https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

OptiWindNet RouteSets: a solver-diverse benchmark dataset for the offshore wind-farm cable routing problem

Mauricio Souza de Alencar[1], Tuhfe Göçmen[1], and Nicolaos A. Cutululis[1]
[1]DTU Wind and Energy Systems, Frederiksborgvej 399, 4000 Roskilde, Denmark
Correspondence: Mauricio Souza de Alencar (masoda@dtu.dk)

Abstract. Offshore wind-farm collection-system design is a cost-relevant combinatorial optimization problem whose difficulty grows exponentially with turbine count. This paper introduces **OptiWindNet RouteSets**, a database of cable-routing solutions to 13 954 distinct problem instances based on built, proposed, and procedurally-generated farm layouts. The database is intended as an open benchmark for routing algorithms, a source of hard instances, a strong baseline for reinforcement-learning solvers, and a training corpus for supervised-learning models for the Wind Farm Cable-Routing Problem.

The solutions were produced using the `optiwindnet` Python package with three solvers: exact mathematical optimization and two meta-heuristics (inexact) – hybrid genetic search (HGS) and Lin–Kernighan–Helsgaun (LKH). Two network topologies are covered: radial (path-based) or branched (tree-based). Each solution contains the feasible network and its metadata (solver used, cable capacity, method configuration, route length, detour overhead, solver runtime, and, for exact runs, a proven optimality gap). The problem instances vary in: number of wind turbines ($\in [50, 200]$), maximum cable capacity ($\in [2, 12]$), and location geometry. For 63% of the problem instances, a solution with $< 1\%$ gap is available (and 79% with $< 2\%$ gap).

We analyze: (a) the total length of meta-heuristic solutions compared to their exact counterparts – HGS median increase is 0.0%, while LKH is 0.6%; (b) the instance difficulty as a function of capacity and turbine count – those two variables interact, the difficulty increases monotonically with count, but exhibits a count-dependent peak across capacities; (c) the length reduction of branched topology compared to radial – 0.34% median; (d) the detour-caused increase in length over the solver-optimized objective – 0.16% median.

1 Introduction

Cable-related costs account for roughly 11 % of the levelized cost of energy of an offshore wind power plant (BVG Associates, 2019). Minimizing the cost of the electrical network that collects power from the wind turbines toward the offshore substation is a combinatorial optimization problem well described in the literature (Pérez-Rúa and Cutululis, 2019), and we refer to it as the Wind Farm Cable-Routing Problem (WFCRP) from here on.

https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.
eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

The attention to the WFCRP is justified by the intractability of the exact mathematical optimization approaches for this NP-hard problem (Papadimitriou, 1978), meaning that no known polynomial-time algorithm can certify a solution is a global optimum. We addressed the complex interaction between geometric constraints and the discrete design space in our previous work (Souza de Alencar et al., 2026b), in which we proposed the *OptiWindNet* framework for the WFCRP. Its essential mechanism involves separating the electrical topology from its planar embedding and introducing detours in feeder cables that would otherwise cross cables laid out by the optimization itself.

Here we introduce *OptiWindNet RouteSets* release `26.05-v4`, a database of WFCRP problem instances and their *OptiWindNet*-generated solutions by different solvers and parameter settings. The dataset is aimed at advancing work on improved methods and providing informed guidance on the trade-offs among design choices, such as network topology and maximum cable capacity, and their effects on the network's total length and problem difficulty.

Analysis of this large empirical dataset can provide quantitative estimates of the effects of design choices and suggest new lines of inquiry. We report findings on four perspectives on the solutions and their metadata: (a) how do the meta-heuristics fare in total network length compared to the exact solver in paired problem instances?; (b) how do cable capacity and turbine count influence problem difficulty?; (c) how does the choice of branching the network on wind turbines or not affect network length?; and (d) how much increase in network length do detours cause?.

Our main contribution is the problem set accompanied by high-quality, metadata-rich, solver-diverse solutions. Analyses (a) and (d) are extensions of similar comparisons presented in (Souza de Alencar et al., 2026b), now including meta-heuristic LKH in (a) and a larger sample that also covers radial MILP solutions in (d). Analyses (b) and (c) are, to the best of our knowledge, novel contributions regarding the WFCRP.

The remainder of the paper is organized as follows. Section 2 describes how the database is organized, Section 3 reports the four perspectives on the data (inexact solution quality, problem difficulty, topology choice and detour impact), closing with conclusions and future inquiry suggestions in Section 4.

2 Data description

OptiWindNet RouteSets is distributed as a single SQLite file `optiwindnet-routesets-r26.05-v4.sqlite` (Souza de Alencar, 2026a) with four tables. The easiest way to use it is via Peewee ORM (Leifer, 2026) models in `optiwindnet.db.model`. The main table is *routeset*, whose rows represent WFCRP solutions. Each solution includes one reference to each of the other three tables: to a wind farm location from table *nodeset*; to a solver and its parameters from table *method*; and to the computer used to produce the solution in table *machine*. Table *machine* is trivial, containing the computer *name* and an optional *attr* field; next we detail the other three tables.

