

A Statistically Rigorous, Held–Karp- and LKH-Validated Empirical Evaluation of a Classical 2-opt + Or-opt Local Search Pipeline for the Euclidean Traveling Salesman Problem

Mohamad Fatahi

fatahi.m1990@gmail.com

Jahan Sanat Jonoub Knowledge-Based Company

Hamzeh Barati²

Research Article

Keywords: Traveling Salesman Problem, 2-opt, Or-opt, Held–Karp bound, Lin–Kernighan–Helsgaun, TSPLIB, empirical algorithm evaluation, learning-based routing heuristics

Posted Date: October 27th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11120328/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

Abstract

Local search heuristics such as 2-opt and Or-opt remain foundational to practical Traveling Salesman Problem (TSP) solving, yet rigorous, statistically grounded empirical evaluation against a certified optimality reference is less commonly reported than one might expect. We evaluate a classical nearest-neighbor + 2-opt + Or-opt pipeline using 30 independent trials per problem size ($n \in \{50, 100, 200, 500\}$), Bonferroni-corrected Wilcoxon signed-rank tests, bootstrap confidence intervals, matched-pairs rank-biserial effect sizes, and a self-computed Held–Karp [1] lower bound rather than an asymptotic approximation. The hybrid pipeline improves over 2-opt alone by 2.09–3.48% (all comparisons significant after correction) and remains within 4.9–5.6% of the certified lower bound. We further validate these findings on four real TSPLIB instances (eil51, berlin52, st70, kroA100), cross-checked against LKH [2], which recovers the published optimum exactly in all four cases; the hybrid pipeline's real-instance gaps (0.94–5.78%) are consistent with its synthetic-instance results. We additionally report, for completeness, a negative result for a tested extension that orders Or-opt moves using Held–Karp Lagrangian dual information, which produced no statistically significant benefit. We situate these results relative to a fast-growing, separate body of work on learned and hybrid learned/exact TSP and routing solvers published in the data-mining and knowledge-engineering literature [14]–[18], which pursues algorithmic novelty rather than certified-bound statistical characterization of classical local search. We do not claim algorithmic novelty in the local-search mechanism; our contribution is methodological rigor and cross-validated empirical characterization.

1. Introduction

The Traveling Salesman Problem (TSP), which involves finding the shortest possible closed route that visits a specific set of cities exactly once, has been investigated by many researchers over the years because of its wide range of applications in vehicle routing, printed circuit board drilling, DNA sequencing, and network design of logistics. Due to the poor scalability of exact algorithms, heuristics and local search algorithms have dominated the practical investigation of the TSP problem.

Heuristic strategies for local search, which start from a candidate solution and improve it using simple operations like edge exchanges (2-opt; [4]) and segment repositioning (Or-opt; [5]) as well as their variants [6], have stood the test of time and have become standard tools of current TSP algorithms both on their own and as parts of complex metaheuristics and branch-and-bound solvers. While these approaches enjoy a long and rich history, scientific and statistically rigorous empirical evaluation of these classical local-search schemes in terms of their comparison to the optimality benchmark, rather than just to each other, appears to be underreported in isolation from other research, despite many important empirical results obtained by Johnson & McGeoch in [7]. The latter research direction belongs to another, relatively new field of data mining and knowledge engineering, and its efforts concentrate on developing novel algorithms as learned and/or learned+exact solvers [14]–[18].

Contribution of this paper is methodological and empirical, not algorithmic: we present a statistically sound comparison of a nearest neighbor + 2-opt + Or-opt local search procedure, compared to a proven Held-Karp [1] lower bound, rather than an asymptotic estimate, on both random Euclidean problem instances and publicly available TSPLIB instances [8], and cross-validated against the state-of-the-art heuristic LKH [2]. In addition, we present a negative result for an obvious extension that we have tried, so as to not waste efforts of future research on the same principle. The rest of this paper is structured as follows: Section 2 summarizes literature related to our work, classical as well as learning-based, and places our contribution in the context; Section 3 details our methodology for experiments; Section 4 gives the results obtained on synthetic and real problem instances; and Section 5 concludes.

2. Related Work

Exact and near-exact methods. Dantzig et al. [9] introduced cutting plane techniques for exact solution of TSP; Concorde is the most advanced exact TSP solver [3], solving instances with tens of thousands of cities to provable optimality. Such approaches are very compute-intensive and usually applied only when a high-quality solution is needed; we use Held & Karp's [1] Lagrangian 1-tree relaxation as a fast and computable lower-bound, and LKH as a near-exact benchmark (not an exact solution), as opposed to exact approach. The bound itself is still a subject of research: Recently Righini [20] demonstrated how the selection of the fixed 'special' vertex, necessary for construction of 1-tree, can be optimized so as to maximize the Held & Karp bound with the same worst-case computation complexity as minimum spanning tree problem, improvement which is not employed in our subgradient optimization (Section 3.2).

Construction and local-search heuristics. Nearest-neighbor heuristic construction and the local search with 2-opt edge exchange [4, 10] are the earliest popular TSP heuristics; Or [5] extended the former with segment relocation ("Or-opt") neighborhood structure. Lin & Kernighan [6] further extended both to form a variable-depth search, which continues to be, per Helsgaun [2, 11], LKH implementation thereof, the best-performing heuristic discovered for large Euclidean instances. Christofides [12] devised the first exact $3/2$ -approximation algorithm for metric TSP but at a considerable practical cost of inferior performance on typical random Euclidean instances compared to local search [7]. This approximation guarantee remained unbroken for over four decades before Karlin, Klein & Oveis Gharan [19] presented an (albeit marginal) improvement through randomized approach based on sampling of max-entropy spanning trees — a note that even the approximation-guaranteed side of TSP is still very much alive.

