Lazy, Goal-Biased LBT-RRT[*][†]

Oren Salzman and **Dan Halperin**
Blavatnik School of Computer Science, Tel-Aviv University, Israel

Abstract

We recently introduced Lower Bound Tree-RRT (LBT-RRT), a single-query sampling-based algorithm that is *asymptotically near-optimal*. LBT-RRT has been shown to be an effective tool for high-quality, anytime motion planning. Similar to the roadmap maintained by RRT* or RRG which allows (asymptotically) to compute the optimal path to *any* configuration, LBT-RRT allows (asymptotically) to compute the near-optimal path to any configuration. When only a high-quality path to the goal is needed, this may lead to unnecessary time-consuming calls to the local planner. Therefore, we suggest a variant of LBT-RRT such that the roadmap maintained by the algorithm allows to extract asymptotically near-optimal paths only for nodes that are part of the goal region. We present simulation results demonstrating the effectiveness of this algorithm.

Introduction and related work

Motion planning is a fundamental research topic in robotics with applications in diverse domains such as surgical planning, computational biology, autonomous exploration, search-and-rescue, and warehouse management. In recent years there is a growing body of work on finding *high-quality* paths.

In their seminal work, Karaman and Frazzoli (Karaman and Frazzoli 2011) propose the PRM* and the RRG and RRT* algorithms all of which are shown to be *asymptotically optimal*. Namely, as the number of samples tends to infinity, the solution obtained by these algorithms converges to the optimal solution with probability one. The performance of RRT* can be improved using several heuristics (Perez et al. 2011; Hauser 2015).

An alternative approach to converge to high-quality solutions fast is to relax the asymptotic optimality of the aformentioned algorithms. LBT-RRT (Salzman and Halperin 2013; 2014) is a single-query sampling-based algorithm that is *asymptotically near-optimal*. Namely, the solution extracted from LBT-RRT converges to a solution that is within a factor of $(1 + \varepsilon)$ of the optimal solution. LBT-RRT allows for interpolating between the fast, yet sub-optimal, RRT algorithm and the typically slower, asymptotically-optimal RRG algorithm.

The asymptotic near-optimality of LBT-RRT is achieved by simultaneously maintaining two roadmaps. Both roadmaps are defined over the same set of vertices but each comprises of a different set of edges. A path in the first roadmap $\mathcal{G}_{lb}$ may not be feasible, but its cost is always a *lower bound* on the cost of paths extracted from RRG (using the same sequence of random nodes). This property is referred to as the *lower bound invariant*. A path extracted from the second roadmap $\mathcal{T}_{apx}$ is always feasible and its cost is within a factor of $(1 + \varepsilon)$ from the lower bound provided by the first roadmap. This property is referred to as the *bounded approximation invariant*.

To maintain the shortest path from the initial configuration to any node in the roadmap, LBT-RRT makes use of a dynamic *single-source shortest-path algorithm* or SSSP for short (see, e.g., (Ramalingam and Reps 1996)).

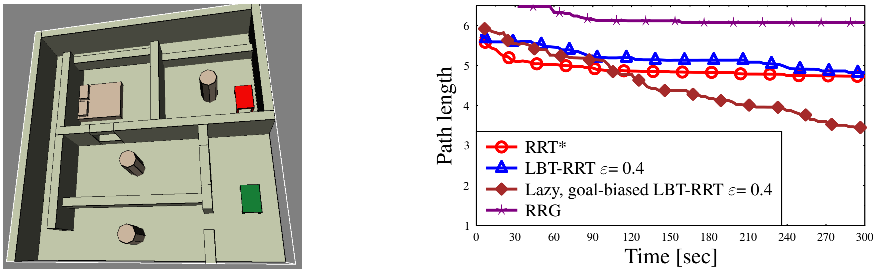
Lazy, Goal-Biased LBT-RRT

LBT-RRT maintains the bounded approximation invariant from the initial configuration to *every* node. This is desirable in settings where a high-quality path to every point in the configuration space is required. However, when only a high-quality path to the goal is needed, this may lead to unnecessary time-consuming calls to the local planner.

Therefore, we suggest the following variant of LBT-RRT, where we relax the bounded approximation invariant such that it holds only for nodes in the goal region. This variant of LBT-RRT differs with respect to the calls to the local planner and with respect to the dynamic shortest-path algorithm used. We replace the SSSP algorithm, which allows to compute the shortest paths to *every* node in a dynamic graph, with Lifelong Planning A* (LPA*) (Koenig, Likhachev, and Furcy 2004). LPA* allows to repeatedly find shortest paths from a given start to a given goal while allowing for edge insertions and deletions. Similiar to A*, this is done by using a heuristic function h that estimates the cost to reach the goal from every node.

[*] This work has been supported in part by the Israel Science Foundation (grant no. 1102/11), by the German-Israeli Foundation (grant no. 1150-82.6/2011), and by the Hermann Minkowski–Minerva Center for Geometry at Tel Aviv University.

