Unified Branch-and-Bound Search for the Steiner Traveling Salesman Problem on Graphs of Convex Sets

Jingtao Tang and Hang Ma

***Abstract*—Planning through multiple task regions requires jointly choosing visitation order, discrete route, and continuous trajectory. This extended abstract presents the Steiner Traveling Salesman Problem (Steiner-TSP) on Graphs of Convex Sets (GCS), which seeks a minimum-cost closed trajectory through designated target sets while allowing optional transit vertices and revisits. Revisits can induce infinitely many candidate walks on a finite GCS; we search them by unified branch-and-bound over rooted walk prefixes. Additive lower-bound-graph costs bound committed prefixes, while a cut-separated connected-flow relaxation bounds the remaining target visits and return to the root. Their sum supports pruning, traversal-independent global certificates, and deferred trajectory optimization for complete target-covering closed walks. Under uniformly positive vertex costs, best-first terminates after finitely many expansions on every feasible instance; depth-first does so once a finite incumbent is available. Both traversals find feasible solutions on all 180 benchmark instances within 30 s and have mean certified gaps below 30%; a simulated mobile-manipulator task demonstrates joint sensing-mode, order, and trajectory selection with action precedences expressed in linear temporal logic over finite traces.**

I. Steiner-TSP on Graphs of Convex Sets

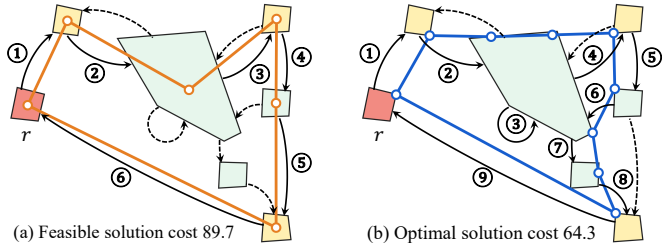
Robotic inspection, equipment servicing, and data collection require visiting task regions and returning to a depot. A mobile manipulator may service each region from different base poses and arm configurations, coupling task order with motion. Roadmap-based generalized TSP selects sampled service configurations and routes [1]; Graphs of Convex Sets (GCS) retain continuous configuration choices within convex regions [2]. For a fixed route, trajectory optimization is convex; jointly choosing the route and trajectory is NP-hard [3]. The TSP on GCS requires visiting every vertex [2], [4], although many regions serve only as transit corridors.

We study the *Steiner-TSP on GCS*, a required-subset generalization of the TSP on GCS. Let $G = (V, E, \mathcal{X})$ be a finite directed GCS. Each vertex v has a compact convex set $\mathcal{X}_v$ and a positive convex cost uniformly lower-bounded by $\eta > 0$; each edge $e = (u, v)$ has a convex feasible transition set $\mathcal{X}_e \subseteq \mathcal{X}_u \times \mathcal{X}_v$ and a nonnegative convex cost. Given a target subset $V_t \subseteq V$ and a root $r \in V_t$, the problem jointly optimizes a finite walk $\pi = (v_1, \dots, v_k)$ and a conditioned trajectory $\tau = (\mathbf{x}_1, \dots, \mathbf{x}_k)$ to minimize $c(\tau)$, which sums vertex and edge costs along the walk. A valid walk follows edges in E; conditioning requires $(\mathbf{x}_i, \mathbf{x}_{i+1}) \in \mathcal{X}_{(v_i, v_{i+1})}$ for $i = 1, \dots, k-1$. Only the targets are required; non-target vertices remain available for transit, and vertices and edges may repeat, matching discrete Steiner-TSP semantics [5]. Feasibility further requires

$$v_1 = v_k = r, \qquad V_t \subseteq \{v_1, \dots, v_{k-1}\}, \qquad \mathbf{x}_1 = \mathbf{x}_k. \quad (1)$$

This extended abstract is a short version of our full paper [6].

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



(a) Feasible solution cost 89.7 (b) Optimal solution cost 64.3

Figure 1. Steiner-TSP on GCS. A closed trajectory starts at the root r (red), visits every target set (yellow), and may use optional sets (green). The feasible no-revisit solution in (a) costs 89.7, whereas the globally optimal solution in (b) costs 64.3 and revisits vertices. Solid arrows denote selected GCS edges; circled numbers indicate traversal order.

Revisits may be necessary for connectivity or reduce trajectory cost (Fig. 1) [4], [7]; unbounded repetitions induce infinitely many finite candidate walks. Existing formulations address this coupling differently. Mixed-integer convex programming (MICP) optimizes route and trajectory jointly [2]; GHOST separates target-tour and GCS-walk search [4]. Both allow revisits across inter-target segments but restrict them within each segment, potentially excluding feasible or cheaper walks. Target-subset augmentation instead encodes visited targets in exponentially many graph layers [8]. Our search explores the complete rooted-walk family, using lower bounds to defer trajectory optimization until a target-covering closed walk is found and to retain global certificates.

