

A Surface-Based Formulation of the Traveling Salesman Problem

Yılmaz Arslanoğlu

Independent Researcher, Hamburg, Germany

Abstract

We present an exact formulation of the symmetric Traveling Salesman Problem (TSP) that replaces the classical edge-selection view with a surface-building approach. Instead of selecting edges to form a cycle, the model selects a set of connected triangles where the boundary of the resulting surface forms the tour. This method yields a mixed-integer linear programming (MILP) formulation where a tree constraint enforces global connectivity, while local connectivity at each vertex is guaranteed via Euler characteristic constraints, replacing the need for subtour elimination. The formulation is exact when applied to the complete set of all triangles, despite being computationally intractable for all but the smallest instances. In practice, it provides a compact and effective heuristic when restricted to a sparse candidate set such as Delaunay triangulation.

Keywords: traveling salesman problem, mixed-integer linear programming, tree connectivity constraints, computational topology, Euler characteristic

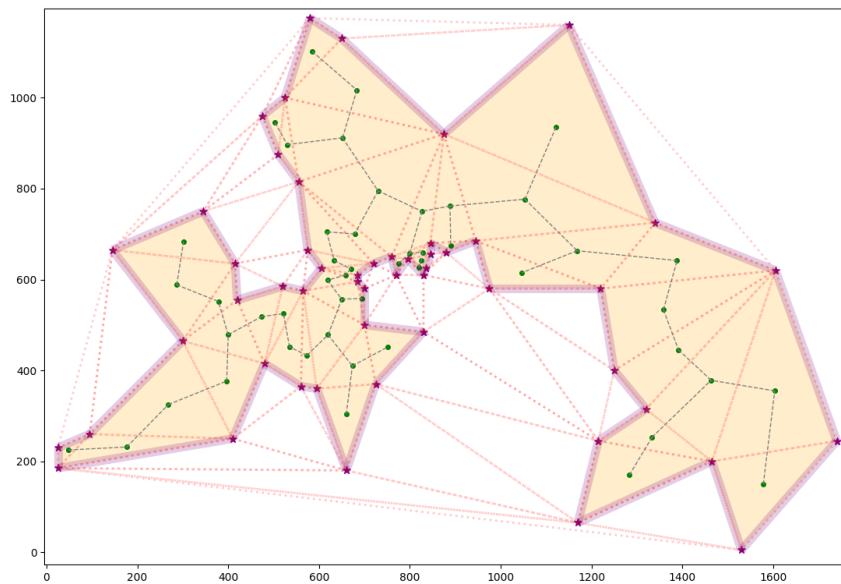


Figure 1: Visual intuition on **berlin52** ($N = 52$): select a connected set of triangles (surface); its boundary is the optimal tour.

1. Introduction

TSP is conventionally modeled on a graph $G = (V, E)$: choose a minimum-weight Hamiltonian cycle by selecting edges $E' \subseteq E$ (Fig. 1). Compact edge-based models enforce local degree conditions but must also encode global connectivity, often via strong but exponential subtour elimination constraints [1], or via weaker compact alternatives such as MTZ [2] or flow-based constraints [3]. We propose a shift in perspective from *graph-theoretic* cycle selection to a *topological* construction. Instead of building the tour as a 1-dimensional sequence of edges, we construct a 2-dimensional surface whose boundary is the Hamiltonian cycle. Specifically, given a candidate triangle set $\mathcal{T} \subseteq \binom{V}{3}$, we select a connected subset of triangles to minimize the total length of its induced boundary edges. When \mathcal{T} is the universal set $\mathcal{K}_{All} := \binom{V}{3}$ (Fig. 2a), the formulation is exact but computationally feasible only for very small N . Restricting \mathcal{T} to a sparse triangulation (e.g., Delaunay - Fig. 2c) yields a highly efficient candidate-set heuristic, though it may render the model infeasible if \mathcal{T} contains no Hamiltonian tour [4].