2.1 Locations – NodeSets

One row per distinct wind farm geometry. The SHA digest of the geometry is the primary key, so identical geometries collapse to a single row even if they were loaded from different source files.

https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.

All 2D coordinates are contained in a 2D matrix stored as a binary blob in field *VertexC*. The coordinates are planar and represent the easting and northing of Universal Transverse Mercator (UTM) geographical coordinates. The UTM zone is not stored, so a unique geodesic position cannot be obtained from this data alone. The role of each coordinate is defined by the other fields. Of particular relevance are the location *name* and its turbine count (T), substation (root) count (R), and how many coordinates only define boundaries (B).

The location description models turbines and substations as dimensionless points on the plane. Boundaries (site borders and exclusion zones) are represented as polygons. Solutions in the *routeset* table use indices to refer to coordinates in the *nodeset* row and contain no additional coordinates.

2.2 Augmented and transformed locations

The dataset has a core of 98 wind farm locations that represent actually built offshore wind farms. This set includes the vast majority of existing offshore wind farms in the turbine-count range of interest. Extending this core *built* group are five hypothetical locations *proposed* by Cazzaro and Pisinger (2022), Yi et al. (2019), and Taylor et al. (2023).

To make the dataset more useful for supervised machine learning model development, we used a data augmentation procedure (described in more detail in Souza De Alencar et al. (2024)). The core algorithms are implemented in `optiwind-net.augmentation`. The new location creation process involves retaining only the borders and substation positions of the actual sites and adding a varying number of turbines across the allowed area at random positions. Augmentation was used for creating single-substation locations, heavily skewing the dataset towards $R = 1$. The few multiple-substation locations represent built or proposed sites.

The *nodeset* field *name* of these new locations follows a convention: their string always start with an exclamation mark, the name of the donor location follows, then a second exclamation and a hash — `!`*`donor_name`*`!`*`digest`*. In addition, suffixes to the name segment indicate transformations were applied, such as reducing from multiple substations to a single one (`.1_OSS`, `.1st_OSS`) or removing the obstacles (`.solid`).

This family structure is important for the statistical use of the database: it allows distinguishing between within-family sensitivity to turbine placement variations and between-family generalization. In machine-learning studies, a family-level train-test split is therefore more stringent than a random split across nodesets, because randomized variants of the same site are geometrically related.

2.3 Solutions – RouteSets

A distinction between the connectivity and the embedding of a network is important for interpreting the rows in table *routeset*. The optimizer minimizes a weighted connectivity problem defined over the available-links graph, which abstracts the geometric details. The lowest total weight found is stored as the *objective* in the metadata, and, for MILP solvers, the best solution to the dual problem is stored as the *bound*, enabling the calculation of a proven relative gap, *relgap*. The solver certifies that the optimal solution to the problem is within that fraction of the best objective value found.

https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.

Table 1. Solvers used – *creator*, backend, topology support and route set counts

creator	backend(s)	topology	route sets	%
MILP.pyomo.cplex	CPLEX[a] 22.1.1 (IBM, 2024)	radial, branched	22 623	46.0
baselines.hgs	HGS-CVRP C library[b] (Vidal, 2022)	radial	13 857	28.2
baselines.lkh	LKH-3.0.14 binary from C implementation (Helsgaun, 2026)	radial	12 166	24.7
MILP.pyomo.gurobi	Gurobi[a] 12.0 (Gurobi Optimization, LLC, 2024)	radial, branched	558	1.1

[a] via `pyomo` (Bynum et al., 2021)

[b] via `hybgensea` (Souza de Alencar, 2026)

The WFCRP solution is the actual set of cable routes in the Euclidean plane that implements the connectivity found by the solver—the *route set* or *embedded network*. The recorded solution *length* is the sum of all physical cable segments after feeder detouring, which may match or exceed the *objective*. The field *detextra* represents the relative increase in total length from the *objective* value to the *length* value. The graph representing the solution before detouring is directly recoverable from the route set and therefore is not stored.

The field *creator* specifies which solver produced that *routeset* entry. Table 1 provides details for each possible value for this field. Note that *creator*, as well as T and R, are redundant information included in table *routeset* for convenience, as they can also be found by following the references to the corresponding *nodeset* and *method* rows.

2.4 Problem instance

A problem instance is a $(\mathit{nodeset}, \mathit{capacity})$ pair, where the *routeset* entry's *capacity* means the maximum cable capacity (in number of wind turbines). Multiple solutions in the *routeset* table may refer to the same problem instance. This is intentional, as the most valuable method comparisons are those that use paired results with a single method difference. Some analyses presented in the sections ahead rely on this kind of comparison.

For some analyses, it is best to use a database query that only uses the best solution (meaning the smallest *length*) for a problem instance among those that fit the desired characteristics (e.g., filtering only *topology is radial* might match three or more route sets that solve the same problem).

