Faster Algorithms for Growing Collision-Free Regions of Constrained Bimanual Configuration Spaces

Thomas Cohn, Peter Werner, and Russ Tedrake

***Abstract*— Decomposition-based motion planners (DBMPs) have shown incredible effectiveness, combining the best aspects of sampling-based planners and trajectory optimization. DBMPs first decompose the free space into collections of convex sets and then optimize trajectories contained in the union of these sets. So far, DBMPs have been unable to deal with the curved constraint manifolds that appear in constrained bimanual manipulation tasks natively. Recent work has leveraged a parametrization of the constraint manifold, induced by analytic inverse kinematics, to build collision-free convex sets on such manifolds. We combine this strategy with recent advances in collision-free convex set generation to build regions up to 500x faster. The resulting sets also have simpler descriptions, (usually) larger volume, and a lower fraction in collision.**

I. Introduction

To plan motions for robot manipulators, algorithms traditionally operate in configuration space ($\mathcal{C}$-space), where each dimension corresponds to an individual degree of freedom of the robot [1]. Given a start and goal configuration, a motion planning algorithm must find a path connecting these points in configuration space. We might also ask the algorithm to find a path which satisfies some additional constraints or minimizes some objective function.

One of the most common constraints considered in motion planning is collision avoidance. Describing the set of collision-free configurations ($\mathcal{C}$-free) is intractable in general, due to the complexity of the mapping from configuration space to task space. Leveraging the fact that an individual configuration can be efficiently checked for collisions, *sampling-based planners* use rejection sampling to build a graph contained in $\mathcal{C}$-free, and reduce the motion planning problem to graph search [2], [3]. On the other hand, *trajectory optimizers* treat the control points of the robot's path as decision variables, and cast collision avoidance as a (nonlinear, nonconvex) constraint [4]. To date, the majority of motion planning algorithms can be clearly cast into one of these two categories.

Recently, a new class of *decomposition-based motion planners* (DBMPs) have emerged; they leverage the construction of an inner approximation of $\mathcal{C}$-space as the union of (positive-measure) convex sets [5]–[10], although this concept has been considered further in the past [11]–[13]. These planners are made possible by algorithms to generate these individual convex sets [14]–[16]. The question of selecting which sequence of convex sets to move through is reminiscent of the graph search problem considered by sampling-based planners. But because the individual convex sets have an interior, there is room to optimize over the possible trajectories within the set. And finally, by the construction of these convex sets within $\mathcal{C}$-free, the collision avoidance constraint is implicitly guaranteed.

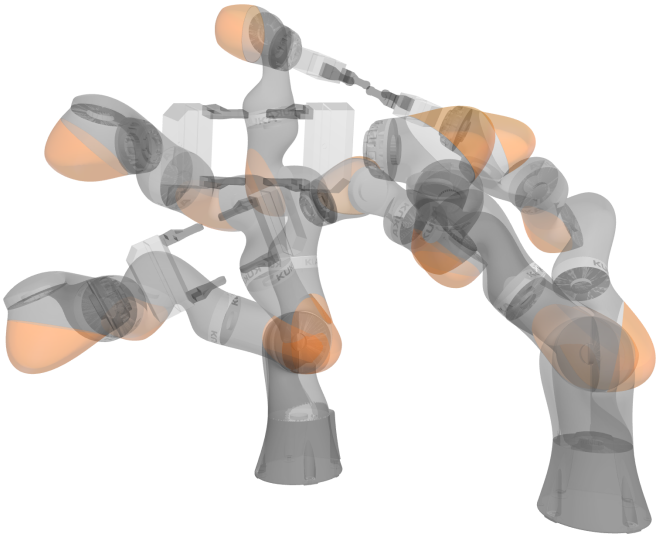

Fig. 1: Three collision-free bimanual configurations from a single convex set. The depicted bimanual configurations correspond to configurations of both arms such that they produce a fixed transformation between both end effectors and are comprised of the joint angles of the leader arm, and the self-motion parameter of the follower arm.

Constructing the requisite decomposition of $\mathcal{C}$-free is tractable because $\mathcal{C}$-free itself is positive-measure. But for constrained systems, the set of kinematically-valid configurations ($\mathcal{C}$-valid) may be a measure-zero subset of the full $\mathcal{C}$-space. We refer to this set as the *constraint manifold* (although it is only truly a manifold if we excise singularities). Sampling-based planners must use specialized techniques to draw samples [17]. Trajectory optimizers must use similar techniques [18] or accept slight constraint violations away from the control points of the trajectory [19].

