Search-Based Spatiotemporal and Multi-Robot Motion Planning on Graphs of Space-Time Convex Sets

Jingtao Tang, Zining Mao, Lufan Yang, and Hang Ma

***Abstract*—Spatiotemporal motion planning couples geometric path selection with timing, and in Multi-Robot Motion Planning (MRMP), every planned trajectory changes the feasible space-time of the other robots. This extended abstract summarizes our recent work on a search-based Graph of Space-Time Convex Sets (ST-GCS) framework for spatiotemporal planning and MRMP. An ST-GCS lifts a spatial Graph of Convex Sets (GCS) into space-time, so a trajectory comprises a discrete path through convex sets and continuous states optimized within the selected space-time collision-free regions. We replace monolithic mixed-integer solving with Best-First Search (BFS) over path-indexed nodes, where each discrete expansion invokes a path-conditioned convex trajectory optimization, with admissible heuristics, upper-bound pruning, and dominance checks reducing the search effort. A general Exact Convex Decomposition (ECD) reserves swept trajectory occupancies for priority-based search and its windowed variants, enabling the framework to achieve substantial speedups, maintain high solution quality, and solve MRMP instances with up to 100 robots within minutes.**

I. Introduction

Spatiotemporal motion planning couples where a robot moves with when it moves: a route that is safe in static geometry may be blocked by a moving obstacle, and feasible corridors may be transient. This coupling is central to Multi-Robot Motion Planning (MRMP), where every robot trajectory becomes a time-varying obstacle for the others. Planning therefore requires a representation of continuous free space-time that can be updated as occupancies change.

A Graph of Convex Sets (GCS) provides a useful starting point by representing a convex decomposition of collision-free space as a graph $G = (V, E, \mathcal{X})$ [1], [2]. Each vertex corresponds to a convex set, and an edge connects two sets with a nonempty intersection. A motion plan combines a discrete vertex path with continuous entry and exit states; set membership and continuity yield a collision-free piecewise-linear trajectory. GCS thus couples route selection and geometric optimization in one mixed discrete–continuous model.

A Graph of Space-Time Convex Sets (ST-GCS) extends GCS by augmenting each state with time [3]. In Fig. 1, spatial convex sets are extruded over a planning horizon and subdivided to exclude dynamic-obstacle occupancies, leaving convex collision-free regions in space-time. A feasible trajectory selects a vertex path and continuous states, while convex constraints impose continuity, forward time, and component-wise velocity limits and optimize its arrival time. Dynamic obstacles and planned robot trajectories can therefore be represented as excluded swept occupancies.

The authors are with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada. {jingtao_tang, zander_mao, lufan_yang, hangma}@sfu.ca.

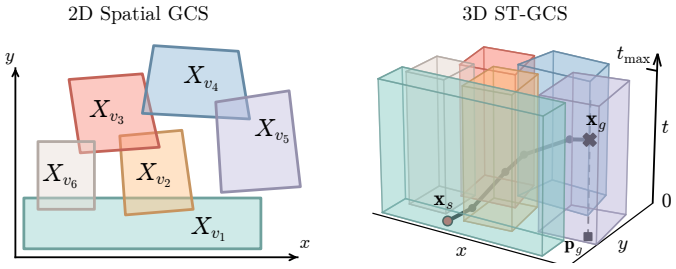


Figure 1. Extending a 2-D spatial GCS into a 3-D ST-GCS. A trajectory selects a path of space-time convex sets and continuous states from the start $\mathbf{x}_s$ to the goal position $\mathbf{p}_g$, while its arrival time is optimized.

Existing ST-GCS methods use MICP or relaxation with randomized rounding [3]; alternatives use GCS-compatible dynamic constraints, safe intervals (SIPP and Zeta*-SIPP), or continuous space-time sampling (ST-RRT*) [4]–[7]. Following search-based GCS solvers [8], [9], our framework searches complete prefix paths because arrival states and continuation costs depend on the prefix.