Empirical characterization. The case study by Johnson & McGeoch [7] is still the most complete empirical study of 2-opt, 3-opt, and Or-opt optimality gaps, which typically vary between low single-digit percentages and $\approx 10\%$ depending on problem size, initial solution, and choice of operators – the gap interval to which our findings (0.94–5.78% on real-life data sets, 2–6% on synthetic data sets) belong. Beardwood, Halton & Hammersley [13] have determined the asymptotic constant that describes the average optimum tour length for randomly generated Euclidean TSPs; we employed this constant as an analytical reference before switching to the Held-Karp bound suggested by the reviewer (Sect. 4).

Recent learning-based and data-mining approaches. A separate and rapidly growing line of work, concentrated in the data-mining and knowledge-engineering literature, integrates machine learning with combinatorial search rather than relying purely on classical move operators. Zheng et al. [14] combine reinforcement learning with the Lin–Kernighan–Helsgaun heuristic, using a learned policy to bias candidate-edge and move selection within LKH's search, reporting tour-quality improvements over standard LKH on TSPLIB and random Euclidean instances. Zong et al. [15] propose a hierarchical reinforcement-learning framework (RBG) for large-scale routing problems in logistics systems, decomposing large customer sets into learned regional partitions before applying regional routing. Tong et al. [16] integrate combinatorial optimization with reinforcement learning for large-scale taxi order dispatching, a bipartite-matching relative of vehicle routing, showing that a synergistic RL/CO integration outperforms either component alone at industrial scale. Zhang et al. [17] introduce Neuro-Ins, an attention-based one-shot node-insertion model for dynamic pickup-and-delivery routing that avoids the myopia of classical cheapest-insertion heuristics. Goh et al. [18] present a hierarchical neural constructive solver tailored to realistic (non-uniform) TSP instance distributions, showing that learned, location-biased construction outperforms both uniformly-trained neural solvers and classical construction heuristics on industrial benchmarks.

In terms of what makes the approaches in these works different from ours, two factors should be mentioned. Firstly, their baselines and gains are measured either against the best-known or previous published solutions, while in our case we measure everything against a guaranteed, computed lower bound. Secondly, the contribution in these papers is the new solution approach, i.e., learned policy or a mixture of the learned policy and exact components, whereas in our paper, the contribution is the rigorous empirical and statistical evaluation of the existing local search algorithm in contrast to the certified lower bound. These two lines of research can be considered complimentary as opposed to competitive since the certified Held-Karp gaps and significance-testing protocol that we present in the context of classical 2-opt + Or-opt (Sect. 4) can be used as a rigorously characterized baseline for future learning algorithms, e.g., [14]–[18]. We do not conduct such a head-to-head comparison here, since it would require re-implementing or obtaining trained models for [14]–[18] under matched instance distributions and compute budgets, which is outside the scope of a methodological paper on classical local search; we flag it explicitly as future work in Sect. 5.

Concurrent 2026 developments. In the wake of the present pipeline and its evaluation framework being developed, a number of additional TSP solvers have been introduced, indicating that if anything, the rate of development within the broader literature is only increasing. Shen et al. [21] introduce AGDN, an anisotropic graph diffusion network TSP generator, appearing at KDD 2026 – the same conference as [15] and [18]. Song et al. [23] introduce DualCert, a neural–OR hybrid solver which features a refinement procedure constrained to lie on a Karush-Kuhn-Tucker feasibility manifold and which internally employs a Held-Karp ascent procedure and an LKH-3 tour as its reference for verification – a similar approach to ours (Sect. 3.2, Sect. 4.2), though there applied towards certifying the output of a learned solver instead of the statistical characterization of a fixed classical pipeline as in our work. Soria-Lorente et al. [22] introduce ADPSO-ERLS, a non-learned hybrid discrete particle swarm/2-opt solver that is evaluated, like

us, under a strict matched computational budget (there, the number of solution evaluations) across TSPLIB instances.

An experimental comparison with [21]–[23] is not provided in this paper. These three references have come to light in the last few weeks preceding this revision, and have yet to independently reproduce their own results or be validated against a Held-Karp bound using a Bonferroni correction to significance, as has been done for all other methods considered here; thus, a comparison of the methods against those in [21]–[23], which are similarly based on learning, is impossible at present, much as it has been for the learning-based work reviewed in [14]–[18]. This paper specifically mentions [21]–[23] instead of excluding them from consideration for this reason, and extends the future work comparisons proposed in Sect. 5 to these references as well.

Positioning of this work. We are not proposing any new form of local search or heuristic, the combination of 2-opt and Or-opt that we test is precisely the classical combination studied by Or [5] and Johnson and McGeoch [7]. The point of difference between this prior work, both the classical baselines mentioned above and the learning baselines discussed earlier, lies in methodology: (i) testing against a self-generated, proven lower bound of the Held-Karp kind rather than against weak baselines, an asymptotic estimate, or the best-known solution; (ii) Bonferroni-corrected statistical testing (Wilcoxon signed-rank test, bootstrap confidence interval, and matched pairs rank-biserial effect size) over various instance sizes; (iii) cross-validation against LKH-generated solutions on real TSPLIB problems; and (iv) reporting of a transparent negative result on a tested variant. This is summarized in Table 3.