2.5 Method

This table stores detailed information about the solver call that generated the route set. It contains one row per unique (*solver_name*, *options*, *funhash*) triple. When adding a new route set to the database, *OptiWindNet* looks for a matching *method* entry already stored, which requires that triple to be exactly equal. However, the value of *funhash* (a hash of the function's byte-code) may, for the same solver, drift across *OptiWindNet* versions. This means that route sets created with the same solver call may refer to distinct method rows. Therefore, a database query using methods should refrain from using their primary key (*digest*) and search instead for the desired *solver_name* and set of (key, value) pairs in *options*.

https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.
Open Access
WIND ENERGY SCIENCE DISCUSSIONS

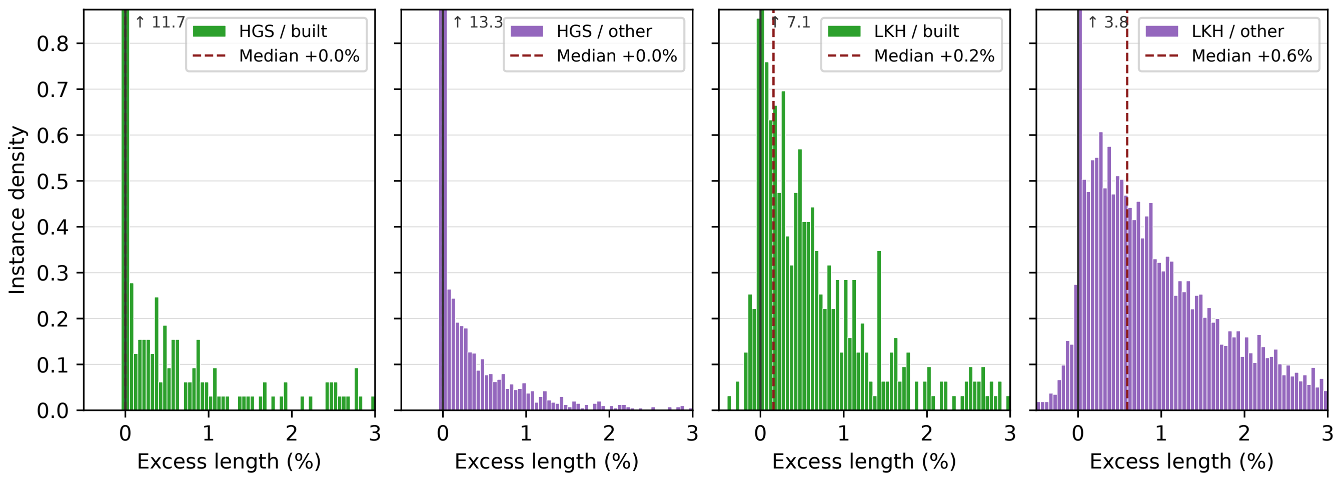

Figure 1. Meta-heuristics post-detours total length increase distribution relative to a MILP-radial reference. Separate distributions for actually built wind farms and all other problem instances.

3 Data analyses

3.1 Meta-heuristics solution quality

We compare the two meta-heuristic solvers against the MILP reference solution using paired problem instances. Both inexact methods were given 120 s of single-core time on the same CPU per problem. For each $(nodeset, capacity)$ pair, the shortest embedded MILP-radial route set is selected as the reference; the HGS-CVRP and LKH-3 route sets for the same pair are then compared to it. The comparison, therefore, isolates the solver while keeping the location geometry, cable capacity, and radial topology fixed.

The paired coverage is nearly complete for HGS-CVRP and slightly smaller for LKH-3: from 8729 MILP-radial reference instances, 8728 have an HGS-CVRP counterpart and 8501 have an LKH-3 counterpart. Figure 1 shows that HGS-CVRP is tightly concentrated around the MILP-radial reference, with a median increase of 0.0% and a 90th percentile of 0.225%. LKH-3 is also close, but with a broader positive tail: its median increase is 0.555% and its 90th percentile is 2.187%.

Negative differences are possible because the MILP route set used here is the shortest available embedded MILP-radial incumbent, regardless of the gap value. Moreover, these solutions were created with a requested gap of 0.5%, meaning that the solver interrupted the optimization once this value is reached. Furthermore, different route sets may require varying amounts of detouring, providing additional opportunities to improve the reference solution. They therefore indicate that a meta-heuristic found a shorter embedded route than the selected MILP incumbent for that pair, rather than that it surpassed a proven optimum. Overall, HGS-CVRP provides near-MILP-quality solutions for this database, while LKH-3 remains competitive but less tightly concentrated around the MILP-radial reference.