Related MRMP coordinators include PP, PBS, kinodynamic K-CBS, and time-discretized CB-GCS [10]–[13]. Our framework combines ST-GCS BFS with Exact Convex Decomposition (ECD) inside PP and PBS; Windowed-PP (WPP) and Windowed-PBS (WPBS) restrict reservations and conflict checking to receding horizons. This extended abstract is a short version of our full paper [14].

II. Search-Based ST-GCS Planning

This section summarizes the single-robot ST-GCS solver: path-indexed BFS separates discrete convex-set path search from continuous trajectory optimization, while heuristics, incumbents, and dominance checks control the search effort. The full paper provides the complete algorithms, optimization formulations, and bounded-suboptimality analysis [14].

A. Path-indexed Best-First Search

ST-GCS planning does not have the vertex-level optimal substructure assumed by ordinary shortest-path search. After a vertex sequence is extended, optimizing the new full sequence may change continuous states along its earlier portion. Consequently, two prefixes that reach the same vertex can admit different arrival regions and different continuation costs; retaining only the currently cheapest arrival at that vertex can discard the globally best solution.

We therefore search over path-indexed nodes. A node N stores a prefix path $N.\pi$, the optimal trajectory $N.\tau$ conditioned on that path, its cost-to-come $N.g = c(N.\tau)$, and a key

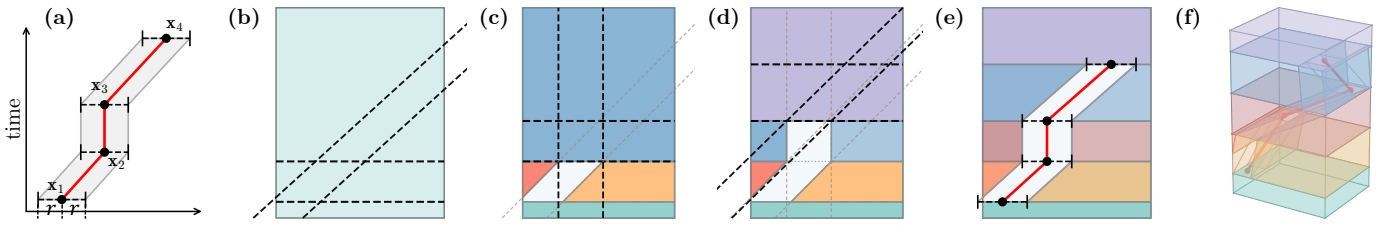


Figure 2. Exact Convex Decomposition (ECD) of a planned trajectory occupancy. (a) A piecewise-linear trajectory and its occupancy, represented by a union of parallelotopes, in 2-D space-time. (b)–(d) Separating hyperplanes progressively slice the intersected convex set. (e) The updated ST-GCS partitions the remaining collision-free region. (f) A 3-D ECD reservation.

$N.f = N.g + \epsilon h(N.\pi)$, where $\epsilon \geq 1$ is a heuristic inflation factor. The OPEN queue returns the node with minimum key. For every adjacent convex set not already in the prefix, the solver extends $N.\pi$ and invokes PATHOPTIMIZE on the entire extended sequence. With the discrete sequence fixed, this subproblem is a convex trajectory optimization over the set entry and exit states. Thus, the framework decouples discrete path exploration from continuous optimization without fixing the geometry or timing of the final trajectory in advance.

This decomposition replaces one large MICP with a sequence of smaller convex programs. It also gives a transparent guarantee: with an admissible heuristic, a valid feasible incumbent, and a safe dominance check, the first returned solution is ϵ-optimal; setting $\epsilon = 1$ recovers cost optimality. This path-indexed construction adapts recent search-based GCS ideas [8], [9] to the non-Markov structure of ST-GCS.

Upper-bound pruning is tied to a feasible incumbent rather than to a lower-bound estimate alone. A fast auxiliary vertex-indexed search supplies that trajectory; although duplicate pruning prevents it from serving as an optimal ST-GCS solver, every incumbent it returns is validated by the same path-conditioned continuous optimization. The path-indexed search can therefore discard a node whose key cannot improve the incumbent and still return a feasible solution if the remaining queue is exhausted.

