Planner Developer Tools (PDT): Reproducible Experiments and Statistical Analysis for Developing and Testing Motion Planners

Jonathan D. Gammell[1], Marlin P. Strub[2], and Valentin N. Hartmann[3]

***Abstract*— The success of sampling-based planning algorithms has made their design and evaluation a popular area of research. Evaluating different algorithms is complicated due to their use of quasirandom sampling. Experiments and analysis must be designed to calculate probabilistic performance from a finite number of individual trials. This requires careful experimental design and statistical analysis.**

Planner Developer Tools (PDT) is a C++ project to make it easier to test, evaluate, and analyze sampling-based planners across problem domains. It provides tools to evaluate Open Motion Planning Library (OMPL) algorithms fairly and also a number of abstract scenarios that isolate specific challenging aspects of the planning problem during algorithm development. It is the result of almost 10 years of development and is available open source to the sampling-based motion planning research community.

I. Introduction

Sampling-based algorithms are powerful approaches to solve motion planning problems [1–3]. They are used on different problems in a variety of disciplines, including robotics. Algorithms are often tailored to the specific nature of the target problem and their design remains a popular area of research.

An important part of designing algorithms is comparing their performance on target problems to that of existing approaches. Measuring this performance accurately and fairly is complicated by sampling-based planners' use of pseudo- or quasirandom sequences which makes performance on each individual run dependent on the specific sequences of samples. Planners instead have to be evaluated across multiple independent runs through statistical measures of performance.

A number of tools exist to facilitate using and evaluating sampling-based planners. The Open Motion Planning Library (OMPL) [4] provides high-quality reference implementations of major sampling-based planning algorithms. The associated PlannerArena and OMPLBenchmarking [5] define data formats to record experiments and planner performance in OMPL and tools to visualize the performance of different algorithms. The Benchmark for Autonomous Robot Navigation (BARN) [6] dataset focuses on evaluating the full navigation stack of ground robots with metrics to quantify problem difficulty and simulated benchmarks. Bench-MR [7] focuses on nonholonomic ground robots and includes tools for evaluating planning performance of algorithms in OMPL and other planning libraries. MotionBenchMaker [8] focuses on robot manipulation and both provides methods to generate realistic planning problems and a dataset of problems for common manipulation robots. Robowflex [9] simplifies the design and evaluation of motion planner algorithms in systems using MoveIt [10], ROS [11], or other libraries and also provides a number of tools to evaluate and visualize planners.

Planner Developer Tools (PDT) is an open-source C++ project designed to facilitate running fair, reproducible, and statistically meaningful comparisons of anytime and nonanytime OMPL planning algorithms in both single- and multiquery settings. It simplifies the evaluation of sampling-based planners with an experimental design that includes a number of best-practices for algorithm benchmarking. It also defines and implements a configuration manager that documents all planner and problem settings to support reproducible experiments.

PDT also simplifies planner evaluation by processing the raw performance measurements and performing statistical analysis (Fig 1). Measurements of independent runs are synchronized to provide data for statistics as a function of computational time. The data is analyzed using nonparametric statistical analysis and presented in an autogenerated report or can be analyzed independently with third-party tools.

PDT was designed for motion planning developers but may also be helpful for practitioners evaluating algorithms on specific problems of interest in a repeatable and statistically meaningful way. It includes a number of abstract planning scenarios that isolate specific aspects of the planning problem in n-dimensions (Fig 2) and an interface to visualize the algorithmic process on these problems in 2D. It also interfaces to problems defined in external libraries and includes an interface to the Open Robotics Automation Virtual Environment (OpenRAVE) [12] and has been used with the Robotic AI (RAI) environment (`https://github.com/MarcToussaint/rai`).

PDT has been developed while designing planning algorithms and was used for the experimental results in the associated publications [13–23]. Full details are available at `https://robotic-esp.com/code/pdt/`.