Recent work, using analytic inverse kinematics as a parametrization of the constraint manifold, enabled trajectory optimization intrinsic to $\mathcal{C}$-valid [20]. The decision variables of the optimization problem described a trajectory in this minimal coordinate system, and costs and constraints were imposed by lifting the trajectory to the full configuration space. This guarantees, by construction, that the whole trajectory obeys the kinematic constraints up to floating point error. Furthermore, convex sets could be constructed through the parameterization, enabling the use of DBMPs.

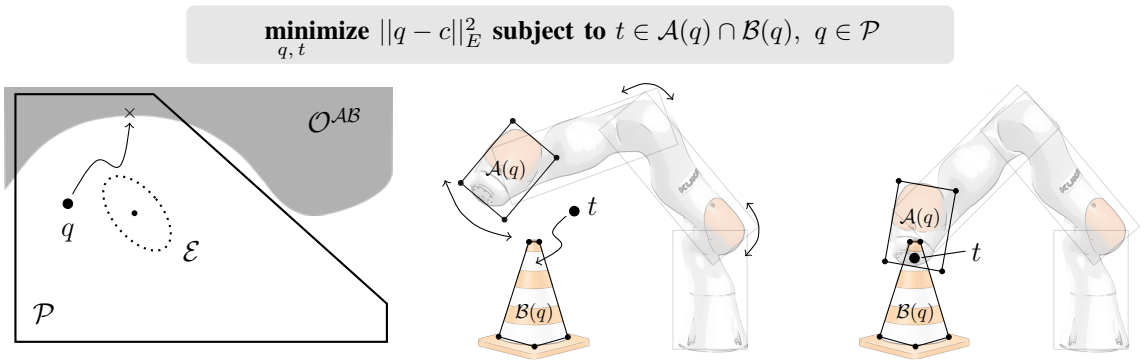

Fig. 2: Counterexample search program, reproduced from [21].

However, the success of DBMPs for constrained bimanual planning was limited by the region generation. The authors of [20] modified the IRIS-NP algorithm for producing collision-free regions in configuration space [15] to operate through the parametrization, but it was slow and produced polytopes that required many hyperplanes to describe. Furthermore, it was difficult to quantify how reliable the regions actually were, as some had significant portions that were actually in collision.

Recently, new algorithms for region generation (without kinematic equality constraints) were presented, offering significant speedups and more consistent performance [21]. In this paper, we extend these algorithms to be compatible with the parametrization machinery, and discuss a variety of engineering improvements. Altogether, we achieve speedups on the order of 100-200x and require just a fraction of the hyperplanes to describe the regions. Furthermore, the regions generated often cover a larger volume, while allowing careful control over the allowable fraction in collision. Finally, we demonstrate that these results are compatible with the *Visibility Clique Cover* algorithm (VCC) [22] for efficiently placing and shaping the convex sets in order to maximize coverage of $\mathcal{C}$-free.

II. Related Work

In the special case of convex obstacles, efficient algorithms for constructing convex collision-free sets have been known for some time [13], [23]. In the context of navigation for a mobile robot, $\mathcal{C}$-space can naturally be identified with task space by inflating obstacles by the "radius" of the robot. This has been especially in path planning for flying drones [24]. Wu et al. [25] specifically build regions along an existing collision-free path, towards producing a better trajectory.

A prominent early algorithm for convex decompositions of $\mathcal{C}$-space is the Sampling-Based Neighborhood Graph (SNG) of Yang and LaValle [11], [26], [27]. Given a point in $\mathcal{C}$-space which is collision-free, the SNG constructs a collision-free sphere of maximum radius (ellipsoids and cylinders are also possible); the radius can be determined from the kinematic Jacobian. More recently, the IRIS algorithm [14] for generating convex collision-free polytopes in $\mathcal{C}$-space was generalized to configuration space using nonlinear programming, with an algorithm called IRIS-NP [15]. However, the user of nonlinear programming, which cannot be solved to global optimality, voids the collision-free guarantees. Instead, IRIS-NP repeatedly tries to solve the nonlinear programs with different initial guesses, using a local solver like IPOPT [28] or SNOPT [29], until a sufficient number of consecutive failures have occurred, and treats this as a "probabilistic certificate".