[†] A paper titled "Asymptotically near-optimal RRT for fast, high-quality, motion planning" has originally appeared in the proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA 2014). The algorithm presented in this manuscript is part of an extended version of this ICRA paper, which is currently under review for publication in a journal.



(a) Home scenario

(b) Solution cost

Figure 1: Simulation results comparing Lazy, goal-biased LBT-RRT with RRT*, RRG and LBT-RRT. (a) Home scenario (provided by the OMPL distribution). Start and target table-shaped robots are depicted in green and red, respectively. (b) Path lengths (after applying short-cutting) as a function of computation time. Length values are normalized such that a length of one represents the length of an optimal path.

Using LPA* allows to defer calls to the local planner. It will be called only for edges that can potentially improve the cost to reach the goal (hence the name, Lazy, Goal-biased LBT-RRT). Following similar arguments as described for LBT-RRT in (Salzman and Halperin 2013), one can show that this variant is asymptotically near optimal. We note that as long as no path has been found, the algorithm performs no more calls to the local planner than RRT. Additionally, it is worth noting that the planner share some common traits with LazyRRG* (Hauser 2015) and BIT* (Gammell, Srinivasa, and Barfoot 2015).

Evaluation

We present an experimental evaluation of the performance of Lazy, Goal-Biased LBT-RRT. The algorithm was implemented using the Open Motion Planning Library (OMPL 0.10.2) (Şucan, Moll, and Kavraki 2012) and our implementation is currently distributed with the OMPL release. All experiments were run on a 2.8GHz Intel Core i7 processor with 8GB of memory.

We compared Lazy, goal-biased LBT-RRT with RRT*, RRG and LBT-RRT on the Home scenario (Fig. 1a). In this scenario, a low quality solution is typically easy to find and all algorithms (except RRG) find a solution with roughly the same success rate as RRT (results omitted). Converging to the optimal solution requires longer running times as high-quality paths pass through narrow passages. Fig. 1b depicts the (smoothed) path length obtained by the algorithms as a function of time. The convergence to the optimal solution of RRG is significantly slower than all other algorithms. Both LBT-RRT and RRT* find a low quality solution (between five and six times longer than the optimal solution) within the allowed time frame (of five minutes) and manage to slightly improve upon its cost (with RRT* obtaining slightly shorter solutions than LBT-RRT). Lazy, goal-biased LBT-RRT converges to high-quality solutions substantially faster than the alternative algorithms.

For more information, the reader is referred to our webpage http://acg.cs.tau.ac.il/projects/LBT-RRT/project-page.

Future Work

Lazy, goal-biased LBT-RRT allows to change the approximation factor between iterations. This is in contrast to LBT-RRT where such an action may possibly require a massive rewiring of $\mathcal{G}_{lb}$. Thus, one can think of the following framework: Run Lazy, goal-biased LBT-RRT with a large approximation factor (large ε), once a stopping criterion has been met, decrease the approximation factor and continue running. This may enable an even quicker convergence to find any feasible path while allowing for refinement as time permits (similar to (Alterovitz, Patil, and Derbakova 2011)).

References

Alterovitz, R.; Patil, S.; and Derbakova, A. 2011. Rapidly-exploring roadmaps: Weighing exploration vs. refinement in optimal motion planning. In *ICRA*, 3706–3712.

Şucan, I. A.; Moll, M.; and Kavraki, L. E. 2012. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine* 19(4):72–82.

Gammell, J. D.; Srinivasa, S. S.; and Barfoot, T. D. 2015. BIT*: Batch informed trees for optimal sampling-based planning via dynamic programming on implicit random geometric graphs. In *ICRA*. to appear.

Hauser, K. 2015. Lazy collision checking in asymptotically-optimal motion planning. In *ICRA*. to appear.

Karaman, S., and Frazzoli, E. 2011. Sampling-based algorithms for optimal motion planning. *I. J. Robotic Res.* 30(7):846–894.

Koenig, S.; Likhachev, M.; and Furcy, D. 2004. Lifelong planning A. *Artificial Intelligence* 155(1):93–146.

Perez, A.; Karaman, S.; Shkolnik, A.; Frazzoli, E.; Teller, S.; and Walter, M. 2011. Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms. In *IROS*, 4307–4313.

Ramalingam, G., and Reps, T. W. 1996. On the computational complexity of dynamic graph problems. *Theor. Comput. Sci.* 158(1&2):233–277.

Salzman, O., and Halperin, D. 2013. Asymptotically near-optimal RRT for fast, high-quality, motion planning. *CoRR* abs/1308.0189, abs/1308.0189.

Salzman, O., and Halperin, D. 2014. Asymptotically near-optimal RRT for fast, high-quality, motion planning. In *ICRA*, 4680–4685.