https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

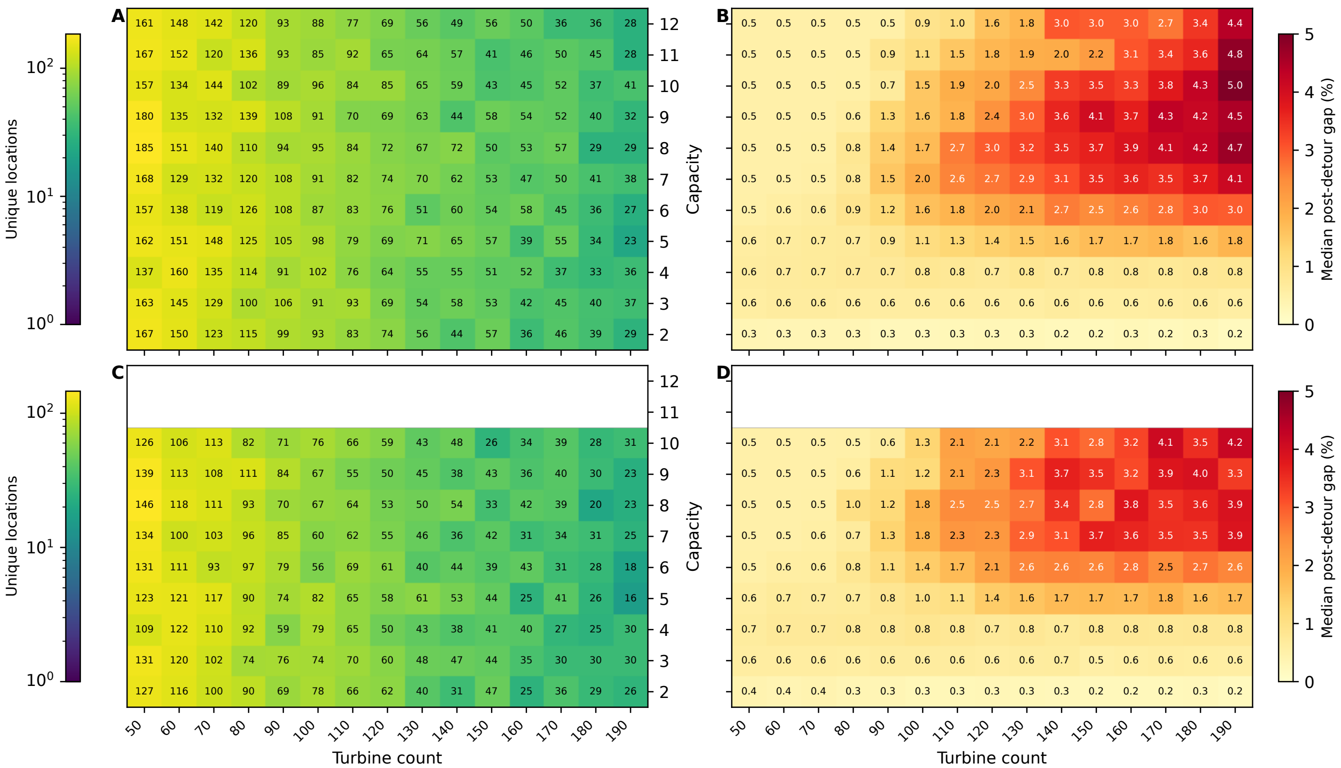

Figure 2. Panels A and C show the unique locations count across variables turbine-count and capacity for the post-detour median gap heatmap on panels B and D, respectively. A and B refer to branched topology, while C and D refer to radial.

3.2 Problem difficulty

We use two proxies from the MILP solver metadata to assess problem difficulty in this section: the relative distance from the solution's total length to the certified lower bound (post-detour gap), and the solver's total runtime. The gap works as a measure of difficulty only because the runtime was capped at 6 h. This means that harder instances caused the solver to terminate at the time limit, with the gap value it reached before then. In simpler cases, the solver terminated earlier, as soon as the requested 0.5 % gap was reached.

The difficulty, as evidenced by the median gap among solutions with a similar turbine count range (10-unit bin) and the same cable capacity, is shown in panels B and D of Figure 2. Panels A and C show how many unique locations in the database fall into the bins of the corresponding heatmats in panels B and D, respectively. The top row (A and B) refers to MILP branched solutions, and the bottom row (C and D) to MILP radial.

Locations are denser in the low-turbine-count region due to the design of the data augmentation procedure, which also aims to uniformly cover a range of turbine minimum spacing values. This range depends on the turbine count, since a higher count requires a smaller spacing to fit all the turbines. Hence, locations at the lower end of the turbine count display more diverse

https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.