Recently, the IRIS-NP2 and IRIS-ZO algorithms were presented [21], which provided significant improvements over IRIS-NP. Opposed to IRIS-NP, IRIS-NP2 leverages random sampling and fast collision checking to provide the nonlinear programs with informed initial guesses. IRIS-ZO bypasses the use of nonlinear programming solvers altogether, using a simple zero-order optimization strategy that is embarrassingly parallel. (Follow up work implemented IRIS-ZO on a GPU to grow regions along trajectories in $\mathcal{C}$-space [30].) Furthermore, these algorithms more precisely certify the fraction of the region in collision.

In contrast to these probabilistic certificates, it is possible to *exactly* certify a region as collision-free with polynomial optimization. The tangent half-angle substitution can be used to rewrite the kinematic equations in polynomial form [31]. From here, sums-of-squares optimization [32] can be used to certify that trajectories [33], or even entire polytopes [16], [34], are collision-free.

All of the above algorithms are focused on building individual regions, but there has been some preliminary work in how to best construct multiple regions for coverage. The VCC algorithm [22] proved significantly better than the naive approach of generating regions from randomly-chosen seeds.

III. Background

All of the "IRIS-style" algorithms have a similar high-level structure. They begin with a collision-free seed configuration and an ellipsoid centered at the seed point that is used to guide the shape of the resulting polytope. Next, they iteratively find nearby configurations that are in collision and construct a hyperplane that separates the center of the ellipsoid from the found collision until the current polytope is sufficiently collision-free. Then, the ellipsoid is updated by computing the maximum volume ellipsoid contained in

the current polytope. These steps are then repeated until a termination condition is met, such as the ellipsoid volume converging.

The original IRIS algorithm found the closest collisions for convex obstacles by solving a convex program – one per obstacle. IRIS-NP found closest collisions by solving the nonlinear program

$$\begin{aligned} \underset{q,t}{\text{minimize}} \quad & ||q-c||_E^2 \\ \text{subject to} \quad & t \in \mathcal{A}(q) \cap \mathcal{B}(q), \\ & q \in \mathcal{P}, \end{aligned} \tag{1}$$

where the current ellipsoid is given by

$$\mathcal{E} = \{x \mid (x-c)^T E(x-c) \leq 1, \ E \succ 0\}, \tag{2}$$

$\mathcal{P}$ is the current polytope expressed as a halfspace intersection, and $(\mathcal{A}(q), \mathcal{B}(q))$ indicate the collision geometries of the considered collision pair expressed in world frame at the configuration q. This setup is visualized in Fig. 2. In order to compute a polytope, this program is repeatedly solved for each geometry pair $(\mathcal{A}, \mathcal{B})$ with different initial guesses until a sufficient number of consecutive solves returned infeasible. This formulation can also be extended to arbitrary constraints of the form $g(q) \leq 0$ by searching for configurations q such that $g(q) \geq 0$.

The algorithmic improvements in IRIS-NP2 and IRIS-ZO over IRIS-NP [21] stem from finding closest collision points more efficiently by leveraging sampling and fast collision checking. IRIS-NP2 draws random samples, and uses those samples which are in collision as the initial guess for (1). These samples can be used directly (the "greedy" strategy) or the sample set can be refined using a line search strategy to find closer samples in collision (the "ray" strategy). IRIS-ZO also draws random samples, but approximately solves (1) with a simple bisection search. Finally, both algorithms used this sampling process to estimate the fraction of the region in collision, producing a rigorous probabilistic certificate.

On the other hand, the IRIS-NP algorithm has been specifically extended to generating regions for kinematically constrained systems – in particular, for constrained bimanual manipulation [20]. Given a smooth mapping $\phi : \mathcal{D} \to \mathcal{C}$ that parametrizes $\mathcal{C}$-valid, the authors generated regions by finding hyperplanes to

1) avoid function domain violations,
2) avoid joint limit violations,
3) avoid non-reachable configurations,

before finally searching across all possible collision pairs.

IV. Methodology

Suppose we have a smooth function $\phi : \mathcal{D} \to \mathcal{C}$, where $\forall \tilde{q} \in \mathcal{D}$, $\phi(q) \in \mathcal{C}$-valid, and $\mathcal{D}$ can be defined as some constrained subset of $\mathbb{R}^m$,

$$\mathcal{D} = \{\tilde{q} \in \mathbb{R}^m \mid \mathcal{D}_k \leq 0, k = 1, \ldots, K\}.$$

We use the same parametrization for the bimanual setup as [20], where ϕ takes in the joint angles for one arm, and an additional self-motion parameter, and computes the joint angles of the remaining arm with analytic IK [35]. The output of ϕ is then the angles for both arms.