II. Unified Branch-and-Bound Search

A. Rooted-walk Prefix Search

We unroll all finite walks beginning at r into a finitely branching prefix tree. A search node n identifies its full prefix $n.\pi = (r, v_2, \dots, v_k)$; keeping the full prefix is necessary since two walks ending at the same vertex can induce different continuous boundary conditions and continuation costs [9], [10]. A node is a discrete solution node when its prefix has visited every target and returned to r. The search solves its possibly infeasible walk-conditioned convex restriction before accepting an incumbent.

The search maintains one *Frontier* of live search nodes whose walk prefixes await exploration. Best-first (BF) selects a node minimizing $f(n) = \hat{g}(n) + h(n)$, whereas depth-first (DF) uses a stack and orders siblings by the same bound.

Whenever a feasible solution node improves the incumbent cost $\bar{c}$, every live prefix whose lower bound is at least $\bar{c}$ is pruned. Convex restrictions are solved only for complete target-covering returns to the root.

An optional nearest-neighbor closed walk supplies the initial incumbent; if its convex restriction is infeasible, the search begins without one. Each iteration prunes by the bound, evaluates solution nodes, and expands every unpruned node—including solution nodes, because a later revisiting return may cost less. The search procedure terminates when the Frontier is exhausted or $\bar{c} \leq \epsilon \min_{n \in \text{Frontier}} f(n)$ for the requested factor $\epsilon \geq 1$. This traversal-independent condition certifies an ϵ-optimal incumbent.

B. Committed-prefix and Cost-to-go Bounds

We reuse a triplet-based lower-bound graph (LBG) that replaces each GCS edge with an LBG vertex and each length-two GCS walk (u, v, w) with a directed transition [4], [10]. The transition cost $\ell_{(u,v,w)}$ is the optimum of a local convex restriction over the three sets. It charges the cost committed at the middle vertex while enforcing local feasibility with both neighboring edges. Adjacent triplets relax their shared continuous variables, so summing their costs lower-bounds any trajectory conditioned on the corresponding GCS walk.

For a prefix $n.\pi$, the additive cost of its fixed LBG walk defines the committed-prefix bound $\hat{g}(n)$. Appending one GCS vertex adds one precomputed triplet cost, making $\hat{g}$ inexpensive to update. To bound the uncommitted part, we propose a connected-flow relaxation on the LBG. It sends cost-bearing flow from the current prefix edge to an artificial closing edge, while directed-cut constraints force that flow to reach an incoming edge of every unvisited target before returning through the root. We solve the exponential cut formulation by row generation: minimum-cut computations identify violated target-connectivity constraints and a persistent linear program is reoptimized. The resulting cost-to-go value $h(n)$ is admissible, so $f(n) \leq c(\tau)$ for every feasible solution (π, τ) extending $n.\pi$.

III. Evaluation and Demonstration

A. Numerical Results

We evaluate six solver variants on 180 minimum-time instances across the *rand*, *maze*, and *iiwa* domains in Fig. 2 (60 per domain across five controlled sizes and 12 seeds). The domains respectively use 2-D polygonal GCSs with every vertex targeted, 3-D voxel-maze decompositions with 10–50 targets, and a 7-DoF joint-space GCS with 3–15 target regions. All instances use degree-5 Bézier segments with C^2 continuity, componentwise velocity bounds, minimum duration $\eta = 0.1\,\text{s}$, and budget $T = 30\,\text{s}$; LBG precomputation and MICP model construction and partial MIP-start generation are excluded. Table I summarizes instance complexity; each triplet is minimum/mean/maximum, and LBG is mean precomputation time in seconds.

The reference cost is the lowest terminal incumbent across solvers. Table II reports Cost Regret relative to this reference,

Table I
Steiner-TSP on GCS instance complexity.

Domain	$\lvert V\rvert$	$\lvert E\rvert$	Degree	$\lvert V_t\rvert$	d	LBG
rand	8/23.6/41	20/60.6/104	1/2.51/7	8/23.6/41	2	0.07s
maze	63/63/63	124/125.2/130	1/1.99/4	10/30/50	3	0.14s
iiwa	18/24/30	104/116.0/128	1/4.93/13	3/9/15	7	4.57s

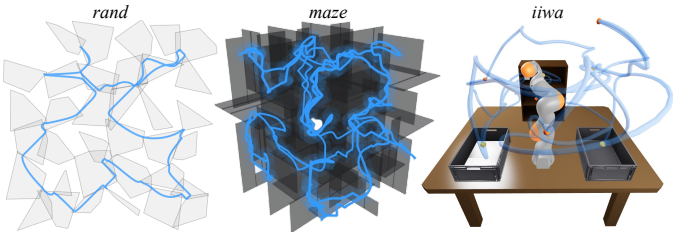


Figure 2. Representative solutions in 2-D *rand*, 3-D *maze*, and 7-DoF *iiwa*; blue curves are optimized trajectories and gray geometry is the GCS decomposition.