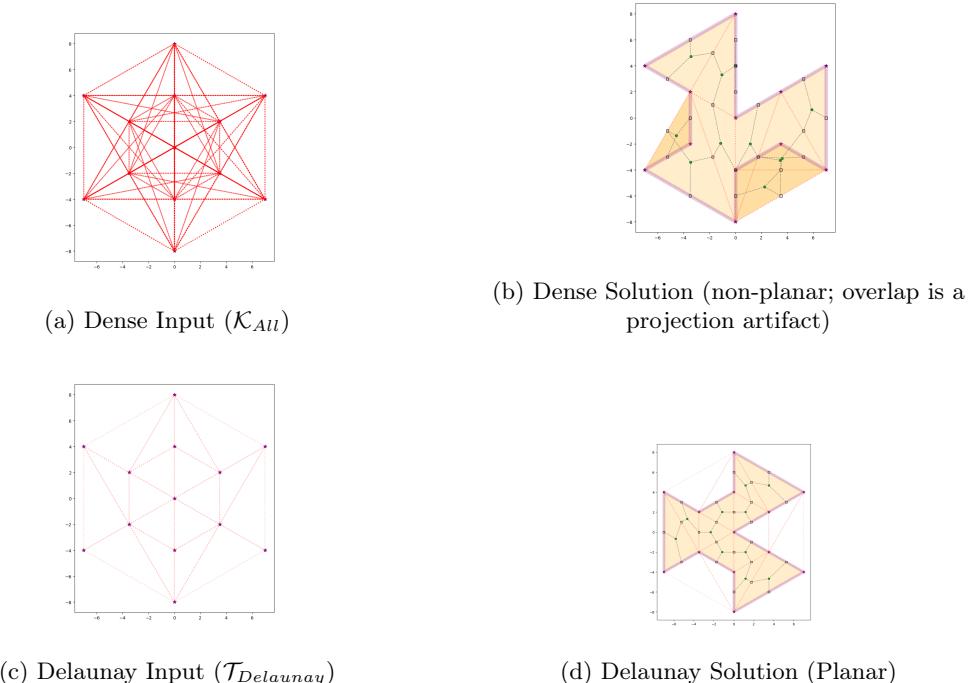


Figure 2: Nested hexagons ($N = 13$): with all triangles the optimum is an abstract non-planar surface; Delaunay input forces a planar one.

1.1. Scope

Our goal is to provide a compact, exact alternative formulation that is structurally different from cycle-based models. We do not aim to compete with state-of-the-art branch-and-cut solvers for pure TSP, but rather to demonstrate that enforcing surface admissibility can yield strong relaxations and useful candidate-set models.

1.2. Related Work

Classical exact methods rely on branch-and-cut with subtour elimination [1, 5]. Tree-structured connectivity constraints are well-studied in flow-based routing formulations [6] and in Steiner vari-

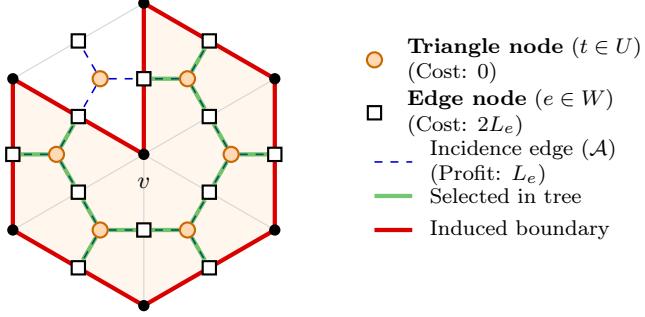


Figure 3: Bipartite incidence graph: triangles x_t (circles) and edges y_e (squares); dashed lines are adjacencies. A feasible solution selects all but one candidate triangle (filled orange), inducing the highlighted boundary (red) around the center; translucent green edges indicate incidence edges chosen in the resulting tree.

ants [7]. Geometric approximation schemes for Euclidean TSP exploit recursive planar subdivisions [8, 9]. Our formulation aligns with boundary-minimization viewpoints from computational topology [10] and geometry [11, 12], planar duality principles [13], and topological optimization of cycles within fixed surfaces [14], but differs in that we optimize the *surface itself* via a simplicial incidence structure. This yields an incidence-graph formulation whose objective has a boundary-cancellation structure related in spirit to prize-collecting terms, except that the “prizes” arise from simplicial incidences, and is related in spirit to geometric covering variants such as TSP with Neighborhoods [15].

1.3. Contribution

Our contribution is threefold:

- (i) We present an equivalence between min-weight Hamiltonian cycles in the *complete* complex \mathcal{K}_{All} (Fig. 2a) and max-weight dual trees in the bipartite incidence graph (Fig. 3);
- (ii) We give a MILP that replaces subtour elimination by a tree constraint plus an Euler characteristic filter (Sec. 3);
- (iii) We demonstrate that the formulation is fully *input-decoupled*: it treats the candidate triangle set \mathcal{T} as a black-box constraint on the search space, working unchanged on the universal set $\mathcal{K}_{All} := \binom{V}{3}$ as well as on sparse geometric triangulations (e.g., Delaunay or Greedy) [4, 16, 17].

Note: For geometric intuition, readers may first consider the planar case described in Appendix D (Fig. D.7) before diving into the general case.

2. Topological Dual via the Incidence Graph

This section introduces the combinatorial structure underlying the surface-based formulation.

2.1. Incidence Graph

Given a TSP instance on $G = (V, E)$, we construct the bipartite *triangle–edge incidence graph* $B = (U \cup W, \mathcal{A})$ (Fig. 3), defined as follows:

- **Triangle nodes (U):** a candidate triangle set $U := \mathcal{T} \subseteq \binom{V}{3}$.
- **Edge nodes (W):** the set of primal edges $W := E$.
- **Incidences (\mathcal{A}):** an arc $(t, e) \in \mathcal{A}$ exists if triangle $t \in U$ contains edge $e \in W$.