B. Heuristics and Dominance Checks

The search is accelerated by complementary guidance and pruning components; their optimization details are omitted here. First, a fast vertex-indexed search supplies a feasible incumbent cost for upper-bound pruning. Second, three admissible heuristics relax different parts of the ST-GCS problem: a motion-only bound ignores graph and set constraints, a triplet-relaxation bound composes local convex transition costs, and an interface-to-set table reuses stronger costs precomputed on the base graph. Their pointwise maximum remains admissible and gives the strongest available estimate for each node.

Third, pairwise dominance checks compare path-indexed nodes that reach the same convex set. The safe set-containment check prunes only when a retained prefix can match every relevant arrival state of the candidate with no greater cost. Two lighter heuristic checks compare either the optimized arrival states or arrival times at sampled spatial positions. These checks trade formal optimality guarantees for stronger empirical pruning. Together, heuristic guidance limits which prefixes are expanded, while dominance and upper-bound checks remove prefixes that need not be explored further.

III. ST-GCS FOR MRMP

This section lifts the single-robot solver to MRMP: ECD converts trajectories into reusable reservations, PP and PBS select which reservations constrain each low-level solve, and windowing limits coordination to relevant time intervals. The full paper provides the ECD and coordination algorithms, well-formedness conditions, and completeness analyses [14].

A. Exact Convex Decomposition

Exact Convex Decomposition (ECD) turns a planned trajectory into a reusable space-time reservation. For a finite-radius robot, its piecewise-linear swept occupancy is conservatively represented by a union of convex parallelotopes. Subtracting this occupancy from a convex ST-GCS set generally leaves a nonconvex region. As illustrated in Fig. 2, ECD slices each intersected set with hyperplanes induced by the occupancy facets, removes the occupied pieces, and rebuilds adjacency among retained pieces with nonempty intersections. The resulting graph is an exact decomposition of the remaining free space relative to the polyhedral occupancy model and requires no fixed temporal grid. For MRMP, the reservation clearance equals the sum of the robot radii; the same graph-update operation handles known moving obstacles and planned robots.

B. Prioritized Planning and Priority-Based Search

Prioritized Planning (PP) fixes a total robot order $\boldsymbol{\sigma} = \langle \sigma_1, \ldots, \sigma_n \rangle$ and maintains a progressively reserved ST-GCS [10]. Robot σ_k is planned with BFS after ECD reserves the endpoint-augmented trajectories of all earlier robots. The augmentation extends each trajectory by waiting at its start before departure and at its goal after arrival, preventing unmodeled collisions during those intervals. If a low-level call fails, PP fails for that order; otherwise, the new trajectory is reserved for every later robot.

Priority-Based Search (PBS) avoids committing to one total order by searching over acyclic partial priority relations [11]. A node N stores a relation $\prec_N$ and the current robot trajectories; the root contains no relations and independently planned trajectories. For robot j, the low-level graph reserves only the endpoint-augmented trajectories in its transitive higher-priority