[1]Estimation, Search, and Planning (ESP) Research Group, Oxford Robotics Institute (ORI), University of Oxford, UK. `gammell@robots.ox.ac.uk`

[2]Jet Propulsion Laboratory (JPL), California Institute of Technology, USA. `marlin.p.strub@jpl.nasa.gov`

[3]Learning and Intelligent Systems Group, TU Berlin, Germany. `valentin.hartmann@ipvs.uni-stuttgart.de`

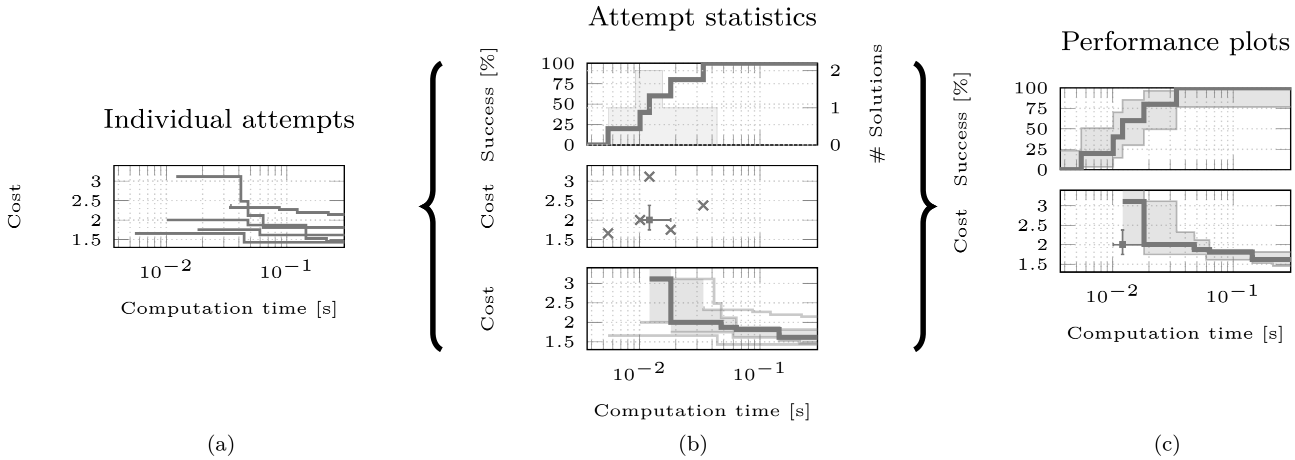

Fig. 1. An illustration of planner performance metrics extended from [21]. A planner is run multiple independent times on a given planning problem creating individual measures of solution cost as a function of computational time (a). The resulting statistics of these experiments are visualized as a histogram of initial solution times and the associated empirical distribution function (b, top), the initial solution cost versus computational time and the associated median (b, middle), and the median solution cost as a function of computational time (b, bottom). These statistics can be summarized for publication as the percent of runs solved as a function of time (c, top) and the median initial solution cost and the median solution cost as a function of time (c, bottom). These statistical measures include both successful and unsuccessful trials and have appropriate nonparametric confidence intervals that make no assumptions about the distribution of results. Note that calculating the median solution cost as a function of time requires interpolating individual measurements to common times.

II. Planner Developer Tools (PDT)

PDT is middleware that sits between OMPL and the planning problem of interest (e.g., simulator, etc.) to analyze planner performance as a function of computational time in a repeatable and reproducible way. It does this by providing:

1) Configuration management for repeatable and reproducible experiments (Section II-A).
2) Experimental design to benchmark nondeterministic planners fairly on single- and multiquery problems (Section II-B).
3) Data processing of asynchronous measurements from individual trials and calculation of nonparametric statistical measures of performance as a function of computational time (Section II-C).
4) Abstract planning scenarios with visualization and the ability to interface with OpenRAVE and other simulators or libraries (Section II-D).

PDT also automatically generates reports that fully document the tested planners and problem settings and the resulting statistical performance in publication-ready plots using LaTeX 2_ε and PGF/TikZ.

A. Configuration Management

An important part of repeatable and reproducible experiments is clear and complete documentation of experimental settings. PDT uses a JSON-based configuration manager to define the settings for the planner, problem, and all other aspects of OMPL (e.g., collision detection resolution, etc.). The configuration manager provides defaults for supported planners and tracks every configuration query and documents all accessed key-value pairs. This separates default settings from the source code and provides complete documentation of the experiment, including settings that may be unknown to the end user. It can even document the specific pseudorandom seed for the random number generator when used with the appropriate publicly available branch of OMPL.