Gap to the solver-reported terminal lower bound, and the time-normalized primal integral (NPI), which averages normalized incumbent excess over $[0, T]$, treating time without an incumbent as one. Feasible is the number of runs with finite incumbents; Closed Nodes and Evaluated Walks count processed Frontier nodes and complete target-covering closed walks sent to a convex restriction, respectively.

Ours (BF) and Ours (DF) find incumbents on all 180 instances, with mean certified gaps of 28.1% and 29.7% and mean NPIs of 0.022 and 0.030. GHOST [4] and MICP [2] find incumbents on 99 and 98 instances, respectively; their restricted-family gaps are not global certificates, so cross-method comparison uses feasibility. The ablations expose complementary bottlenecks. Alt-G replaces $\hat{g}$ with a convex restriction at every generated prefix, closing only 43–736 nodes and evaluating no complete walk beyond the greedy initialization. Alt-H replaces h with a loose metric-closure spanning-tree heuristic, closing 314k–871k nodes while evaluating only 1–56 complete walks. Together, the additive prefix and residual-connectivity bounds focus the Frontier on complete walks without solving a trajectory program at every prefix. BF has better anytime behavior in *rand* and *iiwa*, while DF gives the best gap and NPI in the nearly tree-structured *maze*.

B. Mobile Manipulator Inspection

Figure 3 applies Ours (BF) in simulation to a 10-DoF mobile manipulator that leaves a charging dock, acquires eight images, and returns. It has a 7-DoF arm, wrist-mounted camera, and 3-DoF holonomic base. Its 92-vertex, 208-edge GCS couples collision-free navigation regions with arm-configuration regions at feasible sensing poses. Each task is represented by a group of alternative service vertices, of which at least one must be visited; the eight groups yield 96 sensing-mode assignments, and the search jointly selects modes, order, walk, and trajectory. Ours (BF) closes 642 nodes in $1.72\,\text{s}$ of planning and reduces the greedy trajectory duration from 91.4

Table II
ABLATION AND PERFORMANCE COMPARISON RESULTS.

	Solver	Cost Regret↓	Gap↓	Feasible↑	NPI↓	Closed Nodes	Eval. Walks
rand	**GHOST [4]**	2.1%	50.4%	43/60	0.386	–	294
	MICP [2]	24.5%	100.0%	38/60	0.603	–	–
	Ours (BF)	**0.0%**	**18.3%**	**60/60**	**0.011**	8.6k	666
	Ours (DF)	1.8%	21.7%	**60/60**	0.023	1.2k	429
	Alt-G	28.0%	33.0%	**60/60**	0.205	736	1
	Alt-H	26.8%	54.7%	**60/60**	0.197	749k	56
maze	**GHOST [4]**	**0.0%**	3.4%	12/60	0.802	–	23
	MICP [2]	2.2%	100.0%	**60/60**	0.163	–	–
	Ours (BF)	0.7%	3.6%	**60/60**	0.014	53.5k	168
	Ours (DF)	**0.0%**	**3.1%**	**60/60**	**0.005**	1.3k	173
	Alt-G	2.5%	5.2%	**60/60**	0.027	453	1
	Alt-H	2.5%	52.7%	**60/60**	0.027	871k	1
iiwa	**GHOST [4]**	3.9%	64.6%	44/60	0.383	–	37
	MICP [2]	–	–	0/60	1.000	–	–
	Ours (BF)	**0.2%**	**62.4%**	**60/60**	**0.042**	830	54
	Ours (DF)	4.0%	64.4%	**60/60**	0.062	98	30
	Alt-G	10.6%	64.0%	**60/60**	0.111	43	1
	Alt-H	6.2%	65.1%	**60/60**	0.083	314k	37

to 78.4 s, with a 5.64% certified gap; a PRM generalized-TSP baseline returns a 133.1 s trajectory after 31.6 s of planning without an optimality guarantee [1], [11]. We compile the LTL$_f$ precedence $\{\mathrm{A},\mathrm{B}\} \prec \{\mathrm{E},\mathrm{F}\} \prec \mathrm{D} \prec \{\mathrm{G},\mathrm{H}\} \prec \mathrm{C}$ into a deterministic finite automaton tracked by each search node, where $\prec$ denotes precedence [12], [13]. The constrained search returns an 82.3 s trajectory in about 1.3 s after 17 closed nodes, with a 10.12% certified gap.