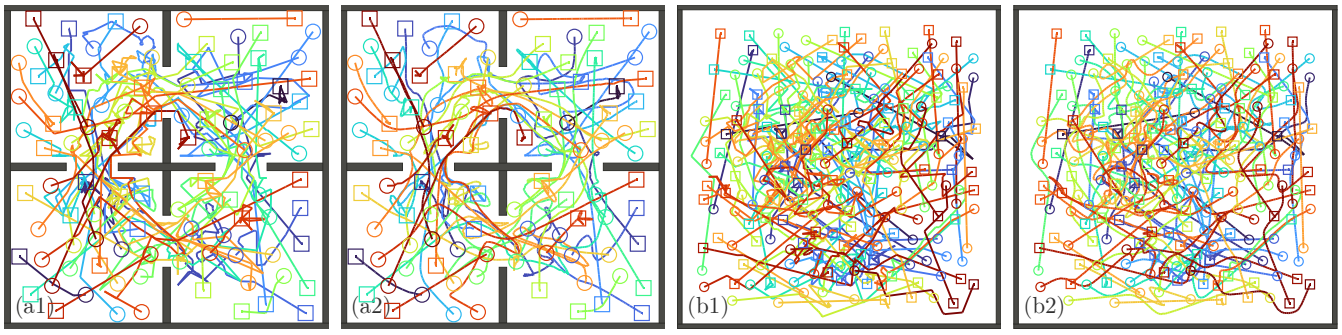


Figure 3. Large-scale planar MRMP with random start–goal pairs. (a1)–(a2) show 50 robots in a four-room workspace; (b1)–(b2) show 100 robots in an empty workspace. In each pair, the first panel is the WPBS+BFS solution and the second is the trajectory-optimized result. Planning/smoothing took 1.64/1.62 minutes for 50 robots and 1.60/3.45 minutes for 100 robots, respectively. Hollow circles and squares mark starts and goals.

set $H_N(j) = \{i \mid i \prec_N^+ j\}$. Thus, incomparable robots retain independent plans until their trajectories conflict.

At each node, PBS tests the endpoint-augmented trajectories pairwise. A collision between robots i and j generates two children, adding either $i \prec j$ or $j \prec i$. For each child, PBS processes the newly lower-priority robot and any descendants whose higher-priority sets may change in topological order. Unaffected trajectories are reused, and a robot is replanned only when its stored trajectory conflicts with a higher-priority reservation. Infeasible children are discarded, and depth-first search of the priority tree returns the first collision-free node. PBS therefore resolves observed conflicts without enumerating every total robot order.

C. Windowed Coordination

Full-horizon PP or PBS can repeatedly reserve long trajectory suffixes and solve increasingly fragmented ST-GCSs. WPP and WPBS restrict both ECD reservations and conflict detection to a planning window $[t, t+\Delta t_{\text{plan}}]$. After a successful solve, only an execution prefix of duration $\Delta t_{\text{exec}} \leq \Delta t_{\text{plan}}$ is committed; the coordinator then advances the window and replans from the new terminal states. Choosing a shorter execution horizon preserves lookahead while allowing earlier replanning.

A short fixed window can hide interactions that must be resolved before the next prefix is committed. The dynamic variant therefore enlarges the planning window when PP fails or when PBS repeatedly produces the same priority pattern without meaningful progress, and resets it after a successful step. Each enlargement retries the same coordination step with twice the planning span and commits no new trajectory prefix, so additional lookahead is introduced only where the nominal horizon is insufficient. This keeps easy local subproblems small while exposing more future interaction in congested windows. Finite-window coordination is incomplete in general because it defers conflicts outside the current horizon, whereas increasing the window to the remaining horizon recovers the corresponding full-horizon method.

Table I
SINGLE-ROBOT ST-PLANNING RESULTS (60 INSTANCES, 60-S LIMIT). SUCC. IS SOLVED FRACTION; TIME AND COST ARE SUCCESSFUL-RUN MEANS (S).

Method	Succ.	Time	Cost
BFS	**1.00**	**0.1**	11.1
MICP	0.52	10.5	12.5
MICP(g)	0.98	1.6	10.7
ST-RRT*	0.95	0.6	10.0
Zeta*-SIPP (r)	0.77	1.8	**9.9**
Zeta*-SIPP ($2r$)	0.68	0.4	10.7

Table II
SELECTED MRMP RESULTS (60 INSTANCES PER n, 180-S LIMIT). SUCC. IS SOLVED FRACTION; TIME, MAKESPAN (MS), AND SUM-OF-COSTS (SOC) ARE SUCCESSFUL-RUN MEANS (S).

