initial solution more quickly and of better quality than the other algorithms and f-biased RRT* finds an initial solution a little more slowly but rapidly improves it over the other algorithms.

Achieving good performance with such a basic abstraction in the more complex domains suggests that f-biasing is robust to the choice of abstraction.

Conclusion

f-biasing can be used to effectively focus the growth of an RRT towards areas of configuration space that result in lower cost solutions. This bias also helps guide the RRT to solutions more quickly by disregarding irrelevant portions of the configuration space.

Acknowledgements

We would like to graciously thank NSF (grant IIS-0812141) and DARPA (grant HR0011-09-1-0021).

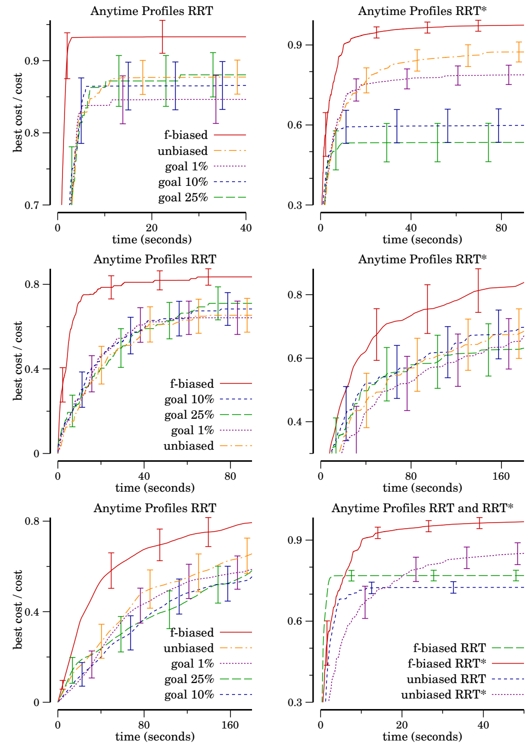

Figure 1: Top: Straight Line Vehicle; Middle: Dubins Vehicle; Bottom: Hovercraft and Straight Line Vehicle

References

Dechter, R., and Pearl, J. 1988. The optimality of A*. In Kanal, L., and Kumar, V., eds., *Search in Artificial Intelligence*. Springer-Verlag. 166–199.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1:269–271.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* SSC-4(2):100–107.

Karaman, S., and Frazzoli, E. 2011. Sampling-based algorithms for optimal motion planning. *IJRR* 30:846–894.

Kiesel, S.; Burns, E.; and Ruml, W. 2012. Abstraction-guided sampling for motion planning. Technical Report UNH-CS-12-01.

Krammer, L.; Granzer, W.; and Kastner, W. 2011. A new approach for robot motion planning using rapidly-exploring randomized trees. In *INDIN*, 263 –268.

Lavalle, S. M., and Kuffner, Jr., J. J. 2000. Rapidly-exploring random trees: Progress and prospects. In *WAFR*, 293–308.

LaValle, S. M. 1998. Rapidly-exploring random trees: A new tool for path planning. Technical report.

LaValle, S. M. 2006. *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press.

Nilsson, N. J. 1969. A mobile automaton: An application of artificial intelligence techniques. In *IJCAI*.

Sturtevant, N., and Geisberger, R. 2010. A comparison of high-level approaches for speeding up pathfinding. *AIIDE* 76–82.

Vonásek, V.; Faigl, J.; Krajník, T.; and Přeučil, L. 2009. RRT-path a guided rapidly exploring random tree. In *RoMoCo*, volume 396 of *Lecture Notes in Control and Information Sciences*. 307–316.