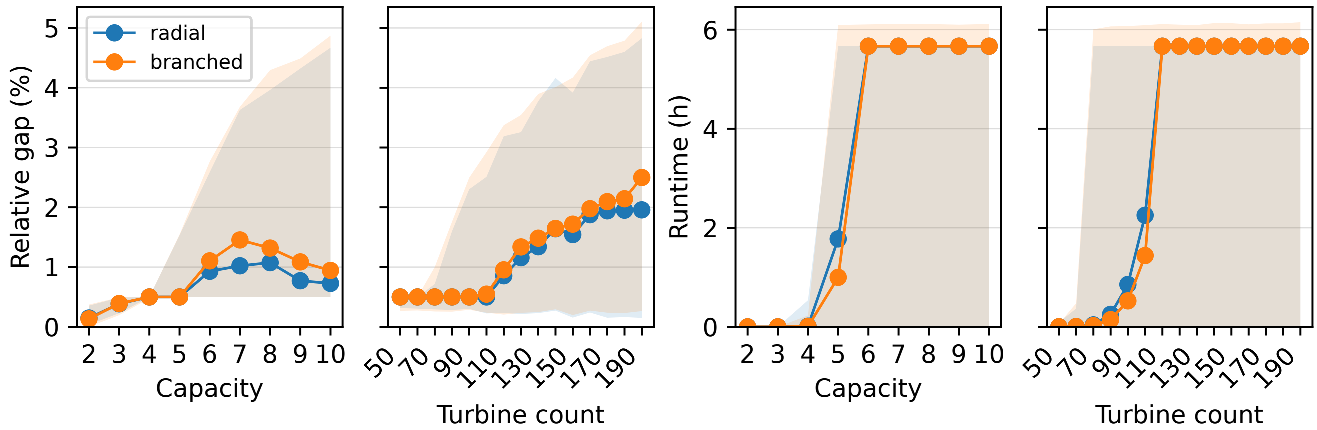

Figure 3. MILP difficulty measured by solver-reported relative gap and runtime. The four panels show the relative gap by Capacity, the relative gap by Turbine count, the runtime by Capacity, and the runtime by Turbine count. Lines are medians for radial and branched MILP; shaded bands show the 10th–90th percentile range. Buckets with $n \leq 5$ are omitted.

patterns of turbine placement, while at the higher end, placement resembles a somewhat regular pattern with some random jitter in the coordinates.

There is a clear tendency for the median gap to increase with problem size, but the relationship with capacity is non-monotonic. Capacities 2 and 3 are special cases: the MILP solver always finds the optimum solution quickly without even building a search tree. The median gap for capacity 3 is higher than the requested 0.5% solely due to detours.

Figure 3 shows the same data, now collapsed into the axes *capacity* and *turbine count*. It corroborates the analysis of the gap figures and, in the runtime perspective panel for turbine count, provides a clear view of the exponential growth in the search space. The transition from 70 to 110 turbines increases the median runtime from seconds to hitting the time cap. The spread of the values is large for both proxies, so there are probably other factors influencing the runtime and gap. The median gap is slightly but consistently higher in the 6-10 capacity range for branched than radial solutions, indicating that branched MILP instances are somewhat harder on average.

3.3 Branched vs. radial topology

For each topology, we select the shortest MILP route set per $(nodeset, capacity)$ pair and keep only problem instances for which both radial and branched MILP solutions are available. This yields 8696 matched pairs. Figure 4 first shows the solution-quality context for the comparison: the selected branched and radial MILP route sets have similar final gap distributions, with median gaps of $0.739\,\%$ and $0.725\,\%$, respectively. The topology comparison is therefore made between route sets with broadly comparable solution quality.

https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.

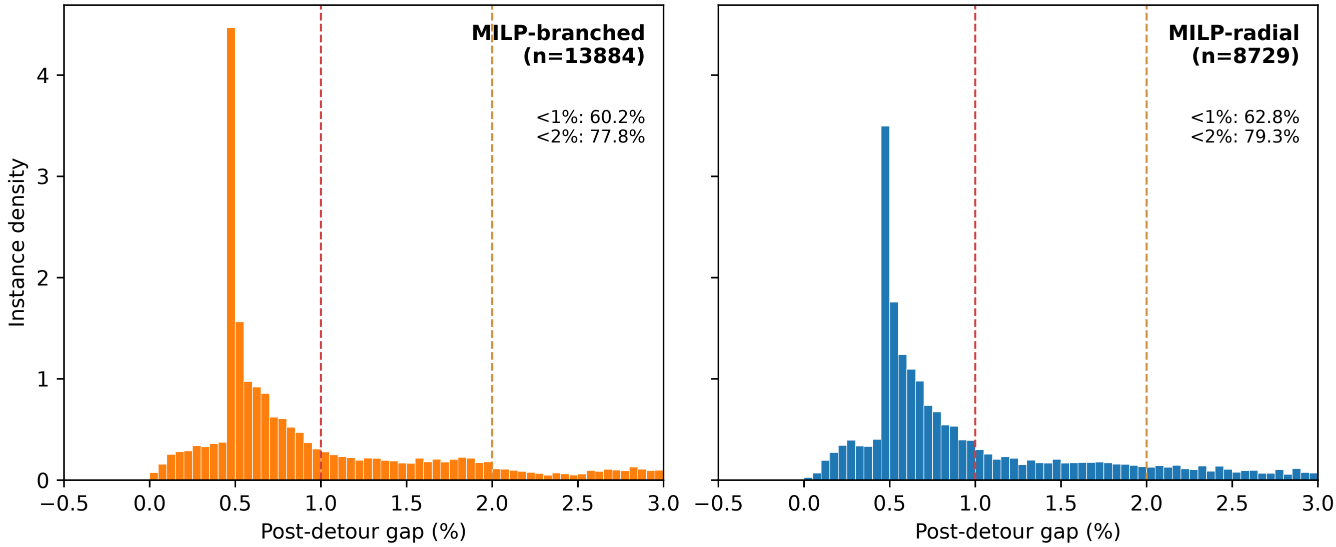

Figure 4. Post-detour gap distributions for MILP solutions grouped by topology.