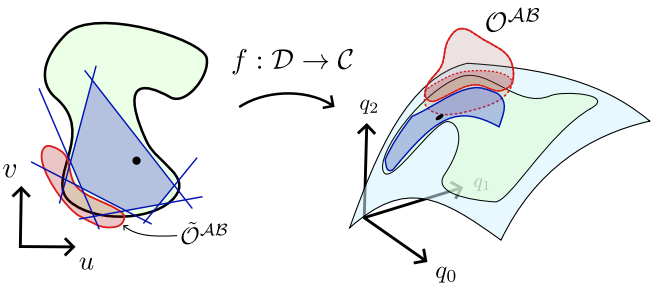

Fig. 3: The general setup for growing regions on a parametrized subspace. On the right is the full configuration space, and the light blue surface is the constraint manifold. On the left is the minimal coordinates system. The red subset is a collision region. The green region is the feasible subset, and the dark blue polytope is the region grown by our algorithm (around the seed point).

Regions are now grown in the parametrized space $\mathcal{D}$. We must specify a new initial domain – for the bimanual setup, the domain is the Cartesian product of the joint limits for the control arm with $[-\pi, \pi]$ for the self-motion angle. Besides searching for counterexamples, we enforce the domain and joint limit constraints presented in [20]. It turns out that the additional reachability constraint is unnecessary, as the domain constraints will guarantee reachability.

The bisection search of IRIS-ZO adapts very naturally to this formulation, except instead of directly checking the sample for a collision, the algorithm checks for any constraint violations before applying the parametrization to check for collisions in the full configuration space. IRIS-NP2 identifies which constraint was violated or which geometries were in collision, and then solves the appropriate nonlinear program. Since constraints are imposed in $\mathcal{D}$, Eq. (6) of [15] can be used directly. For collisions, we modify (1):

$$\begin{aligned} \underset{\tilde{q},t}{\text{minimize}} \quad & ||\tilde{q}-c||_E^2 \\ \text{subject to} \quad & t \in \mathcal{A}(\phi(\tilde{q})) \cap \mathcal{B}(\phi(\tilde{q})), \\ & \tilde{q} \in \mathcal{P}. \end{aligned} \tag{3}$$

Because $\tilde{q}$ now lives in the parametrized space, the mapping ϕ must be applied to obtain the full robot configuration, before examining the configuration-space obstacles. Because ϕ is compatible with automatic differentiation, this program can be solved with generic nonlinear optimizers like SNOPT [29] and IPOPT [28].

V. Experiments

We grow regions with IRIS-ZO and IRIS-NP2 (using both the greedy and ray sampling strategies) for the constrained bimanual setup used in [20]. We use the same 19 seed points for the fixed grasp regions, and compare. Results are shown in Table I. Compared to the specialized version of IRIS-NP presented in [20], these new algorithms are significantly more performant.