A feasible solution of our formulation corresponds to selecting a connected subset of triangles together with the induced active edge set, represented as an active subgraph $B' \subseteq B$.

2.2. Net Weight and Boundary Cancellation

To model boundary minimization, we associate a cost with each primal edge node $e \in W$ and a profit with each incidence $(t, e) \in \mathcal{A}$. For an active subgraph B' , define the net weight

$$W(B') := \sum_{(t, e) \in \mathcal{A}(B')} p(t, e) - \sum_{e \in V(B') \cap W} c(e). \quad (1)$$

With the choice $c(e) = 2L_e$ and $p(t, e) = L_e$, internal edges (shared by two selected triangles) cancel out, while boundary edges contribute exactly their length. As a result, optimizing $W(B')$ is equivalent to minimizing the total length of the induced boundary tour. [23].

3. Mathematical Formulation

We formulate the problem as finding a subgraph in the bipartite incidence graph $B = (U \cup W, \mathcal{A})$ that corresponds to a valid surface. Let $x_t, y_e, z_{te} \in \{0, 1\}$ be decision variables representing the selection of triangles, edges, and their incidences, respectively. The objective is to minimize the length of the boundary $\partial\mathcal{K}$ of the resulting surface complex \mathcal{K} :

$$\text{Min } Z = \sum_{e \in W} 2L_e y_e - \sum_{(t, e) \in \mathcal{A}} L_e z_{te} \quad (2)$$

Eq. (2) is exactly $Z = -W(B')$ for the induced active subgraph B' (Eq. 1), so minimizing Z corresponds to maximizing the net weight. The objective exhibits boundary cancellation and equals the exact length of the induced boundary tour. [23].

We enforce topological validity via four structural constraints (see Appendix C for the explicit MILP constraints):

1. **Global Tree Connectivity:** B' must form a single connected tree (cf. gray dashed lines connecting green nodes in Fig. 1). This ensures \mathcal{K} is simply connected (topologically a disk).
2. **Cardinality:** To guarantee \mathcal{K} is a triangulation of a topological disk with N vertices, we enforce $\sum x_t = N - 2$ (triangles) and $\sum y_e = 2N - 3$ (edges).
3. **Manifold Regularity (Node Degrees):** Every primal edge must be incident to at most two triangles (see Fig. 4A). In the dual, this restricts the degree of edge nodes y_e within the incidence graph.
4. **Local Connectivity (Euler Filter):** To prevent vertex singularities (e.g., “bowties,” Fig. 4B), the neighborhood of each vertex v must be connected (cf. the center vertex in Fig. 3, where a boundary path arises only by omitting an incident triangle). We enforce this using the local Euler characteristic of the incidence subgraph $H_v \subseteq B'$ [10]:

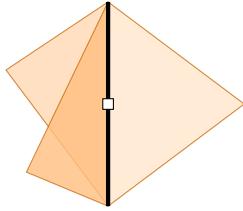
$$\chi(H_v) = |\mathcal{V}(H_v)| - |\mathcal{E}(H_v)| = 1 \quad \forall v \in V. \quad (3)$$

Here H_v denotes the active subgraph of B' induced by the triangles and edges incident to v (see Appendix C for the explicit definition). Under the global tree constraint, enforcing $\chi(H_v) = 1$ leads to a simple path, ruling out disconnected vertex links and local “bowtie” singularities (Fig. 4B) [23].

4. Theoretical Analysis

All formal correctness proofs (boundary cancellation, disk admissibility, Hamiltonicity, and the resulting exactness of the MILP on the full complex) are given in the companion note [23]. Here we keep only the high-level takeaway: the objective measures the induced boundary length, and the tree+manifold+Euler constraints ensure the selected 2-complex is a disk whose boundary is a single Hamiltonian cycle.

A. Non-manifold edge



B. "Bowtie" singularity

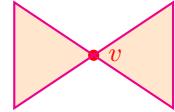


Figure 4: Forbidden anomalies: (A) Edge incident to > 2 triangles; (B) A "bowtie" vertex violating the Euler Filter.

5. Computational Validation and Results

We validated the correctness of the formulation on simple topological cases before measuring performance.

5.1. Visual Verification (Topology)

To provide a visual verification, we apply the model to an instance consisting of two nested hexagons with a central node ($N = 13$). Fig. 2 compares the solver's behavior under two different candidate triangle sets: the universal set \mathcal{K}_{All} and the Delaunay triangulation $\mathcal{T}_{Delaunay}$. In both cases, the formulation successfully extracts the optimal surface within the chosen input complex.

5.2. Validation on Non-Metric Spaces

To test the topological generality of our formulation, we generated random symmetric non-metric instances of small size (including $N = 10$) where edge weights do not satisfy the triangle inequality. The input was \mathcal{K}_{All} . The solver successfully extracted valid dual trees and primal tours; Fig. 5 shows a representative $N = 10$ example. We verified optimality against a standard TSP formulation (MTZ).