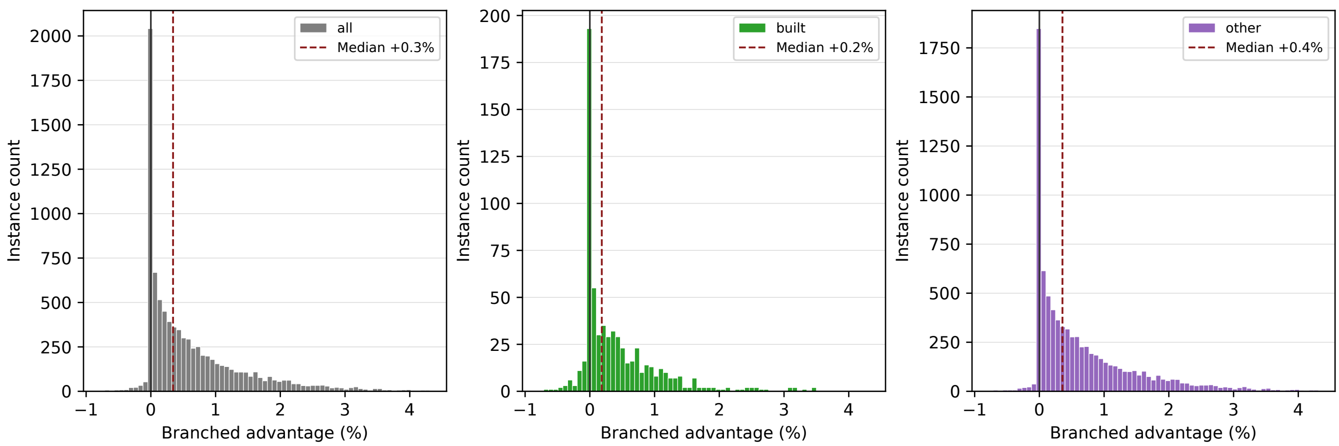

Figure 5. Distribution of relative reduction in detoured total length of each branched solution over their radial counterpart (positive means branched is shorter). Panels show all matched instances, built instances, and other instances.

Figure 5 shows that branched topology usually shortens the embedded network, but by a modest amount. Across all matched
160 instances, the median advantage is $0.34\,\%$ and the mean is $0.64\,\%$; branched route sets are shorter in 6741 of the 8696 matched cases. Built actual instances show a smaller median advantage ($0.18\,\%$) than the remaining cases ($0.36\,\%$).

The stratified results in Fig. 6 show that branching becomes more attractive for higher cable capacities. When looking across turbine counts, the relative advantage decreases with turbine count, but the high variance justifies examining trade-offs on a case-by-case basis.

https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.

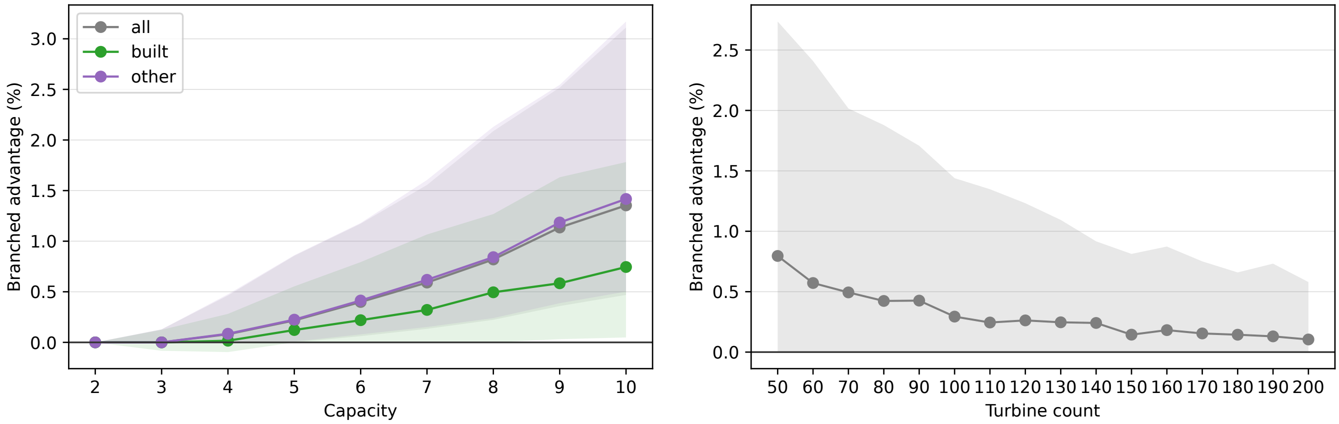

Figure 6. Branched advantage from branched MILP topology stratified by Capacity and Turbine count. Lines show the median and shaded bands show the 10th–90th percentile range. The Capacity panel is restricted to capacities up to 10, and the Turbine-count panel starts at 50 turbines.