C. Toward Multi-Robot Collaboration

Future work will extend the framework to small robot teams by combining space-time coordination [14] with joint planning for handovers [15] and cooperative carrying [16]. We will study task allocation and service synchronization alongside convex restrictions for shared-object and contact modes, and develop admissible bounds for team objectives to support valid certificates.

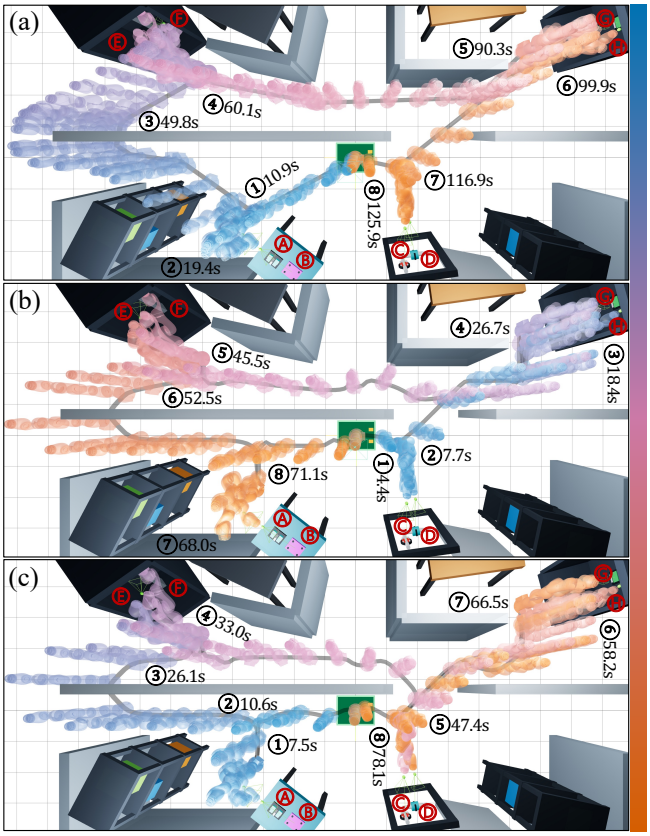


Figure 3. Inspection trajectories from (a) PRM and Ours (BF) (b) without and (c) with LTL$_f$ precedences; labels show task order, timestamps, and arm-pose progression.

REFERENCES

[1] M. Saha, T. Roughgarden, J.-C. Latombe, and G. Sánchez-Ante, "Planning tours of robotic arms among partitioned goals," *The International Journal of Robotics Research*, vol. 25, no. 3, pp. 207–223, 2006.

[2] T. Marcucci, "Graphs of convex sets with applications to optimal control and motion planning," Ph.D. dissertation, Massachusetts Institute of Technology, 2024.

[3] 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.

[4] J. Tang and H. Ma, "GHOST: Solving the traveling salesman problem on graphs of convex sets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 43, 2026, pp. 36 421–36 428.

[5] J. Rodríguez-Pereira, E. Fernández, G. Laporte, E. Benavent, and A. Martínez-Sykora, "The Steiner traveling salesman problem and its extensions," *European Journal of Operational Research*, vol. 278, no. 2, pp. 615–628, 2019.

[6] J. Tang and H. Ma, "Unified branch-and-bound search for the Steiner traveling salesman problem on graphs of convex sets," *arXiv preprint arXiv:2608.21319*, 2026.

[7] S. Morozov, T. Marcucci, B. P. Graesdal, A. Amice, P. A. Parrilo, and R. Tedrake, "Mixed discrete and continuous planning using shortest walks in graphs of convex sets," *arXiv preprint arXiv:2507.10878*, 2025.

[8] G. Luna and T. Summers, "Augmented graphs of convex sets and the traveling salesman problem," *arXiv preprint arXiv:2604.06406*, 2026.

[9] 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.

[10] 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.

[11] 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, 1996.

[12] G. De Giacomo and M. Y. Vardi, "Linear temporal logic and linear dynamic logic on finite traces," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 2013, pp. 854–860.

[13] V. Kurtz and H. Lin, "Temporal logic motion planning with convex optimization via graphs of convex sets," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3791–3804, 2023.

[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.

[15] Z. Wei, X. Luo, and C. Liu, "Hierarchical temporal logic task and motion planning for multi-robot systems," in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025.

[16] C. K. Verginis, Y. Kantaros, and D. V. Dimarogonas, "Planning and control of multi-robot-object systems under temporal logic tasks and uncertain dynamics," *Robotics and Autonomous Systems*, vol. 174, p. 104646, 2024.