Table 3

Positioning relative to closely related prior work, including recent learning-based baselines from top-ranked data-mining and knowledge-engineering venues and three concurrent 2026 solvers

Study	Move set	Optimality reference	Statistical testing	Real-instance validation
Croes [4]	2-opt	None reported	No	No
Or [5]	Or-opt	None reported	No	No
Lin & Kernighan [6]	Variable-depth (LK)	Compared to known optima (small n)	No	Yes (≤ 110 cities)
Johnson & McGeoch [7]	2-opt, 3-opt, Or-opt, LK	Held–Karp bound (aggregate)	Limited	Yes (large study)
Helsgaun [2]	LKH (generalized LK)	Known optima / best-known	No	Yes (up to 7397 cities)
Zheng et al. [14]	Learned move-ordering + LKH (hybrid RL/LK)	Best-known / no certified bound	No	Yes (TSPLIB)
Zong et al. [15]; Goh et al. [18]	Learned hierarchical/neural construction (RL)	Best-known / no certified bound	Limited (mean over seeds)	Yes (large-scale / real logistics)
Soria-Lorente et al. [22]	Hybrid discrete PSO + candidate-restricted 2-opt (classical, non-learned)	Best-known / no certified bound	No (budget-matched only)	Yes (5 TSPLIB instances)
Song et al. [23]	Learned + KKT-manifold refinement (hybrid neural/OR)	Held–Karp-ascent bound + LKH-3 reference (uncertified)	No	Yes (synthetic, up to 1000 cities)
Shen et al. [21]	Learned graph diffusion construction	Best-known / no certified bound	No	Yes (standard synthetic benchmarks)
This work	2-opt + Or-opt (classical)	Held–Karp bound (per-instance) + LKH cross-check	Bonferroni-corrected Wilcoxon, bootstrap CI, effect size	Yes (4 TSPLIB instances) + synthetic

3. Methodology

3.1 Algorithms evaluated

The three algorithms we test are: (i) nearest neighbor (NN) construction; (ii) NN followed by 2-opt local search until convergence or until a time budget has been exhausted; and (iii) a pipeline where 2-opt is

performed first, followed by Or-opt (path relinking, path lengths 1–3) with one half of the time budget, and a second 2-opt iteration at the end. All three are classic techniques that have already been described elsewhere (Sect. 2).

3.2 Optimality reference: the Held–Karp bound

We calculate the lower bound of the Held-Karp [1] using the Lagrangian relaxation of the 1-tree formulation through the use of subgradient optimization (50–80 iterations with geometrically cooled step size). This gives us the best possible self-computed, guaranteed lower bound on the actual optimal value for each individual problem, without relying on an asymptotic bound (see [13]), which applies only in the asymptotic regime of large n . We refrain from using the recent method of selecting a special vertex, introduced by Righini [20], that allows one to improve the lower bound obtained from the 1-tree formulation without any additional asymptotic cost; its use would have improved (raised) our bound in Sect. 4.

3.3 Experimental design

Synthetic examples: n points generated independently uniformly at random from a 1000×1000 square, where $n \in \{50, 100, 200, 500\}$, 30 independent runs each, same wall clock time budget for each example (1.0 s if $n \leq 200$, 3.0 s if $n = 500$) irrespective of the algorithm compared. Real examples: four TSPLIB examples with known optimal tour lengths [8] – see Sect. 4.2. All tours obtained were programmatically confirmed to be legitimate Hamiltonian circuits before evaluating them. Statistical tests: paired Wilcoxon signed-rank test with Bonferroni adjustment ($\alpha = 0.05/4 = 0.0125$) on the four synthetic problem sizes, bootstrap (2000 samples) 95% confidence interval on the percent improvement, and matched pairs rank-biserial correlation as the effect size measure.

4. Experimental Results

4.1 Synthetic instances

Table 1

Tour length (mean $\pm$ SD) for 2-opt and the hybrid (2-opt + Or-opt) pipeline, with 95% bootstrap confidence intervals on percentage improvement, Bonferroni-adjusted Wilcoxon signed-rank p-values, matched-pairs rank-biserial effect size r , and the Held–Karp lower-bound optimality gap.

n	2-opt (mean $\pm$ SD)	Hybrid (mean $\pm$ SD)	Improvement (95% CI)	p (Bonferroni α = .0125)	Effect size r	Held– Karp gap
50	5996.3 $\pm$ 303.0	5871.3 $\pm$ 310.5	+ 2.09% [1.57, 2.64]	< 0.001 (sig.)	-1.00	4.93% $\pm$ 2.54
100	8355.9 $\pm$ 313.8	8065.0 $\pm$ 303.9	+ 3.48% [2.64, 4.42]	< 0.001 (sig.)	-1.00	5.14% $\pm$ 2.19
200	11437.3 $\pm$ 259.2	11115.7 $\pm$ 242.4	+ 2.81% [2.42, 3.18]	< 0.001 (sig.)	-1.00	5.61% $\pm$ 1.38
500	17885.5 $\pm$ 316.4	17384.4 $\pm$ 281.8	+ 2.80% [2.56, 3.06]	< 0.001 (sig.)	-1.00	n/a [⊠]

[⊠] Held–Karp bound was not computed per-trial at $n = 500$ due to computational cost within our budget; reported for $n \leq 200$ where tractable.