B. Experimental Design

Timing planning algorithms in a fair and meaningful way on modern consumer computers is difficult. PDT provides an experimental design for both single- and multiquery planning problems. It includes fixes to timing irregularities previously encountered in standard libraries and best-practices such as randomizing the order of planners and timing planner construction and setup. This both makes it easier for future researchers to gather meaningful comparisons and evaluations and can serve as a repository of best practices in the motion planning field.

C. Statistical Analysis

Planner performance is quantified by statistical measures of solution time and cost and PDT facilitates these calculations for both anytime and nonanytime planners in single- and multiquery problems. Statistical measures as a function of computational time require synchronizing the measurements of each independent run to common times by interpolating between the two closest measurements. These processed data are used to calculate percent of trials solved, median initial solution costs, median initial solution times, and median solution cost as a function of computational time for each planner with appropriate nonparametric confidence intervals (Fig. 1). Medians allow these metrics to include unsolved trials in a meaningful way by treating these times and/or costs as infinite. Multiquery performance is evaluated for both individual queries and as a function of the planning query.

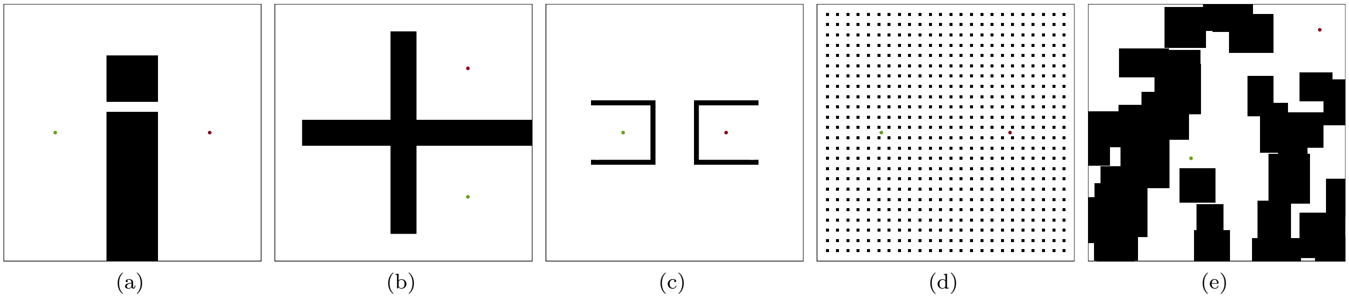

Fig. 2. An illustration of some abstract planning scenarios provided by PDT, with obstacles in black and the start and goal states as green and red dots, respectively. The scenarios include two-homotopy classes with a narrow gap (a), a multiroom setting with a circuitous solution (b), start and goal enclosures (c), regularly repeating obstacle creating many homotopy classes (d), and randomly generated obstacles (e). All scenarios are extendable to n-dimensions and are defined by configurable parameters.

D. Abstract Problems and OpenRAVE

The general planning problem is challenging in many unique ways. PDT provides simple abstract scenarios that isolate a number of the specific challenges of planning, including narrow passages, enclosures, and multiple homotopy classes (Fig. 2). These simple scenarios can be used to understand how a planning algorithm performs on specific challenges through statistical analysis or by using the provided interface to visualize the incremental search process. PDT can also provide statistical analysis of planning problems defined in external libraries, including OpenRAVE through the provided interface, and has been previously used with RAI.

III. Conclusion

PDT is an open-source C++ project to facilitate fair and reproducible evaluations of anytime and nonanytime OMPL planning algorithms. It provides tools to simplify experimental design, manage problem and planner configuration, calculate meaningful nonparametric statistical measures of performance, generate publication-ready figures, and visualize the planning process. It can be used with the provided abstract scenarios that isolate specific aspects of the planning problem and on problems defined in OpenRAVE or other external libraries. More details are available at `https://robotic-esp.com/code/pdt/`.

Acknowledgement

This work has been supported indirectly by the project funding of individuals working on its development, including from the Natural Sciences and Engineering Research Council of Canada (NSERC) through the NSERC Canadian Field Robotics Network (NCFRN), the Ontario Ministry of Research and Innovation's Early Researcher Award Program, UK Research and Innovation and EPSRC through Robotics and Artificial Intelligence for Nuclear (RAIN) [EP/R026084/1] and ACE-OPS: From Autonomy to Cognitive assistance in Emergency OPerationS [EP/S030832/1], and the German Research Foundation's (DFG) Integrative Computational Design and Construction for Architecture Cluster of Excellence (EXC 2120).