Seed	Time [s]				Number of Hyperplanes				Relative Volume				Fraction in Violation [%]			
Point	NP	ZO	Greedy	Ray	NP	ZO	Greedy	Ray	NP	ZO	Greedy	Ray	NP	ZO	Greedy	Ray
1	273.80	1.55	**1.02**	1.89	369	69	**53**	**53**	1	2.19×10^{-3}	4.23×10^{2}	$\mathbf{6.81 \times 10^{2}}$	5.87	0.00	0.01	0.29
2	360.70	1.30	**1.03**	1.49	416	62	54	**41**	1	4.36×10^{-4}	3.04×10^{2}	$\mathbf{5.03 \times 10^{2}}$	2.14	0.13	0.02	0.22
3	273.87	2.98	**1.35**	3.00	179	129	54	**52**	1	4.62	5.54×10^{3}	$\mathbf{5.81 \times 10^{3}}$	0.00	0.29	0.09	0.30
4	296.34	**3.79**	4.01	6.26	302	153	63	**56**	1	1.64×10^{-1}	1.83×10^{1}	$\mathbf{3.59 \times 10^{1}}$	0.00	0.11	0.09	0.42
5	280.35	3.14	**1.47**	2.62	370	132	70	**60**	1	2.29×10^{-1}	1.05×10^{1}	$\mathbf{1.16 \times 10^{1}}$	0.08	0.23	0.03	0.23
6	375.20	**1.17**	1.34	1.34	481	53	51	**36**	1	6.49×10^{-3}	1.91×10^{1}	$\mathbf{2.16 \times 10^{2}}$	6.31	0.01	0.16	0.55
7	233.16	**1.50**	2.17	1.75	185	90	57	**46**	1	1.60×10^{-1}	$\mathbf{2.42 \times 10^{2}}$	2.42×10^{2}	0.00	0.11	0.04	0.46
8	428.44	1.97	**1.61**	3.38	234	102	56	**50**	1	5.71	3.42×10^{3}	$\mathbf{3.93 \times 10^{3}}$	0.00	0.17	0.06	0.06
9	390.54	2.84	**1.32**	1.40	216	130	50	**49**	1	2.18	$\mathbf{2.25 \times 10^{3}}$	1.73×10^{3}	0.00	0.32	0.06	0.32
10	353.06	**1.29**	1.79	2.47	579	61	62	**47**	1	0.99×10^{-2}	1.08×10^{1}	$\mathbf{1.26 \times 10^{1}}$	3.09	0.03	0.06	0.39
11	300.84	**0.74**	1.65	1.89	436	**38**	54	39	1	6.72×10^{-3}	3.96×10^{2}	$\mathbf{4.61 \times 10^{2}}$	3.59	0.14	0.03	0.31
12	295.49	**1.22**	1.33	1.81	396	66	53	**46**	1	5.03×10^{-3}	$\mathbf{2.48 \times 10^{2}}$	1.89×10^{2}	3.79	0.09	0.03	0.09
13	433.61	**1.18**	1.27	1.73	420	51	57	**50**	1	5.66×10^{-3}	$\mathbf{2.82 \times 10^{1}}$	2.30×10^{1}	2.49	0.04	0.04	0.12
14	254.72	**1.34**	2.03	3.71	454	72	56	**53**	1	9.21×10^{-2}	2.62×10^{1}	$\mathbf{4.30 \times 10^{1}}$	0.35	0.15	0.15	0.47
15	243.63	1.52	**1.06**	2.06	346	64	51	**44**	1	1.04×10^{-1}	2.97×10^{4}	$\mathbf{4.47 \times 10^{4}}$	0.45	0.04	0.02	0.24
16	290.49	**1.41**	1.71	1.44	226	72	58	**42**	1	9.21×10^{-1}	2.41×10^{3}	$\mathbf{4.66 \times 10^{3}}$	0.00	0.16	0.05	0.15
17	248.97	2.14	**1.98**	2.63	252	107	61	**59**	1	1.05	2.96×10^{2}	$\mathbf{3.69 \times 10^{2}}$	0.00	0.21	0.04	0.15
18	263.98	2.22	**1.52**	2.31	299	109	63	**55**	1	1.08	4.45×10^{2}	$\mathbf{2.92 \times 10^{3}}$	0.10	0.20	0.06	0.47
19	216.20	**1.35**	1.72	1.60	383	66	52	**50**	1	4.01×10^{-2}	7.84×10^{1}	$\mathbf{9.09 \times 10^{1}}$	1.04	0.07	0.07	0.24
Min	216.20	**0.74**	1.02	1.34	179	38	50	**36**	1	3.11×10^{-3}	3.54×10^{2}	$\mathbf{3.99 \times 10^{2}}$	0.00	0.00	0.01	0.06
Mean	305.97	1.82	**1.65**	2.36	344	85	56	**48**	1	2.30×10^{-1}	0.99×10^{2}	$\mathbf{1.22 \times 10^{2}}$	1.54	0.13	0.06	0.29
Max	433.61	**3.79**	4.01	6.26	579	153	70	**60**	1	1.64×10^{-1}	3.18×10^{1}	$\mathbf{3.59 \times 10^{1}}$	6.31	0.32	0.16	0.55

TABLE I: A comparison of the original IRIS-NP algorithm with the three new approaches: IRIS-ZO, IRIS-NP2 (Greedy), and IRIS-NP2 (Ray). Relative volume is approximated by the maximum-volume inscribed ellipsoid, and fraction in collision is estimated with 10,000 samples. Across the board, the new algorithms are much faster than IRIS-NP, require fewer hyperlpanes to describe the collision-free polytopes, and satisfy the target fraction in collision of $\leq 1\%$. In addition, the IRIS-NP2 approaches yield larger volume regions.

