13th International Conference on Applied Informatics
Eszterházy Károly Catholic University
Eger, Hungary, February 19–21, 2026

Learning Transferable Geometric Edge Priors for Euclidean TSP Using Graph Neural Networks

Ali Jlidi[a*], **László Kovács**[a*]

[a]Institute of Information Technology, University of Miskolc, Hungary
ali.jlidi@uni-miskolc.hu,laszlo.kovacs@uni-miskolc.hu

Abstract

The Euclidean Traveling Salesman Problem (TSP) remains a cornerstone of combinatorial optimization [1]. While neural approaches, particularly Graph Neural Networks (GNNs), have shown promise [5], many struggle with generalization across problem scales [6]. This paper introduces **TGEP-TSP**, a novel learning framework that decouples the learning of reusable structural knowledge from the construction of feasible solutions. Instead of predicting complete tours end-to-end, a single GNN is trained to learn a *transferable geometric edge prior*—scoring edges based on their likelihood of appearing in high-quality tours. This prior is learned from heuristic solutions on sparse k-nearest neighbor graphs across a mix of instance sizes (20–100 cities). A soft degree regularization term guides the model towards TSP-valid structures. At inference, a constrained greedy decoding algorithm converts the probabilistic edge scores into valid Hamiltonian cycles, which can be optionally refined with lightweight 2-opt local search.

Problem 1 (Euclidean Traveling Salesman Problem)**.** *Given n cities $V = \{v_1, \dots, v_n\}$ with coordinates $p_i = (x_i, y_i) \in [0,1]^2$, find a Hamiltonian cycle $\pi : [n] \to [n]$ minimizing:*

$$L(\pi) = \sum_{i=1}^{n} \|p_{\pi(i)} - p_{\pi(i \bmod n+1)}\|_2.$$

The combinatorial complexity grows as $(n-1)!/2$, making exact solutions infeasible for large n.

*This research was supported by the University of Miskolc, Hungary.

Definition 2 (Transferable Geometric Edge Prior)**.** A scoring function $f : \mathcal{E} \to [0, 1]$ that estimates the likelihood of an edge appearing in a high-quality TSP tour. This function is learned to be invariant to instance size and specific node configuration, capturing fundamental geometric principles of Euclidean TSPs [4].

Experiments on synthetic Euclidean instances demonstrate that TGEP-TSP effectively generalizes to unseen sizes, including out-of-distribution problems with 200 cities. After 2-opt refinement, it often matches or slightly surpasses the performance of its teacher heuristic (Nearest Neighbor + 2-opt). Preliminary tests on TSPLIB benchmarks confirm the robustness of the learned prior to real-world geometric distributions.

Theorem 3. *A Graph Neural Network trained on a mixed-size dataset of Euclidean TSP instances can learn a geometric edge prior that generalizes to larger, unseen instance sizes. This is evidenced by the model maintaining a low optimality gap on $n = 200$ node problems despite being trained on $n \leq 100$.*

Methodological Framework

The TGEP-TSP framework operates in two distinct phases:

1. **Global Learning Phase**: A GNN is trained on a diverse set of instances. The learning objective combines a weighted binary cross-entropy loss for edge classification with a soft degree regularization loss $\mathcal{L}_{\text{deg}} = \lambda \sum_{i=1}^{n} (\sum_{j \in \mathcal{N}(i)} s_{ij} - 2)^2$, encouraging the predicted edge subgraph to approximate the degree-2 constraint of a Hamiltonian cycle.

2. **Constrained Decoding Phase**: For a new instance, edge scores from the trained model are used within a greedy, constraint-aware algorithm [7]. This decoder incrementally builds a tour by selecting high-score edges while explicitly enforcing degree and subtour elimination constraints, guaranteeing a valid Hamiltonian cycle as output.

Table 1. Performance on synthetic Euclidean TSP instances. A negative gap indicates improvement over the teacher heuristic [3]. The model generalizes effectively to $n = 200$, an unseen size during training.

Method	**Avg. Gap to Teacher (%)**			
	$n = 20$	$n = 50$	$n = 100$	$n = 200$
Nearest Neighbor + 2-opt (Teacher)	0.0	0.0	0.0	0.0
TGEP-TSP (no refinement)	2.8	3.1	3.7	4.5
TGEP-TSP + 2-opt	**-1.3**	**-1.1**	**-0.6**	**1.8**

Table 1 summarizes key results, showing that TGEP-TSP with refinement improves upon its teacher on in-distribution sizes and remains competitive on larger, out-of-distribution problems.

Remark 4. The success of TGEP-TSP stems from its separation of concerns: the GNN learns a *transferable* prior over edge utility, while combinatorial feasibility is enforced by a dedicated, correct-by-construction decoder. This contrasts with end-to-end neural methods that entangle learning and feasibility, often hindering generalization [8].

Conclusion and Impact

This work establishes learning transferable geometric priors as a viable and robust paradigm for neural combinatorial optimization [2]. By focusing on learning reusable structural knowledge rather than complete solution strategies, TGEP-TSP achieves strong generalization across scales. The framework is orthogonal and complementary to powerful local search heuristics, providing high-quality starting solutions for further refinement. Future work will explore learning from optimal solutions, extending to non-Euclidean spaces, and integrating the decoding constraints more deeply into the learning process.

References

[1] D. L. Applegate, R. E. Bixby, V. Chvatal, W. J. Cook: *The Traveling Salesman Problem: A Computational Study*, Princeton University Press, 2006.

[2] Y. Bengio, A. Lodi, A. Prouvost: *Machine learning for combinatorial optimization: A methodological tour d'horizon*, European Journal of Operational Research 290.2 (2021), pp. 405–421.

[3] K. Helsgaun: *An effective implementation of the Lin–Kernighan traveling salesman heuristic*, European Journal of Operational Research 126.1 (2000), pp. 106–130.

[4] C. K. Joshi, T. Laurent, X. Bresson: *An efficient graph convolutional network technique for the travelling salesman problem*, INFORMS Journal on Computing 34.4 (2022), pp. 1854–1869.

[5] W. Kool, H. van Hoof, M. Welling: *Attention, Learn to Solve Routing Problems!*, International Conference on Learning Representations (2019).

[6] Q. Liu, A. Mandalika, C. R. Shelton: *How Good is Neural Combinatorial Optimization?*, arXiv preprint arXiv:2209.10913 (2022).

[7] M. Prates, P. H. C. Avelar, H. Lemos, L. C. Lamb, M. Y. Vardi: *Learning to solve NP-complete problems: A graph neural network for decision TSP*, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019.

[8] O. Vinyals, M. Fortunato, N. Jaitly: *Pointer Networks*, in: Advances in Neural Information Processing Systems, 2015.