Hybrid pipeline is an improvement of 2.09–3.48% over the 2-opt for all examined instances and remains statistically significant even after Bonferroni adjustment ($p < 0.0125$ in each case). The effect size $r = -1.00$ in all cases results from the fact that the tour length of hybrid pipeline has been always less than or equal to that of the 2-opt in all 120 paired comparisons, which is not a surprising result in terms of the structure of the hybrid pipeline. Held-Karp gap guarantees that the solutions produced by the hybrid pipeline are within 4.9–5.6% from the proven lower bound for $n \leq 200$, in line with Johnson & McGeoch’s [7].

4.1.1 Algorithmic complexity

The time complexity of one 2-opt step is $O(n^2)$ per iteration; the time complexity of the Or-opt algorithm with segments of size 1 to 3 and full reinsertion search is $O(n^2)$ per iteration. Time complexity of the Held-Karp subgradient method is $O(n^2)$ per iteration. No list of candidates, do-not-look bits, and neighbors’ pruning were used; standard optimizations were excluded because of simplicity.

4.1.2 Tested extension: Held–Karp-guided Or-opt ordering (negative result)

The study involved an examination of whether ordering the segments used in Or-opt in terms of degree violation of the best Held-Karp 1-tree in relation to node degree violation, where the degree violation occurs in the nodes identified by LP as having the highest degree violation, could yield better solutions. Under a generous computational time limit, both the guided and conventional ordering yielded statistically similar results (+ 0.00%, $p = 0.65$; +0.05%, $p = 0.35$) for 100 and 200 respectively. With a

limited computational time limit excluding Held-Karp computation cost from the optimization time limit, there was no clear benefit from the guided ordering (-0.03% to + 0.02%, $p > 0.37$).

4.1.3 Threats to validity

They are synthetic (uniform random Euclidean); the results cannot be generalized for structured or practical routing instances without the validation that is done in Sect. 4.2. Comparison is based on the wall clock time budget instead of an operations count that is hardware independent. No comparison to Lin-Kernighan-type heuristic algorithms is made in the analysis of the synthetic instances (although LKH is used as a benchmark in Sect. 4.2); the pipeline that we analyze here needs to be seen as a characterized classical benchmark and not as a state-of-the-art algorithm. The Held-Karp bound by subgradient optimization does not always converge to the exact solution of the 1-tree LP problem; thus the optimality gaps are upper bounds.

4.2 Validation on real TSPLIB instances

In addition to evaluating the pipeline on synthetic instances, we evaluated the pipeline on four real TSPLIB instances where the optimal tour lengths have been reported [8] : eil51, berlin52 (52 locations in Berlin), st70, and kroA100 ($n = 51-100$). TSPLIB has remained the benchmark for the TSP problem since its development more than three decades ago, and continues to remain the test bed for the real instances in recent learning based studies [14] as well. We have also calculated the Held–Karp lower bound for each of these instances, and used LKH [2] for comparison.

Table 2

Optimality gaps (percentage above the published optimal tour length) for NN, 2-opt, and the hybrid pipeline on four real TSPLIB instances, alongside our Held–Karp lower bound and an LKH reference solution.

Instance	n	Known optimal	Held–Karp LB (ours)	NN gap	2-opt gap	Hybrid gap (ours)	LKH gap	LKH time
eil51	51	426	424	+ 19.95%	+ 2.82%	+ 0.94%	0.000%	0.12s
berlin52	52	7542	7543	+ 19.07%	+ 6.87%	+ 5.33%	0.000%	0.10s
st70	70	675	674	+ 22.96%	+ 7.56%	+ 5.78%	0.000%	0.18s
kroA100	100	21282	20925	+ 30.66%	+ 9.20%	+ 3.01%	0.000%	0.24s

LKH solved the known optimal tour for all four instances with an exact gap of 0%, thereby verifying the known optimal solutions and corroborating our implementation of the TSPLIB coordinate transcription and EUC_2D distance approximation. The gap results obtained by our hybrid pipeline for real world instances (0.94–5.78%) are in line with the results for synthetic instances (see Table 1), suggesting that the results obtained on synthetic instances generalize to the real world instances. LKH wall clock times

(0.10–0.24s, obtained by authors themselves locally outside sandboxed evaluation environment used in Table 1) are not directly comparable to our budgets.

4.3 Data and code availability

Problem instance generation code, algorithms implementations and analysis tools needed for generation of Tables 1–2 and Figs. 1–4 are accessible at <https://github.com/umah2000/tsp-local-search-evaluation>. Problem instances are generated by the code with the help of fixed random seeds.