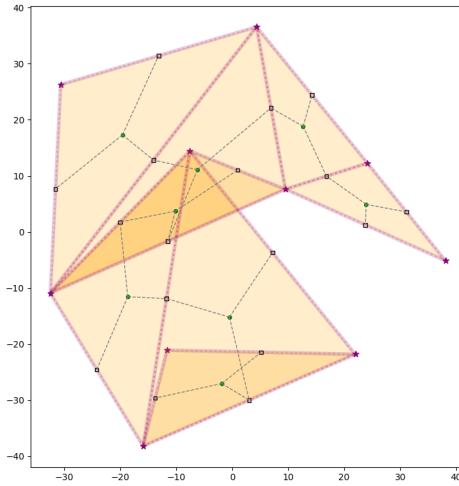


Figure 5: Random non-metric instance ($N = 10$, visualized by MDS). The formulation relies only on topological connectivity, allowing it to solve instances where the triangle inequality is violated.

5.3. Comparison

We compare our surface formulation against the standard Lifted-MTZ baseline on identical sparse inputs (see Appendix A for detailed setup and full results). The surface-based model typically reduces branch-and-bound (B&B) effort and can solve larger instances more reliably within the class of compact, static MILPs considered. Among surface-based variants, enforcing the global tree via single-commodity flow yields the tightest relaxation and dominates the MTZ-style tree enforcement as N increases. On TSPLIB Euclidean instances where both baselines are available (`berlin52`, `st70`, `ch130`) under Delaunay sparsification, the Surface-Based-Flow model solves at the root (1 B&B node), while the Lifted-MTZ baseline requires 2,936–16,058 nodes (Appendix A).

5.4. When does the formulation work well?

The formulation is exact when the candidate triangle set \mathcal{T} is the complete complex $\binom{V}{3}$. In practice, performance heavily relies on restricting \mathcal{T} to a sparse candidate complex.

Empirically, the approach works best when \mathcal{T} is a planar triangulation that contains at least one Hamiltonian tour. Delaunay triangulations often provide a strong default in Euclidean instances, while other geometric candidates (e.g., Greedy triangulations) can improve solution quality depending on the problem type.

5.5. Degenerate Inputs

We further tested “hard-to-solve” instances [18] and degenerate inputs [19] (Appendix B). Figure B.6 visualizes representative solutions for these instances. On `Tnm199`, our formulation effectively exploits the richer Greedy triangulation to achieve a 1.85% gap (vs. 6.17% for Delaunay), surpassing the 5.69% gap of the Christofides heuristic [20]. This confirms that the topological approach can identify superior tours when provided with higher-quality geometric candidates. On `p100` parallel-line degeneracies, triangulation-based methods degrade badly (14%–119% gap vs. 2.52%), exposing a limitation of sparse candidate sets on near-degenerate geometry.

6. Conclusion

We have presented a surface-based formulation for TSP that shifts the decision space from edges to triangles. This changes the optimization object from a primal cycle to a dual surface: instead of directly enforcing Hamiltonicity on G , we enforce disk-admissibility of a 2-complex whose boundary is the tour. Our experiments indicate that this topological shift can yield improved B&B behavior relative to compact, static edge-based baselines on sparse geometric inputs, particularly when enforcing the global tree constraint via flow. While exact convergence becomes computationally challenging for larger instances in our current setup, the results suggest that the surface-based model provides a useful foundation for integrating additional geometric or topological constraints that are cumbersome in primal edge formulations, without changing the core model.

Acknowledgements

We thank Prof. Stefan Hougardy for his feedback and for providing the test instances. We also thank Anh Quyen Vuong for his suggestions regarding the formulation structure. Finally, we extend our special thanks to the developers of HiGHS for their commitment to providing a powerful and open-source optimization technology.

A. Detailed Computational Results (Standard Instances)

A.1. Detailed Setup and Performance Analysis

To evaluate both the topological properties and the solver-specific performance, we benchmarked three formulations:

1. **Conventional:** Standard Lifted-MTZ formulation [21] restricted to edges in the Delaunay triangulation.
2. **Surface-Based (MTZ):** Topological formulation using MTZ-based tree constraints [2].
3. **Surface-Based (Flow):** Topological formulation using Single-Commodity Flow tree constraints [6].