Allowing feeder cables to branch at turbines can reduce the total network length, since the radial network problem is a constrained version of the branched problem. However, branching increases project complexity and requires additional electrical equipment at the turbines with more than two neighbors. The relevant question is therefore not whether branched layouts are geometrically shorter in isolation, but whether the cable-length reduction is large enough to justify the additional installation and component costs. The results of the paired comparison suggest the expected magnitude of the potential gain, which will inform the decision on using topology as one of the design variables in the economic optimization.

3.4 Detour impact

The solver objective is computed on the available-links graph before feeder detours are introduced. Detouring converts the optimized connectivity into a feasible embedded route set, and can therefore increase the final cable length relative to the value optimized by the MILP. Detour-caused increase is the relative increase from the solver-optimized objective to the final embedded route length.

The shortest route set is selected per $(nodeset, capacity, topology)$, and only rows with non-null Detour-caused increase are included. Figure 7 shows that this increase is small for most instances: the median is $0.171\,\%$ for radial topology and $0.154\,\%$ for branched topology.

Figure 8 and Figure 9 show that detour impact is largest at capacities 3-6 and decreases as capacity increases within the observed range, while the turbine-count dependence is weaker over the covered range. Radial solutions tend to be slightly more affected by detours, as expected given the greater number of connection options in branched networks. Still, impacts are overwhelmingly below 0.6% across all MILP solutions.

https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.

WIND ENERGY SCIENCE DISCUSSIONS
Open Access

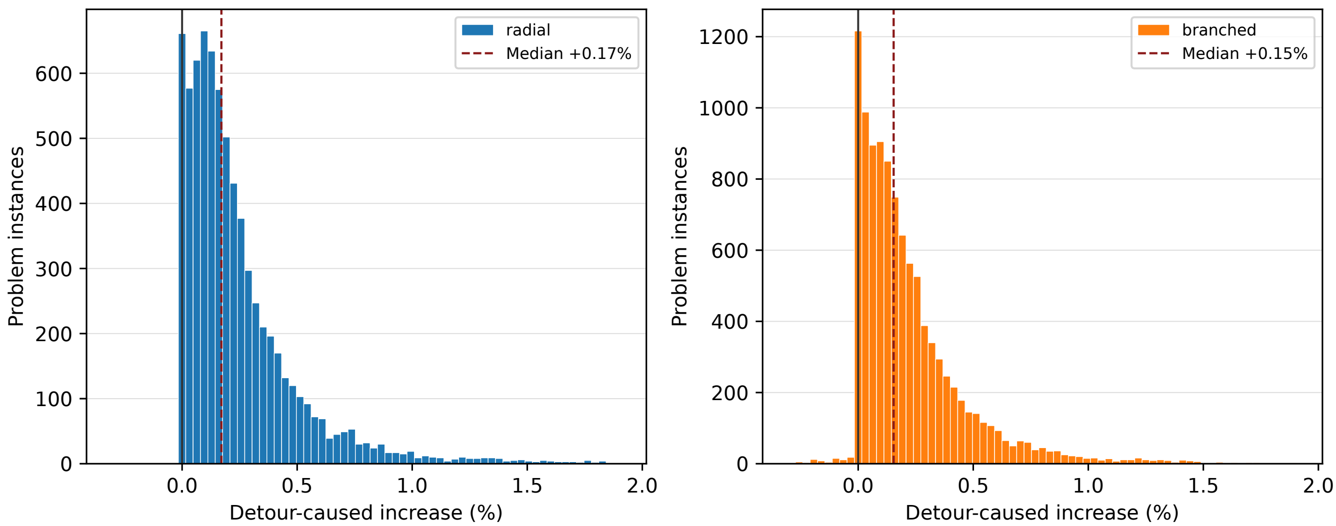

Figure 7. Distribution of detour-caused increase by topology.