References

[1] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, ser. Intelligent Robotics and Autonomous Agents. MIT Press, May 2005.

[2] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, July 2006.

[3] J. D. Gammell and M. P. Strub, "Asymptotically optimal sampling-based motion planning methods," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 295–318, May 2021.

[4] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, Dec. 2012, library available from https://ompl.kavrakilab.org/.

[5] M. Moll, I. A. Şucan, and L. E. Kavraki, "Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 96–102, Sept. 2015.

[6] D. Perille, A. Truong, X. Xiao, and P. Stone, "Benchmarking metric ground navigation," in *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 4–6 Nov. 2020, pp. 116–121.

[7] E. Heiden, L. Palmieri, L. Bruns, K. O. Arras, G. S. Sukhatme, and S. Koenig, "Bench-MR: A motion planning benchmark for wheeled mobile robots," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 3, pp. 4536–4543, July 2021.

[8] C. Chamzas, C. Quintero-Peña, Z. Kingston, A. Orthey, D. Rakita, M. Gleicher, M. Toussaint, and L. E. Kavraki, "MotionBenchMaker: A tool to generate and benchmark motion planning datasets," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 882–889, Apr. 2022.

[9] Z. Kingston and L. E. Kavraki, "Robowflex: Robot motion planning with moveit made easy," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 23–27 Oct. 2022.

[10] D. Coleman, I. A. Şucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a MoveIt! case study," *Journal of Software Engineering for Robotics*, vol. 5, no. 1, pp. 3–16, May 2014.

[11] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *Proceedings of the Workshop on Open Source Robotics, IEEE International Conference on Robotics and Automation (ICRA)*, May 2009.

[12] R. Diankov, "Automated construction of robotic manip-

ulation programs," Ph.D. dissertation, Carnegie Mellon University (CMU), Sept. 2010.

[13] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 14–18 Sept. 2014, pp. 2997–3004.

[14] ——, "Batch Informed Trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 26–30 May 2015, pp. 3067–3074.

[15] S. Choudhury, J. D. Gammell, T. D. Barfoot, S. S. Srinivasa, and S. Scherer, "Regionally Accelerated Batch Informed Trees (RABIT*): A framework to integrate local information into optimal path planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 16–21 May 2016, pp. 4207–4214.

[16] J. D. Gammell, "Informed anytime search for continuous planning problems," Ph.D. dissertation, University of Toronto, Feb. 2017.

[17] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Informed sampling for asymptotically optimal path planning," *IEEE Transactions on Robotics (T-RO)*, vol. 34, no. 4, pp. 966–984, Aug. 2018.

[18] ——, "Batch Informed Trees (BIT*): Informed asymptotically optimal anytime search," *The International Journal of Robotics Research (IJRR)*, vol. 39, no. 5, pp. 543–567, Apr. 2020.

[19] M. P. Strub and J. D. Gammell, "Advanced BIT* (ABIT*): Sampling-based planning with advanced graph-search techniques," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 31 May – 31 Aug. 2020, pp. 130–136.

[20] ——, "Adaptively Informed Trees (AIT*): Fast asymptotically optimal path planning through adaptive heuristics," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 31 May – 31 Aug. 2020, pp. 3191–3198.

[21] M. P. Strub, "Leveraging multiple sources of information to search continuous spaces," Ph.D. dissertation, University of Oxford, 2021.

[22] M. P. Strub and J. D. Gammell, "Adaptively Informed Trees (AIT*) and Effort Informed Trees (EIT*): Asymmetric bidirectional sampling-based path planning," *The International Journal of Robotics Research (IJRR)*, vol. 41, no. 4, pp. 390–417, Apr. 2022.

[23] V. N. Hartmann, M. P. Strub, M. Toussaint, and J. D. Gammell, "Effort Informed Roadmaps (EIRM*): Efficient asymptotically optimal multiquery planning by actively reusing validation effort," in *Proceedings of the International Symposium on Robotics Research (ISRR)*, 25–30 Sept. 2022.