Model Selection: For experiments using sparse planar inputs (Delaunay and Greedy), we utilized the **Planar Simplification** model described in Appendix D. This reduces the problem size by eliminating explicit edge node variables (y_e), as the incidence graph collapses into the standard dual graph. We utilized **HiGHS v1.12** [22] on an **Apple M4 Max**. Unless noted, runtime results are arithmetic means of 10 runs with distinct random seeds. Problem sizes (Vars/Cons) are reported after solver presolve. Table A.1 summarizes the performance across all instances. The results reveal a clear transition in which the **Surface-Based-Flow** formulation dominates **Surface-Based-MTZ** as N grows. For small instances ($N \leq 130$), both surface-based models typically solve at the root. For larger instances, the single-commodity flow constraint yields a substantially tighter relaxation, reflected in dramatically smaller Branch-and-Bound trees (e.g., 142 nodes vs. 23,205 for **pcb442**). This supports the interpretation that our formulation benefits from flow-based connectivity enforcement more strongly than from MTZ-style ordering constraints, even though both encode the same topological admissibility conditions. We compare against compact static models, excluding dynamic cut-generation methods (e.g., Concorde [5]) which remain state-of-the-art for pure TSP.

Table A.1: **Combined Computational Results.** Post-presolve MILP size and performance; values are 10-run means (unless noted), with a timeout limit of 1-hour.

Instance	Model	MILP Size		HiGHS	
		Vars	Cons	Time (s)	[Solved]
Berlin52 ($N = 52$)	Lifted-MTZ	341	523	0.08s	[10/10]
	Surface-Based-MTZ	553	1,107	0.02s	[10/10]
	Surface-Based-Flow	684	1,054	0.03s	[10/10]
St70 ($N = 70$)	Lifted-MTZ	463	723	14.23s	[10/10]
	Surface-Based-MTZ	759	1,521	4.35s	[10/10]
	Surface-Based-Flow	926	1,440	0.20s	[10/10]
Ch130 ($N = 130$)	Lifted-MTZ	879	1,362	108.6s	[10/10]
	Surface-Based-MTZ	1,458	2,922	0.29s	[10/10]
	Surface-Based-Flow	1,793	2,775	1.62s	[10/10]
Tnm199 ($N = 199$)	Surface-Based-Flow (Del)	2,764	4,283	0.51s	[10/10]
	Surface-Based-Flow (Grd)	2,090	3,202	4.76s	[10/10]
A280 ($N = 280$)	Lifted-MTZ	1,907	2,992	DNF	[0/5]
	Surface-Based-MTZ	3,188	6,379	29.0s	[10/10]
	Surface-Based-Flow	3,923	6,056	5.68s	[10/10]
Pcb442 ($N = 442$)	Surface-Based-MTZ	5,195	10,393	1656s*	[7/10]
	Surface-Based-Flow	6,412	9,879	81.1s	[10/10]
d657 ($N = 657$)	Surface-Based-Flow	9,634	14,837	TLE (0.60%)	>30k

* For **pcb442** Surface-Based-MTZ, 3/10 runs exceeded the 1-hour time limit; time and nodes are averaged over the 7 solved runs.

A.2. Impact of Input Sparsity on Lifted-MTZ

To justify the choice of Delaunay-restricted input sets for the baseline comparison, we evaluated the performance of the standard Lifted-MTZ formulation on the complete graph K_N (Table A.2). The results show a massive divergence in performance. On **st70**, HiGHS runtime increases by roughly 13.5x (14.2s vs. 191.7s) when moving from the Delaunay-restricted graph to the complete graph K_N . The topological formulation remains extremely fast on the sparse complex (0.2s), indicating that the topological structure and the restricted geometry both contribute materially to scalability (a further ~ 71 x speedup).

Table A.2: **Lifted-MTZ on Complete Graph vs. Delaunay.** On **st70** ($N = 70$), sparsity gives ~ 13.5 x and the surface-based approach gives a further ~ 71 x.

Instance	Edge Set	HiGHS (s)
St70	Complete (K_N)	191.7s
	Delaunay	14.2s
	Topological	0.2s

B. Extended Experimental Results (Hard Instances)

To further evaluate the impact of the input complex on solution quality, we tested the formulation on particularly difficult instances.