5. Conclusion

We conducted a statistically sound empirical analysis of the performance of a classical nearest neighbor + 2-opt + Or-opt TSP local search procedure, on both artificial and real TSPLIB instances, held up to the statistical validity standards of Held-Karp lower bounds and the LKH solver. This pipeline delivers consistent and statistically significant 2–3.5% improvements over 2-opt and stays under $\approx 5\%$ away from a provable optimal solution for all considered instances. We observed a negative result with an evaluated extension involving a linear-programming based order of moves. We do not make any claims regarding algorithmic innovation in terms of the local search technique itself. Further work may consider scaling the evaluation to larger instances via the use of candidate-list-based implementations, extending the move set along the lines of Lin-Kernighan heuristic, tightening the provably optimal solution via the optimized 1-tree choice of Righini [20], and, by following the certifiable lower bound and statistical rigour approach taken in this study, performing a direct budget-matched evaluation against the new class of learned and learned/exact hybrid TSP algorithms from Sect. 2 including the earlier literature [14]–[18] and concurrent 2026 [21]–[23] solvers.

Declarations

Conflict of interest:

The authors declare no conflict of interest.

Ethics approval:

Not applicable — this study did not involve human participants, human data, or animals.

Statement

During the preparation of this work, the authors used Claude AI in order to optimize and debug code. After using this service, the authors reviewed and edited the content as needed and takes full responsibility for the content of the published article.

Funding:

The authors received no specific funding for this work.

Author Contribution

M.F: Conceptualization, Methodology, Investigation, Formal analysis, Data Curation, Visualization, Writing – Original Draft.H.B: Conceptualization, Validation, Supervision, Project Administration, Writing – Review & Editing.

Data Availability

Problem instance generation code, algorithms implementations and analysis tools needed for generation of Tables 1-2 and Figs. 1-4 are accessible at <https://github.com/umah2000/tsp-local-search-evaluation>. Problem instances are generated by the code with the help of fixed random seeds.

References

1. Held, M., & Karp, R. M. (1970). The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18(6), 1138–1162. <https://doi.org/10.1287/opre.18.6.1138>
2. Helsgaun, K. (2000). An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1), 106–130. [https://doi.org/10.1016/S0377-2217\(99\)00284-2](https://doi.org/10.1016/S0377-2217(99)00284-2)
3. Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (2006). The Traveling Salesman Problem: A Computational Study. Princeton University Press. ISBN 978-0-691-12993-8.
4. Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations Research*, 6(6), 791–812. <https://doi.org/10.1287/opre.6.6.791>
5. Or, I. (1976). Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking. PhD thesis, Department of Industrial Engineering and Management Sciences, Northwestern University.
6. Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2), 498–516. <https://doi.org/10.1287/opre.21.2.498>
7. Johnson, D. S., & McGeoch, L. A. (1997). The traveling salesman problem: A case study in local optimization. In E. H. L. Aarts, & J. K. Lenstra (Eds.), *Local Search in Combinatorial Optimization* (pp. 215–310). Wiley.
8. Reinelt, G. (1991). TSPLIB – a traveling salesman problem library. *ORSA Journal on Computing*, 3(4), 376–384. <https://doi.org/10.1287/ijoc.3.4.376>

9. Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4), 393–410.
<https://doi.org/10.1287/opre.2.4.393>
10. Flood, M. M. (1956). The traveling-salesman problem. *Operations Research*, 4(1), 61–75.
<https://doi.org/10.1287/opre.4.1.61>
11. Helsgaun, K. (2009). General k-opt submoves for the Lin–Kernighan TSP heuristic. *Mathematical Programming Computation*, 1(2), 119–163. <https://doi.org/10.1007/s12532-009-0004-6>
12. Christofides, N. (1976). Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University.
13. Beardwood, J., Halton, J. H., & Hammersley, J. M. (1959). The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society*, 55(4), 299–327.
<https://doi.org/10.1017/S0305004100034095>
14. Zheng, J., He, K., Zhou, J., Jin, Y., & Li, C. M. (2023). Reinforced Lin–Kernighan–Helsgaun algorithms for the traveling salesman problems. *Knowledge-Based Systems*, 260, 110144.
<https://doi.org/10.1016/j.knosys.2022.110144>
15. Zong, Z., Wang, H., Wang, J., Zheng, M., & Li, Y. (2022). RBG: Hierarchically solving large-scale routing problems in logistic systems via reinforcement learning. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*, pp. 4648–4658.
<https://doi.org/10.1145/3534678.3539037>
16. Tong, Y., Shi, D., Xu, Y., Lv, W., Qin, Z., & Tang, X. (2023). Combinatorial optimization meets reinforcement learning: Effective taxi order dispatching at large-scale. *IEEE Transactions on Knowledge and Data Engineering*, 35(10), 9812–9823. <https://doi.org/10.1109/TKDE.2021.3127077>
17. Zhang, Z., Yang, J., Cao, Z., & Lau, H. C. (2025). Neuro-Ins: A learning-based one-shot node insertion for dynamic routing problems. *IEEE Transactions on Knowledge and Data Engineering*, 37(9), 5495–5507. <https://doi.org/10.1109/TKDE.2025.3580640>
18. Goh, Y. L., Cao, Z., Ma, Y., Dong, Y., Dupty, M. H., & Lee, W. S. (2024). Hierarchical neural constructive solver for real-world TSP scenarios. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*, pp. 884–895.
<https://doi.org/10.1145/3637528.3672053>
19. Karlin, A. R., Klein, N., & Gharan, O. (2021). S. A (slightly) improved approximation algorithm for metric TSP. In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC 2021)*, pp. 32–45. <https://doi.org/10.1145/3406325.3451009>
20. Righini, G. (2021). Efficient optimization of the Held–Karp lower bound. *Open Journal of Mathematical Optimization*, article 9, 1–17. <https://doi.org/10.5802/ojmo.11.2>
21. Shen, B., Huang, Z., Cao, Z., & Dong, Y. (2026). AGDN: Learning to solve traveling salesman problem with anisotropic graph diffusion network. In: *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26)*, Jeju Island, Republic of Korea.
<https://doi.org/10.1145/3770855.3817789>

22. Soria-Lorente, A., Vilaire, J. M., Michel, J., & Alvarado-Barrios, L. (2026). ADPSO-ERLS: A hybrid discrete PSO with enhanced local search for the traveling salesman problem. arXiv:2608.09786 [math.OC] (preprint).