	$n=8$				$n=12$				$n=20$			
Method	Succ.	Time	MS	SoC	Succ.	Time	MS	SoC	Succ.	Time	MS	SoC
WPBS+BFS	**1.00**	6.9	12.5	100	**0.98**	16.9	14.5	174	**0.85**	46.5	**15.5**	310
K-CBS	0.98	**4.2**	25.6	140	0.85	21.8	28.1	221	0.33	83.5	31.0	395
CB-GCS	0.08	48.4	**6.1**	**32.6**	0.02	171	8.5	48.5	0.00	–	–	–
PP+ST-RRT*	0.80	4.9	72.1	123	0.53	**12.5**	91.1	216	0.22	**11.7**	769	909
PBS+Zeta*-SIPP	0.23	94.9	8.1	46.5	0.05	75.6	**5.9**	**42.7**	0.00	–	–	–
WPP+BFS	0.82	14.8	12.9	103	0.48	49.9	13.6	163	0.12	128	17.0	341
PBS+BFS	0.98	27.2	11.6	63.0	0.62	73.6	17.3	101	0.03	129	19.3	**197**

IV. EVALUATION AND DEMONSTRATIONS

Both tables use 20 instances each from random convex-set graphs, recursively divided mazes, and IRIS-generated covers. In Table I, BFS solves all 60 single-robot cases with a 0.1 s mean runtime. Its guided convex subproblems avoid MICP's global integer search; MICP(g) reduces that burden through relaxation but still rounds and optimizes candidate paths. ST-RRT* may miss narrow space-time corridors within the budget; Zeta*-SIPP trades passage coverage on coarse grids against larger searches on fine grids.

The MRMP queries begin at $t = 0$ with intersecting independent trajectories. WPBS+BFS has the highest success rates, reaching 85% at $n = 20$ (Table II). Successful-run costs in both tables reflect different subsets and are not directly comparable.

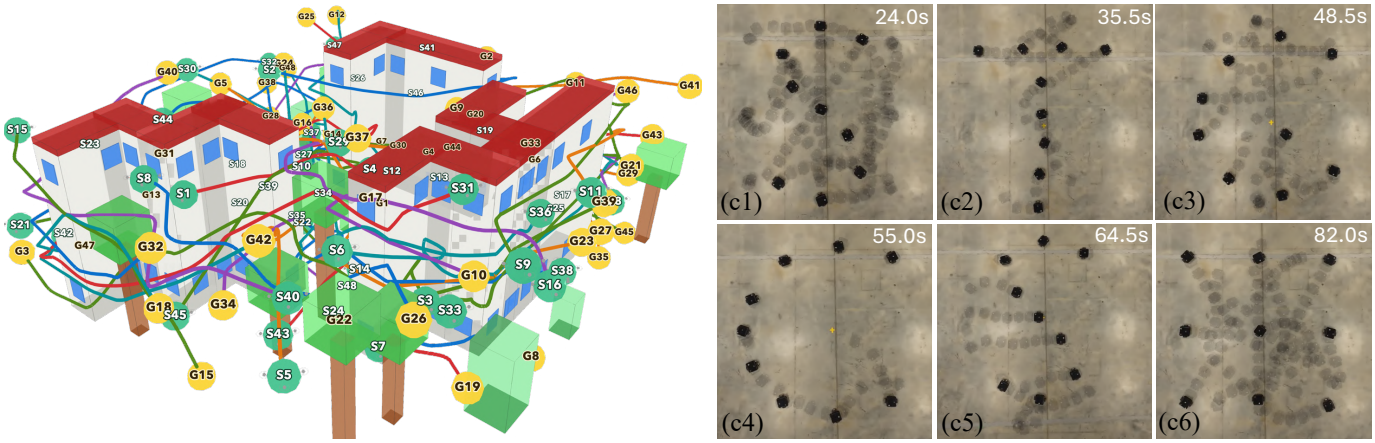


Figure 4. Three-dimensional and physical demonstrations. Left: trajectory-optimized solution for 48 UAVs in a cluttered village (planning/smoothing: 3.75/2.72 minutes). Right: snapshots of nine robots spelling S, T, G, C, and S; translucent ghosts show preceding motion traces.

These results suggest complementary coordination bottlenecks. Fixed priorities can block later robots in WPP+BFS and PP+ST-RRT*, whereas full-horizon PBS+BFS accumulates reservations and graph fragmentation. K-CBS and CB-GCS incur repeated low-level solves as conflicts accumulate; CB-GCS also expands a time-discretized graph, while PBS+Zeta*-SIPP inherits the grid-resolution tradeoff. WPBS+BFS combines priority revision with short reservation windows, limiting each replanning problem. Complete results and ablations appear in our full paper [14].

