

Resolution-Exact Algorithms for Link Robots

Yi-Jen Chiang^{1*} Zhongdi Luo^{2†} Chee Yap^{2‡}

¹ Polytechnic Institute of NYU

² New York University

Algorithmic motion planning has had a 30-year run in which various approaches and theories have competed. Divergent paths have been taken by practical robotics and theoretical motion planners. There are three main approaches to algorithmic motion planning: exact, sampling and subdivision approaches [9]. The exact approaches have been developed by Computational Geometers [6] and in computer algebra [2]. However, their correct implementation is highly non-trivial because of numerical errors. Although an approach known as Exact Geometric Computation (EGC) can lead to correct implementation [5], it is still highly complex to implement and expensive to compute. According to [13], there is no known good implementation of exact planners for more than 3 degrees of freedom (DOF). The sampling approach includes the famous PRM [7] framework and its many variants [10]. This is currently the dominant paradigm among roboticists. The subdivision approach is one of the earliest approaches to motion planning [3]. Recently, we have revisited this approach from a theoretical standpoint [11, 12]. What is new is the introduction of *soft predicates* and *resolution-exactness* [11, 12]: taken together, we can completely avoid exact computation. They lead to new classes of practical and theoretically sound motion planning algorithms. They seem to recover all the practical advantages of the PRM framework, but provide much stronger theoretical guarantees. In this paper we continue this line of research.

In the robotics community, an informal measure of the practical success of any method is whether it “can solve” motion planning for various canonical robots. Since we currently lack appropriate complexity analysis, what is meant by “can solve” is that these methods terminate in reasonable time on judiciously chosen input obstacle environments. This is a reasonable way to try to understand the limits and applicability of these methods. Canonical robots are first classified by the dimensionality of the physical space (i.e., planar or spatial), and then by their degrees of freedom (DOF). Choset et al. [4] pointed out that sampling methods “can solve” (in the above sense) robots with medium to high DOFs; these are out of reach for exact methods. Subdivision methods are said to reach medium DOFs (say 4-10 DOFs). In

particular, they noted that a certain link robot [8] with 9 DOFs can only be solved by sampling methods.

The three approaches (sampling, subdivision and exact) provide increasing strength in their algorithmic guarantees. So the above empirical observations about their relative abilities is not surprising. Barring other issues, we should try to use the strongest algorithmic method available to solve motion planning problems for a given robot. For instance, existing exact techniques can solve for planar disc robots very efficiently, even with the correct implementation of exact predicates. What we argue for subdivision methods¹ is that (1) it is better fit for the requirements of robotics than exact methods, and (2) it avoids the halting problem of sampling methods [11]. Hence our interest in developing the subdivision methods. We believe such methods “can solve” considerably higher DOFs than is often suggested. This cannot be done in simplistic ways: certainly we cannot afford to use a tree whose size is exponential in the depth d (cf. [1]). Our adaptive framework avoids this. Moreover, it is critical that the rotational DOFs be given a different treatment from translational DOFs. This paper makes a contribution of new techniques to this end.

We follow the general framework of subdivision motion planning in [11, 12], where we draw attention to the roles of soft predicates and global search strategies. The present paper focuses on the 2-link planar robots with 4 DOFs. We make three main contributions:

(A) Soft predicates for 2-link robots. As envisioned in [11], soft predicates can exploit a wide variety of techniques that trade-off ease of implementation against efficiency. Here we introduce the notion of **length-limited forbidden angles** for link robots.

(B) A “T/R Splitting” technique based on splitting translational and rotational DOFs in different phases. Since a 2-link robot has 4 DOFs, naive subdivision would split each box into $2^4 = 16$ children. This would considerably slow down the algorithm. A natural idea [11] is to consider two regimes: boxes are originally in the “large regime” in which we only split the translational DOFs. When the boxes are sufficiently small, in the

*Email: yjc@poly.edu.

†Email: z1562@nyu.edu.

‡Email: yap@cs.nyu.edu.

¹ Properly construed. E.g., we should not use subdivision methods as another way to solve exact problems, thus relying on exact predicates.

“small regime”, we also split the angular DOFs. The obvious method to do the latter is to split every DOF – for the case of a rigid planar robot with 3 DOFs, the method of [11] splits each box in the small regime into $2^3 = 8$ children. This idea cannot scale to our 4-DOF robots. As experiments indicate, this greatly slows down the algorithm. Our new idea is to do the angular split only *once*, at the level just above the leaves. This turns out to simplify and speed up the algorithm significantly. Our new technique appears to also speed up the original 3-DOF problems (experiments to be done).

(C) Implementation of the above techniques. Combined with various search strategies, we provide experimental validation of our algorithms.