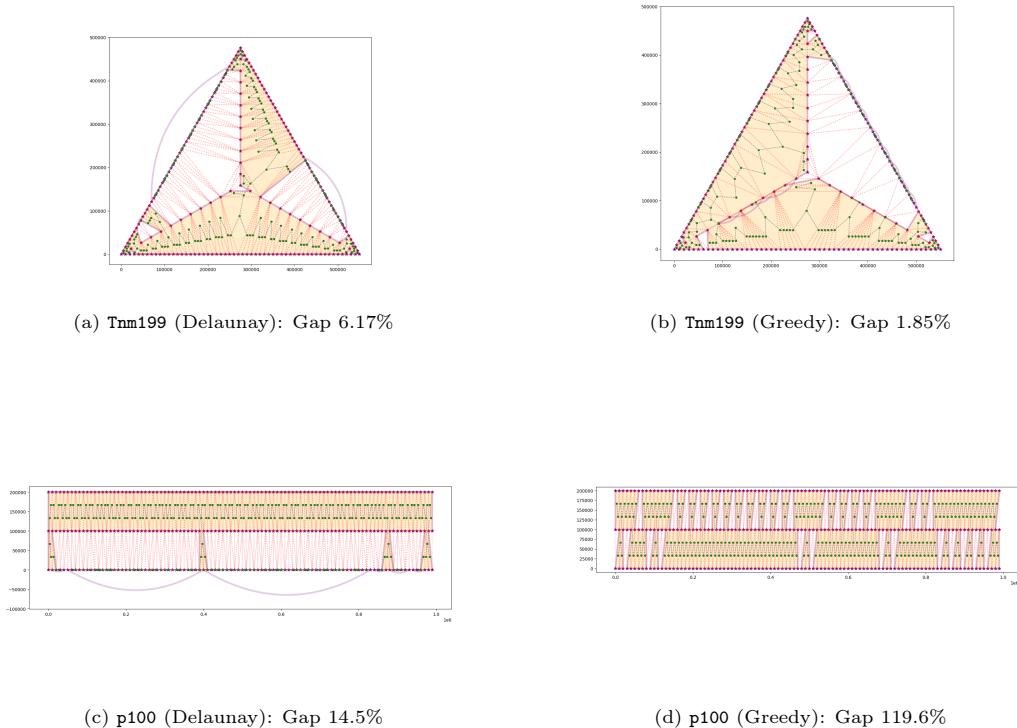


Figure B.6: Solution quality on Hard Instances. Top Row: On **Tnm199**, Greedy (b) finds a superior surface to Delaunay (a). Bottom Row: On the degenerate **p100**, heuristics fail; Greedy (d) is especially catastrophic compared to Delaunay (c).

B.1. Analysis of Hard Instances: Tnm199

We tested on **Tnm199** ($N = 199$) from the "Hard-to-Solve" dataset [18]. This instance is constructed to be practically intractable for standard exact solvers; Concorde requires 411,222 seconds (≈ 4.7 days) to find the global optimum. We compared our topological method using two input triangulation strategies (Delaunay and Greedy) against the standard Christofides algorithm. The results (Table B.3) illustrate the model's dependence on input quality. While the Delaunay-based solution (Fig B.6a) is structurally limited (similar to Christofides), the Surface-Based-Flow model on the Greedy triangulation successfully extracts the superior geometry within the complex (Fig. B.6b), narrowing the gap to **1.85%**. This demonstrates that the topological formulation scales in quality with the richness of the input complex, unlike fixed constructive heuristics.

Table B.3: **Results on Tnm199.** Greedy input enables a tighter gap than Delaunay and Christofides.

Method	Time (s)	Objective	Gap (%)
Global Opt. (Concorde)	>400k	3,139,778	0.00
Christofides	< 1.00	3,318,561	+5.69
Surface-Based-Flow (Delaunay)	0.51	3,333,452	+6.17
Surface-Based-Flow (Greedy)	4.76	3,197,767	+1.85

B.2. Sensitivity Analysis: Parallel Line Degeneracy

To test the limitations of sparsification under degenerate geometry, we evaluated the instance **p100.100000** ($N = 100$), which arranges points on three parallel lines [19]. We compared the Surface-Based-Flow formulation (using Delaunay and Greedy inputs) against the standard Christofides heuristic. The results in Table B.4 reveal a critical vulnerability in geometric sparsifiers. While Christofides remains robust (2.52% average gap), the triangulation-based methods suffer substantially (Fig. B.6c-d).

Table B.4: **Results on p100.100000.** Parallel-line degeneracy breaks triangulation-based candidate sets.

Method	Objective	Gap (%)
Global Opt. (Exact)	4,160,200	0.00
Christofides	4,265,038	+2.52
Surface-Based-Flow (Delaunay)	4,762,994	+14.49
Surface-Based-Flow (Greedy)	9,136,966	+119.63

C. Explicit MILP Formulation

Here we provide the complete Mixed-Integer Linear Program corresponding to the high-level formulation in Section 3.

C.1. Logical Linking and Closure

Active simplices must be connected via the triangle–edge incidence relation \mathcal{A} .

$$z_{te} \leq x_t, \quad z_{te} \leq y_e \quad \forall (t, e) \in \mathcal{A}. \quad (\text{C.1})$$

Define the induced degrees

$$\deg_U(t) := \sum_{e:(t,e) \in \mathcal{A}} z_{te} \quad \forall t \in U, \quad \deg_W(e) := \sum_{t:(t,e) \in \mathcal{A}} z_{te} \quad \forall e \in W.$$

Degree/Closure Constraint: If a triangle is active, all three of its edges must be active:

$$\deg_U(t) = 3x_t \quad \forall t \in U. \quad (\text{C.2})$$

C.2. Topological Regularity

These global counts follow from the Euler characteristic of the intended disk-like complex [10].

$$\sum_{t \in U} x_t = N - 2 \quad (\text{Triangle Cardinality}) \quad (\text{C.3})$$

$$\sum_{e \in W} y_e = 2N - 3 \quad (\text{Edge Cardinality}) \quad (\text{C.4})$$

$$y_e \leq \deg_W(e) \leq 2y_e \quad \forall e \in W \quad (\text{Degree/Manifold}). \quad (\text{C.5})$$