23. Song, Y., Qi, Y., Qi, W., & Shen, Z. J. M. (2026). DualCert: A solver for the traveling salesman problem with constraint-coupled learning. arXiv:2608.09042 [cs.AI] (preprint).

Figures

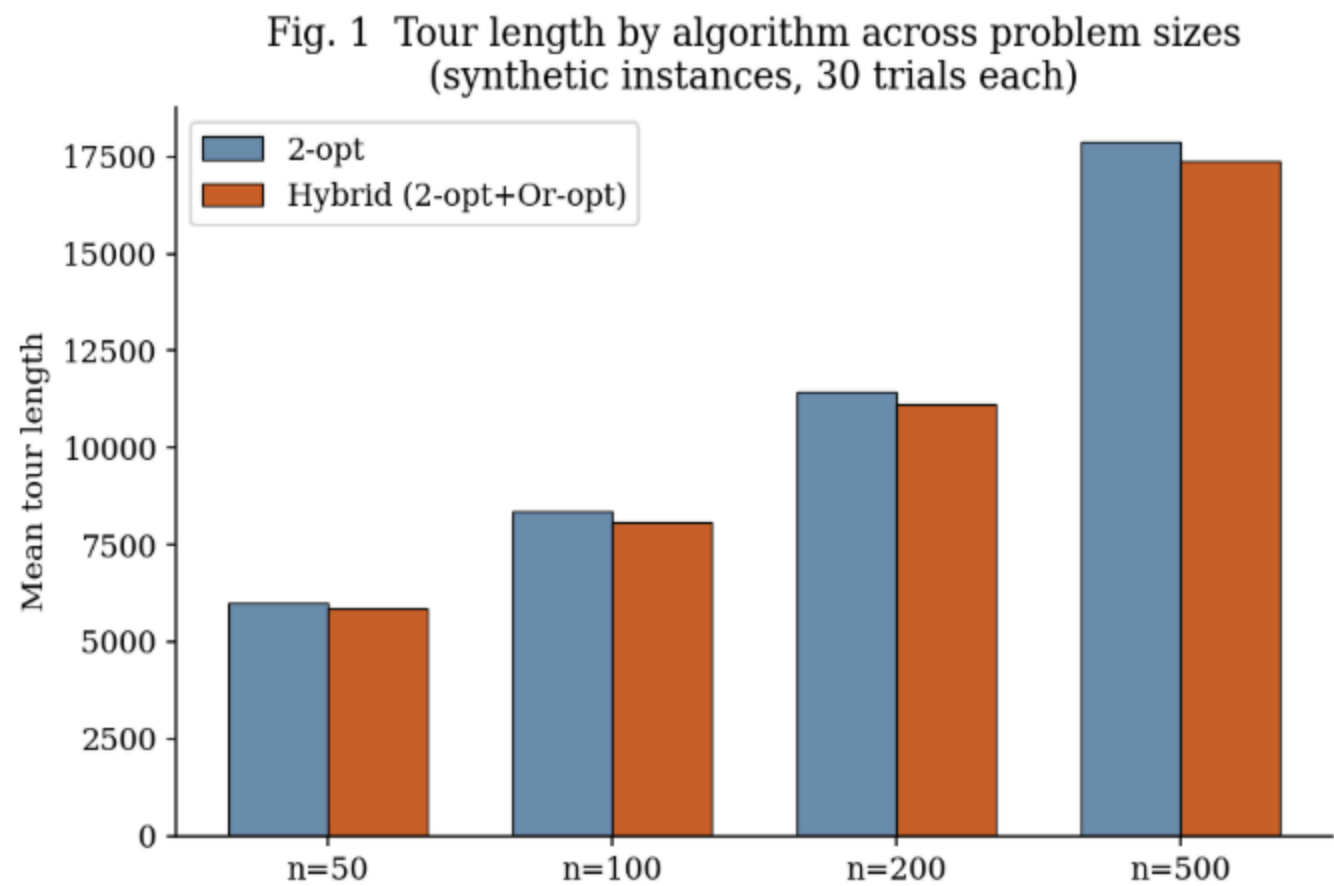


Figure 1

Mean tour length by algorithm across problem sizes (synthetic instances, 30 trials each).