IRIS-ZO, IRIS-NP2 (greedy), and IRIS-NP2 (ray) are consistently at least 100x faster than IRIS-NP, and sometimes more than 500x faster. The fastest algorithm was either IRIS-ZO and IRIS-NP2 (greedy), with about a 50/50 split. The new algorithms generally required $\frac{1}{2}$ to $\frac{1}{10}$ the hyperplanes of IRIS-NP, with IRIS-NP2 (ray) almost always requiring the fewest. IRIS-NP2 (ray) usually produced the largest regions, although IRIS-NP2 (greedy) sometimes made slightly larger regions. IRIS-ZO sometimes produced smaller regions than IRIS-NP, although this may be correlated with IRIS-NP regions that had large fractions in collision. IRIS-NP frequently produced regions with a large fraction in collision or in violation of constraints – eight out of nineteen regions exceeded the target of 99% valid. In contrast, all the new algorithms cleared this threshold.

We also use our new methods together with VCC. However, the feasible set is very small measure in the parametrization, so the visibility graph construction ends up being intractable. For example, the algorithm generated 8 regions to cover less than 1% of the intersection of $\mathcal{C}$-valid and $\mathcal{C}$-free in 437 seconds, but only 21 seconds were used for region generation.

VI. Discussion

In this paper, we have described how to generalize the IRIS-ZO and IRIS-NP2 algorithms to grow regions along constraint manifolds. The new algorithms significantly outperformed the specialized IRIS-NP variant of [20]. Although the engineering improvements certainly make a difference, the majority of the old algorithm's runtime was spent in the nonlinear solver (SNOPT or IPOPT), and the results regarding number of faces, volume, and fraction in collision are independent of the runtime. The significant reduction in number of faces and increase in volume makes the downstream planners more efficient, and the reduced fraction in collision makes them more reliable.

When comparing the new algorithms, the results are similar to those of [21]: IRIS-ZO was the fastest, but produced the smallest regions and had the most hyperplanes; IRIS-NP2 (ray) was the slowest, but required the fewest hyperplanes and generally produced the largest regions; IRIS-NP2 (greedy) represented something of a middle ground. Although further experiments are required to make universal conclusions about the differences between IRIS-ZO, IRIS-NP2 (greedy), and IRIS-NP2 (ray), we can easily conclude that these new approaches are significantly better than the specialized IRIS-NP algorithm used in [20].

As for VCC, region building is so fast now that the visibility graph construction is the major bottleneck. (Whereas before, region building was the bottleneck.) This problem is exacerbated by the small measure of the free space for kinematically constrained systems under the parametrization, where even drawing the samples is incredibly time-consuming. This suggests new algorithms for generating good covers of regions are a promising direction, especially in cases where the free-space volume is small.

VII. Acknowledgements

This work was supported by the Toyota Research Institute and the National Science Foundation Graduate Research Fellowship Program under Grant No. 2141064. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the reviews of the National Science Foundation.

References

[1] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE transactions on computers*, vol. 32, no. 02, pp. 108–120, 1983.

[2] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, 1998.

[3] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 2002.

[4] T. A. Howell, B. E. Jackson, and Z. Manchester, "Altro: A fast solver for constrained trajectory optimization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7674–7679.

[5] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Science robotics*, vol. 8, no. 84, p. eadf7843, 2023.

[6] T. Marcucci, P. Nobel, R. Tedrake, and S. Boyd, "Fast path planning through large collections of safe boxes," *IEEE Transactions on Robotics*, 2024.

[7] T. Cohn, M. Petersen, M. Simchowitz, and R. Tedrake, "Non-euclidean motion planning with graphs of geodesically convex sets," *The International Journal of Robotics Research*, p. 02783649241302419, 2023.

[8] S. Garg, T. Cohn, and R. Tedrake, "Planning shorter paths in graphs of convex sets by undistorting parametrized configuration spaces," *IEEE Robotics and Automation Letters*, 2025.

[9] J. Tang, Z. Mao, L. Yang, and H. Ma, "Space-time graphs of convex sets for multi-robot motion planning," *arXiv preprint arXiv:2503.00583*, 2025.

[10] T. Marcucci, M. Halm, W. Yang, D. Lee, and A. D. Marchese, "A biconvex method for minimum-time motion planning through sequences of convex sets," *arXiv preprint arXiv:2504.18978*, 2025.