C.3. Local Euler Filter and Connectivity

For each city $v \in V$, let

$$U(v) := \{t \in U \mid v \in t\}, \quad W(v) := \{e \in W \mid v \in e\},$$

and define the restricted incidence set

$$\text{Conn}(v) := \{(t, e) \in \mathcal{A} \mid t \in U(v), e \in W(v)\}.$$

Define the local counts

$$T(v) := \sum_{t \in U(v)} x_t,$$

$$E(v) := \sum_{e \in W(v)} y_e,$$

$$I(v) := \sum_{(t, e) \in \text{Conn}(v)} z_{te},$$

and the local Euler characteristic

$$\chi(v) := T(v) + E(v) - I(v).$$

We enforce [10]

$$\chi(v) = 1 \quad \forall v \in V. \quad (\text{C.6})$$

C.4. Global Tree Constraint (Directed Single-Commodity Flow)

We designate a root node r dynamically. To break symmetries and improve performance, we restrict the candidate root set $R_{\text{cand}} \subset U$ to the set of triangles incident to the city with the minimum degree in the mesh. We introduce binary variables $r_t \in \{0, 1\}$ for $t \in R_{\text{cand}}$ and enforce:

$$\sum_{t \in R_{\text{cand}}} r_t = 1, \quad r_t \leq x_t \quad \forall t \in R_{\text{cand}}. \quad (\text{C.7})$$

For all other nodes $i \notin R_{\text{cand}}$ (including all edge nodes W), we define $r_i := 0$. To enforce linearity and tighter relaxations, we model a *Rooted Directed Tree* on the bipartite graph. Let

$$A_{\text{dir}} = \{(t, e), (e, t) \mid (t, e) \in \mathcal{A}\}$$

be the set of directed arcs. The total number of active nodes is a known constant

$$K = (N - 2) + (2N - 3) = 3N - 5.$$

Active-node indicator. For each node $i \in U \cup W$, define

$$\text{active}_i := \begin{cases} x_t & \text{if } i = t \in U, \\ y_e & \text{if } i = e \in W. \end{cases}$$

1. Directed Arc Selection: For every incidence $(t, e) \in \mathcal{A}$, we introduce two directed binary variables $a_{te}, a_{et} \in \{0, 1\}$ and enforce:

$$a_{te} + a_{et} = z_{te} \quad \forall (t, e) \in \mathcal{A}. \quad (\text{C.8})$$

2. Flow Capacity and Conservation: Let $f_{uv} \geq 0$ be the continuous flow on arc $(u, v) \in A_{\text{dir}}$. Let M be a sufficiently large constant (e.g., $M = K$).

$$f_{uv} \leq M \cdot a_{uv} \quad \forall (u, v) \in A_{\text{dir}} \quad (\text{Capacity}) \quad (\text{C.9})$$

$$\sum_{u:(u,i) \in A_{\text{dir}}} f_{ui} - \sum_{w:(i,w) \in A_{\text{dir}}} f_{iw} = \text{active}_i - K \cdot r_i \quad \forall i \in U \cup W \quad (\text{Conservation}) \quad (\text{C.10})$$

3. Rooted Tree Structure:

Every active non-root node has exactly one incoming arc:

$$\sum_{u:(u,i) \in A_{\text{dir}}} a_{ui} = \text{active}_i - r_i \quad \forall i \in U \cup W. \quad (\text{C.11})$$

D. Special Case: The Planar Simplification

In the general case (Section 3), the Bipartite Graph is necessary because a primal edge e (edge node) may be incident to many triangles. However, if the candidate set \mathcal{T} is a planar triangulation (e.g., Delaunay), every primal edge is incident to at most 2 triangles. In this specific case, the explicit edge nodes y_e can be removed. The Bipartite Graph B collapses into a simple dual graph, where edges connect neighboring triangles (Fig. D.7). Consequently, the Euler filter simplifies from the bipartite form ($\chi(v) = 1$) to the standard dual-graph form ($|V_{\text{dual}}| - |E_{\text{dual}}| = 1$). This reduces the variable count by $O(|E|)$ and simplifies the constraints. The manifold degree constraint becomes implicit.

Boundary Cancellation. In this planar case, the objective simplifies by eliminating explicit edge-node variables, and can be derived from Eq. (2).

$$\text{Min } Z = \sum_{t \in \mathcal{T}_{\text{selected}}} P(t) - 2 \sum_{e \in \mathcal{E}_{\text{dual}}} L(e) \quad (\text{D.1})$$

where $P(t)$ is the perimeter of triangle t and $L(e)$ is the length of the primal edge shared by the two triangles connected by dual edge e .