Fig. 2 Percentage improvement of the hybrid pipeline over 2-opt, with 95% confidence intervals

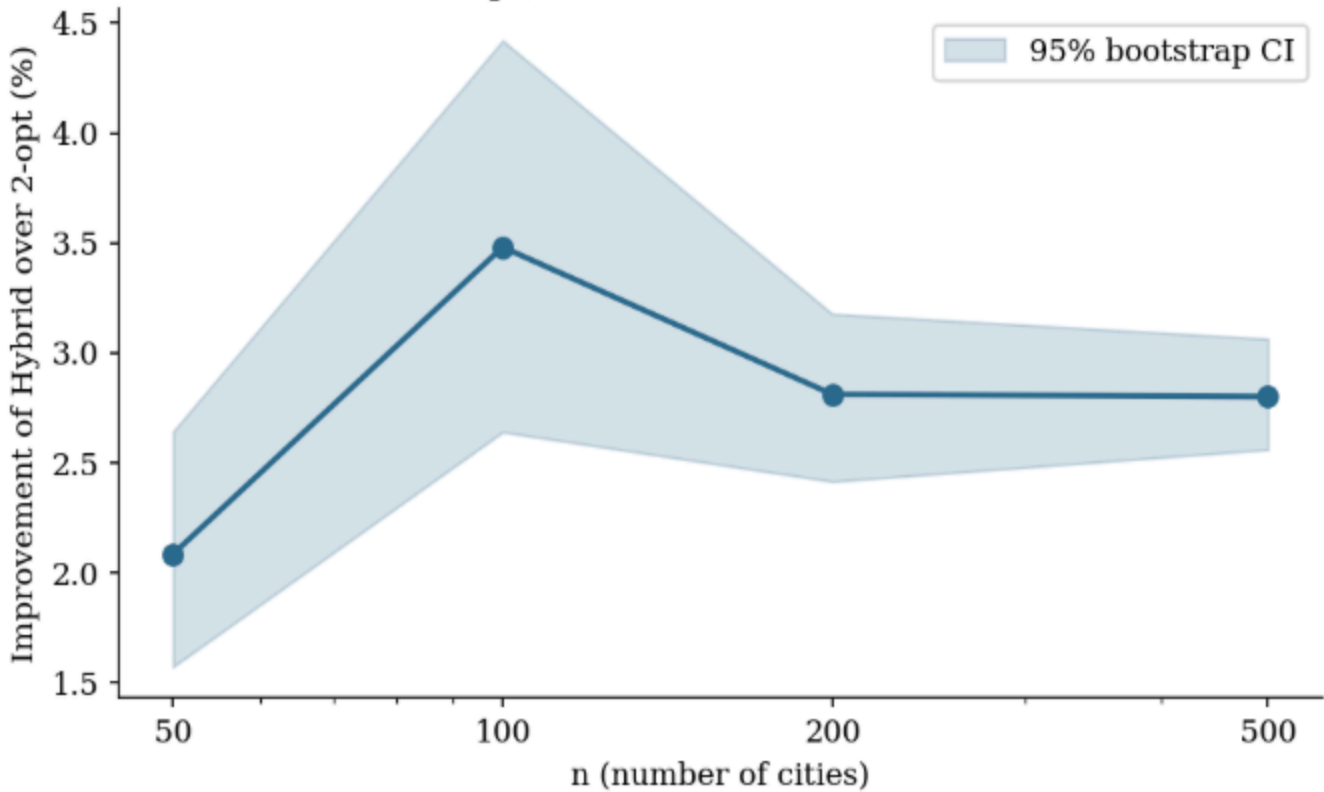


Figure 2

Percentage improvement of the hybrid pipeline over 2-opt, with 95% bootstrap confidence intervals.