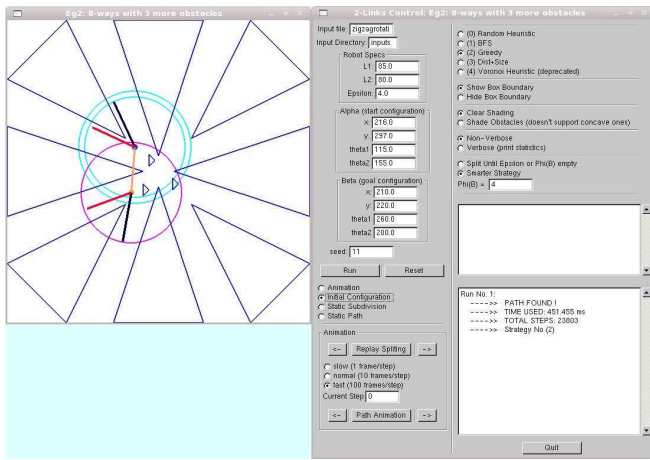


Figure 1: Initial configuration on the input “eg2”. The obstacle set has 8 big triangles forming 8-way “corridors”, plus 3 small triangles in the center. We also show the starting and ending configurations of the 2-link robot (enclosed in a single circle and double circles respectively).

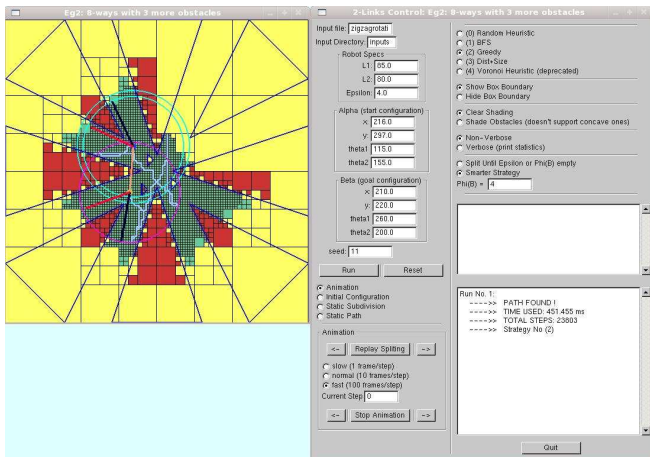


Figure 2: Final configuration after running on the input “eg2”. The resulting path is shown, and the leaf boxes obtained during the subdivision search for the path are displayed and color coded.

We have implemented in C++ our motion planner for 2-link robots. Our code and datasets are freely distributed with the **Core Library**², where various parameter settings for the experiments on some highly non-trivial instances are reproducibly encoded in Makefile targets. One such instance is shown in Figs. 1 and 2. A video clip showing the animation of the resulting path is available³. For all those highly non-trivial instances, our planner can find solution paths efficiently in real time.

References

- [1] J. Barraquand and J.-C. Latombe. Robot motion planning: a distributed representation approach. *IJRR*, 10(6):628–649, 1991.
- [2] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*. Springer, 2nd edition, 2006.
- [3] R. A. Brooks and T. Lozano-Perez. A subdivision algorithm in configuration space for findpath with rotation. In *Proc. 8th IJCAI, Vol. 2*, pp. 799–806, 1983. Morgan Kaufmann Pub. Inc.
- [4] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Boston, 2005.
- [5] D. Halperin, E. Fogel, and R. Wein. *CGAL Arrangements and Their Applications*. Springer-Verlag, Berlin, 2012.
- [6] D. Halperin, L. Kavraki, and J.-C. Latombe. Robotics. In *Handbook of Discrete and Computational Geometry*, chapter 48, pages 1065–1093. CRC Press LLC, 2nd edition, 2004.
- [7] L. Kavraki, P. Švestka, C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robotics and Automation*, 12(4):566–580, 1996.
- [8] L. E. Kavraki. *Random Networks in Configuration Space for Fast Path Planning*. PhD thesis, Stanford U., 1995.
- [9] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, 2006.
- [10] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT press, Cambridge, MA, 2005.
- [11] C. Wang, Y.-J. Chiang, and C. Yap. On Soft Predicates in Subdivision Motion Planning. In *Proc. ACM Sympos. Computational Geometry (SoCG ’13)*, pages 349–358, 2013. Full paper (invited and submitted to *Comput. Geom.: Theory Appl. (CGTA), Special Issue for SoCG ’13*): <http://cis.poly.edu/chiang/CGTA-robot.pdf>.
- [12] C. K. Yap. Soft Subdivision Search in Motion Planning. In *Proc. Robotics Challenge and Vision Workshop (RCV ’13)*, 2013. **Best Paper Award**, sponsored by Computing Community Consortium (CCC). RSS 2013. Full paper: <http://cs.nyu.edu/exact/papers/>.
- [13] L. Zhang, Y. J. Kim, and D. Manocha. Efficient cell labelling and path non-existence computation using C-obstacle query. *Int’l. J. Robotics Research*, 27(11–12), 2008.

² <http://cs.nyu.edu/exact/core/download/core/>.

³ <http://cs.nyu.edu/exact/gallery/2link/2link.html>.