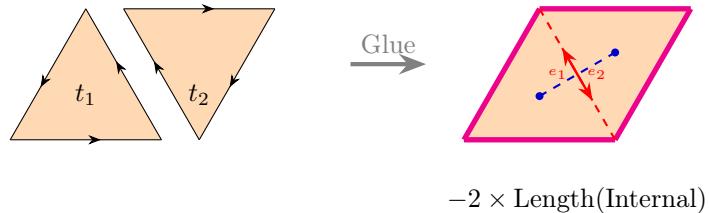


Figure D.7: Combinatorial boundary identity: internal shared edges cancel by opposite orientations, leaving only the boundary.

References

- [1] G. B. Dantzig, R. E. Fulkerson, S. M. Johnson, Solution of a Large-Scale Traveling-Salesman Problem, *Operations Research* 2(4), 393–410 (1954).
- [2] C. E. Miller, A. W. Tucker, R. A. Zemlin, Integer Programming Formulation of Traveling Salesman Problems, *Journal of the ACM* 7(4), 326–329 (1960).
- [3] K. R. Fox, B. Gavish, S. C. Graves, An n -Constraint Formulation of the (Time-Dependent) Traveling Salesman Problem, *Operations Research* 28(4), 1018–1021 (1980).
- [4] M. B. Dillencourt, Toughness and Delaunay triangulations, *Discrete & Computational Geometry* 5(6) (1990), 575–601.
- [5] D. L. Applegate, R. E. Bixby, V. Chvátal, W. J. Cook, *The Traveling Salesman Problem: A Computational Study*, Princeton University Press, 2006.
- [6] B. Gavish, S. C. Graves, The Traveling Salesman Problem and Related Problems, Technical Report GR-078-78, Operations Research Center, Massachusetts Institute of Technology, 1978.
- [7] A. N. Letchford, S. D. Nasiri, D. O. Theis, Compact Formulations of the Steiner Traveling Salesman Problem and Related Problems, *European Journal of Operational Research* 228(1), 83–92 (2013).
- [8] S. Arora, Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and Other Geometric Problems, *Journal of the ACM* 45(5), 753–782 (1998).
- [9] J. S. B. Mitchell, Guillotine Subdivisions Approximate Polygonal Subdivisions, *SIAM Journal on Computing* 28(3), 839–867 (1999).
- [10] H. Edelsbrunner, J. Harer, *Computational Topology: An Introduction*, American Mathematical Society, 2010.
- [11] S. P. Fekete, A. Haas, M. Hemmer, J. Trögel, Computing Nonsimple Polygons of Minimum Perimeter, in: *Proceedings of the 15th International Symposium on Experimental Algorithms (SEA 2016)*, Springer, pp. 134–149, 2016.
- [12] S. P. Fekete, A. Haas, P. Keldenich, M. Perk, A. Schmidt, Computing Area-Optimal Simple Polygonizations, *ACM Journal of Experimental Algorithms* 27, 2.6:1–2.6:23 (2022).
- [13] H. Whitney, A Theorem on Graphs, *Annals of Mathematics* 32(2), 378–390 (1931).
- [14] E. W. Chambers, J. Erickson, A. Nayyeri, Minimum Cuts and Shortest Homologous Cycles, in: *Proceedings of the 25th Annual Symposium on Computational Geometry (SoCG 2009)*, ACM, pp. 377–385, 2009.
- [15] A. Dumitrescu, J. S. B. Mitchell, Approximation Algorithms for TSP with Neighborhoods in the Plane, *Journal of Algorithms* 48(1), 135–159 (2003).
- [16] G. Das, D. Joseph, Which triangulations approximate the complete graph?, in: *Proceedings of the International Symposium on Optimal Algorithms*, Springer, 1989, pp. 168–192.
- [17] A. N. Letchford and N. A. Pearson, Good triangulations yield good tours, *Computers & Operations Research* 35 (2008), 638–647.

- [18] S. Hougardy, X. Zhong, Hard to Solve Instances of the Euclidean Traveling Salesman Problem, Technical Report, Research Institute for Discrete Mathematics, University of Bonn, 2018.
- [19] S. Hougardy, On the Integrality Ratio of the Subtour LP for Euclidean TSP, *Operations Research Letters* 42(8), 495–499 (2014).
- [20] N. Christofides, Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem, Technical Report, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.
- [21] M. Desrochers, G. Laporte, Improvements and Extensions to the Miller-Tucker-Zemlin Subtour Elimination Constraints, *Operations Research Letters* 10(1), 27–36 (1991).
- [22] Q. Huangfu, J. A. J. Hall, Parallelizing the Dual Revised Simplex Method, *Mathematical Programming Computation* 10(1), 119–142 (2018).
- [23] Y. Arslanoğlu, “A Structural Equivalence of Symmetric TSP to a Constrained Group Steiner Tree Problem,” *arXiv preprint arXiv:2602.05773*, 2026.
<https://arxiv.org/abs/2602.05773>.