For the demonstrations only, we optionally smooth the collision-free piecewise-linear ST-GCS solutions with a fixed-time quadratic program. The QP fixes a uniform time grid augmented with the original knot times. The QP then optimizes spatial samples to reduce velocity variation and deviation from the input while preserving endpoints and component-wise velocity limits. Both endpoints of each optimized segment remain in its original convex ST-GCS set, so the entire segment stays within that collision-free region. Linearized halfspaces enforce inter-robot separation at sampled times; consequently, pairwise separation is certified only at those samples. This postprocessing is not part of BFS or PBS and is excluded from the benchmark runtimes in Table II.

Figure 3 demonstrates WPBS+BFS on 50 robots navigating narrow four-room bottlenecks and 100 robots whose initial trajectories form a single 97-robot conflict component. Both cases are planned in under two minutes before optional smoothing.

The left side of Fig. 4 extends the demonstration to 48 UAVs making long-distance trips through a cluttered 3-D village represented by 119 convex sets.

The physical deployment shown on the right of Fig. 4 coordinates nine differential-drive robots through five rearrangements spelling S, T, G, C, and S. The complete task is planned offline in 16.28 s with a 20 cm inter-robot clearance, providing a 6 cm margin over the 14 cm combined physical diameter. During execution, the smoothed trajectories are discretized at 0.05 s intervals and evaluated against a shared clock; a 60-Hz tracker combines segment-wise feedforward commands with body-frame pose-error feedback. The robots complete all rearrangements without inter-robot collisions.

References

[1] 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.

[2] T. Marcucci, J. Umenberger, P. Parrilo, and R. Tedrake, "Shortest paths in graphs of convex sets," *SIAM Journal on Optimization*, vol. 34, no. 1, pp. 507–532, 2024.

[3] J. Tang, Z. Mao, L. Yang, and H. Ma, "Space-time graphs of convex sets for multi-robot motion planning," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 8683–8690.

[4] M. D. Osburn, C. K. Peterson, and J. L. Salmon, "Systematic constraint formulation and collision-free trajectory planning using space-time graphs of convex sets," *arXiv preprint arXiv:2508.10203*, 2025.

[5] M. Phillips and M. Likhachev, "Sipp: Safe interval path planning for dynamic environments," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 5628–5635.

[6] Y. Zou and C. Borst, "Zeta*-sipp: Improved time-optimal any-angle safe-interval path planning." in *IJCAI*, 2024, pp. 6823–6830.

[7] F. Grothe, V. N. Hartmann, A. Orthey, and M. Toussaint, "St-rrt*: Asymptotically-optimal bidirectional motion planning through space-time," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3314–3320.

[8] S. Y. C. Chia, R. H. Jiang, B. P. Graesdal, L. P. Kaelbling, and R. Tedrake, "GCS*: Forward heuristic search on implicit graphs of convex sets," *arXiv preprint arXiv:2407.08848*, 2024.

[9] R. Natarajan, C. Liu, H. Choset, and M. Likhachev, "Implicit graph search for planning on graphs of convex sets," *Robotics: Science and Systems (RSS)*, 2024.

[10] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," *Algorithmica*, vol. 2, pp. 477–521, 1987.

[11] H. Ma, D. Harabor, P. J. Stuckey, J. Li, and S. Koenig, "Searching with consistent prioritization for multi-agent path finding," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 7643–7650.

[12] J. Kottinger, S. Almagor, and M. Lahijanian, "Conflict-based search for multi-robot motion planning with kinodynamic constraints," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 494–13 499.

[13] S. Zhao, A. G. Philip, S. Rathinam, H. Choset, and Z. Ren, "Cb-gcs: Conflict-based search on the graph of convex sets for multi-agent motion planning," in *2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)*. IEEE, 2025, pp. 2208–2214.

[14] J. Tang, Z. Mao, L. Yang, and H. Ma, "Search-based spatiotemporal and multi-robot motion planning on graphs of space-time convex sets," *arXiv preprint arXiv:2607.00444*, 2026.