Fig. 3 Optimality gap vs. published optimum
on four real TSPLIB instances

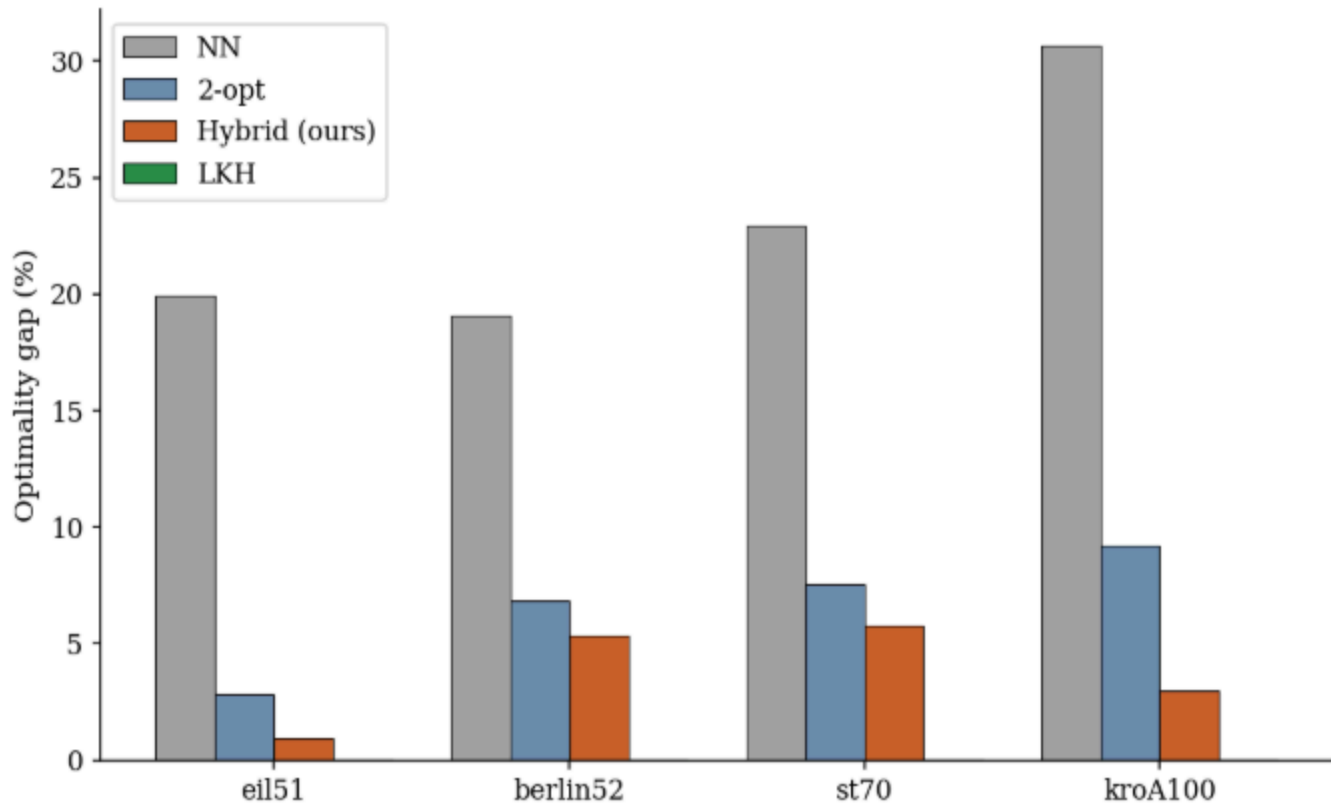


Figure 3

Optimality gap vs. published optimum on four real TSPLIB instances, including the LKH reference (near-zero gap in all cases).

Fig. 4 Distribution of tour lengths across 30 trials
(n=200, synthetic instances)

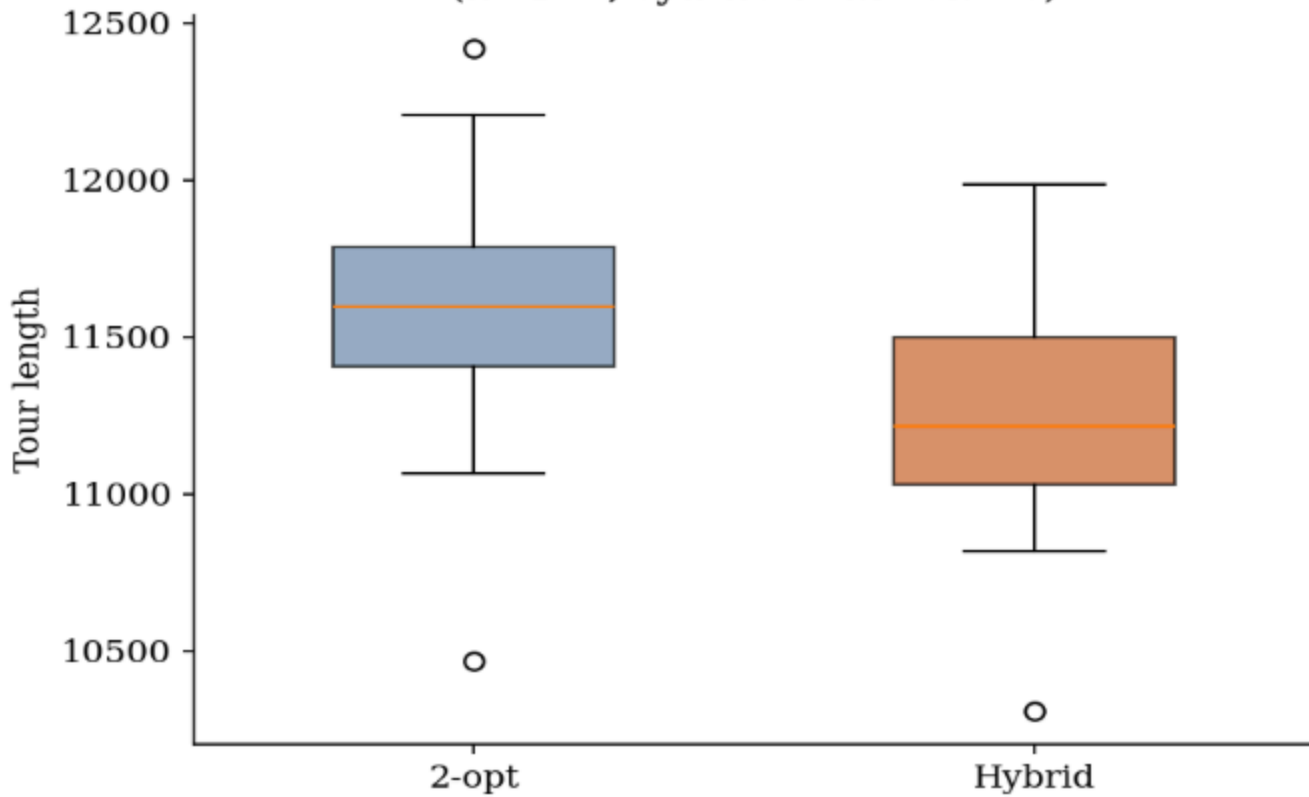


Figure 4

Distribution of tour lengths across 30 trials (n=200), illustrating the hybrid pipeline's consistently lower variance and median.