[11] L. Yang and S. M. LaValle, "A framework for planning feedback motion strategies based on a random neighborhood graph," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 544–549.

[12] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *2014 IEEE-RAS international conference on humanoid robots*. IEEE, 2014, pp. 279–286.

[13] ——, "Efficient mixed-integer planning for uavs in cluttered environments," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 42–49.

[14] ——, "Computing large convex regions of obstacle-free space through semidefinite programming," in *Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2015, pp. 109–124.

[15] M. Petersen and R. Tedrake, "Growing convex collision-free regions in configuration space using nonlinear programming," *arXiv preprint arXiv:2303.14737*, 2023.

[16] H. Dai, A. Amice, P. Werner, A. Zhang, and R. Tedrake, "Certified polyhedral decompositions of collision-free configuration space," *The International Journal of Robotics Research*, vol. 43, no. 9, pp. 1322–1341, 2024.

[17] Z. Kingston, M. Moll, and L. E. Kavraki, "Sampling-based methods for motion planning with constraints," *Annual review of control, robotics, and autonomous systems*, vol. 1, no. 1, pp. 159–185, 2018.

[18] R. Bordalba, T. Schoels, L. Ros, J. M. Porta, and M. Diehl, "Direct collocation methods for trajectory optimization in constrained robotic systems," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 183–202, 2022.

[19] R. Bonalli, A. Bylard, A. Cauligi, T. Lew, and M. Pavone, "Trajectory optimization on manifolds: A theoretically-guaranteed embedded sequential convex programming approach," *arXiv preprint arXiv:1905.07654*, 2019.

[20] T. Cohn, S. Shaw, M. Simchowitz, and R. Tedrake, "Constrained bimanual planning with analytic inverse kinematics," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6935–6942.

[21] P. Werner, T. Cohn, R. H. Jiang, T. Seyde, M. Simchowitz, R. Tedrake, and D. Rus, "Faster algorithms for growing collision-free convex polytopes in robot configuration space," *arXiv preprint arXiv:2410.12649*, 2024.

[22] P. Werner, A. Amice, T. Marcucci, D. Rus, and R. Tedrake, "Approximating robot configuration spaces with few convex sets using clique covers of visibility graphs," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10 359–10 365.

[23] A. Sarmientoy, R. Murrieta-Cidz, and S. Hutchinsony, "A sample-based convex cover for rapidly finding an object in a 3-d environment," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 3486–3491.

[24] Q. Wang, Z. Wang, M. Wang, J. Ji, Z. Han, T. Wu, R. Jin, Y. Gao, C. Xu, and F. Gao, "Fast iterative region inflation for computing large 2-d/3-d convex regions of obstacle-free space," *IEEE Transactions on Robotics*, 2025.

[25] Y. Wu, I. Spasojevic, P. Chaudhari, and V. Kumar, "Towards optimizing a convex cover of collision-free space for trajectory generation," *IEEE Robotics and Automation Letters*, 2025.

[26] L. Yang and S. M. LaValle, "An improved random neighborhood graph approach," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 254–259.

[27] L. Yang and S. M. Lavalle, "The sampling-based neighborhood graph: An approach to computing and executing feedback motion strategies," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 419–432, 2004.

[28] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[29] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.

[30] P. Werner, R. Cheng, T. Stewart, R. Tedrake, and D. Rus, "Superfast configuration-space convex set computation on gpus for online motion planning," *arXiv preprint arXiv:2504.10783*, 2025.

[31] M. Raghaven and B. Roth, "Kinematic analysis of the 6r manipulator of general geometry," in *The fifth international symposium on Robotics research*, 1991, pp. 263–269.

[32] P. A. Parrilo and R. R. Thomas, *Sum of Squares: Theory and Applications*. American Mathematical Soc., 2020, vol. 77.

[33] A. Amice, P. Werner, and R. Tedrake, "Certifying bimanual rrt motion plans in a second," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 9293–9299.

[34] A. Amice, H. Dai, P. Werner, A. Zhang, and R. Tedrake, "Finding and optimizing certified, collision-free regions in configuration space for robot manipulators," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2022, pp. 328–348.

[35] C. Faria, F. Ferreira, W. Erlhagen, S. Monteiro, and E. Bicho, "Position-based kinematics for 7-dof serial manipulators with global configuration control, joint limit and singularity avoidance," *Mechanism and Machine Theory*, vol. 121, pp. 317–334, 2018.