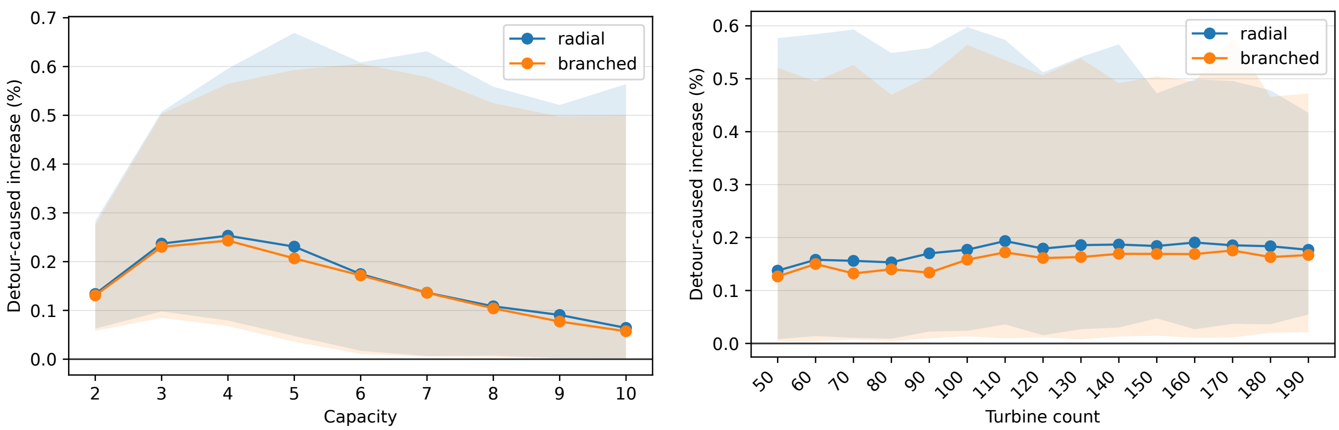

Figure 8. MILP Detour-caused increase stratified by Capacity and Turbine count. Lines show the median and shaded bands show the 10th–90th percentile range for radial and branched topology. The Capacity panel is restricted to capacities up to 10, and buckets with $n \leq 5$ are omitted.

4 Conclusion

We introduced *OptiWindNet RouteSets*, a database of 49 204 optimized WFCRP route sets over 12 814 wind-farm geometries and 13 954 distinct $(nodeset, capacity)$ problem instances. The release covers radial and branched network topologies, cable capacities from 2 to 12 turbines, and three solver families: exact MILP, HGS-CVRP and LKH-3. The solution networks, together with the solver parameters and output metadata, provide a reproducible benchmark for cable-routing algorithms and a labeled corpus for data-driven models of network cost and optimization difficulty.

https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.

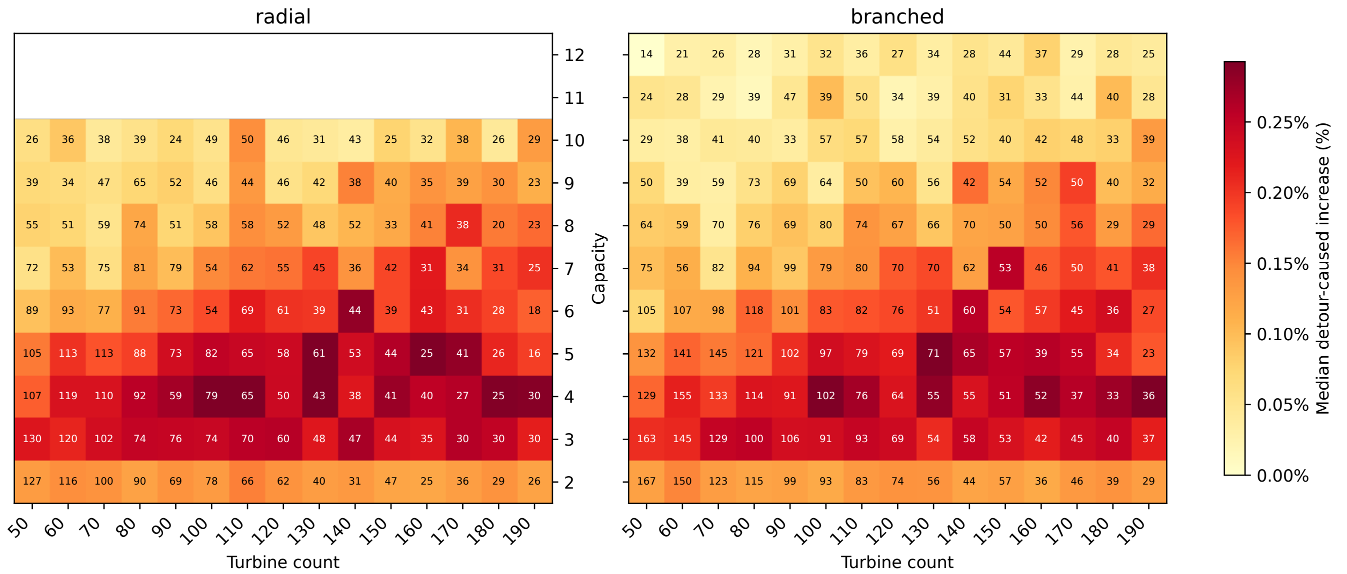

Figure 9. Median MILP Detour-caused increase by Capacity and Turbine count for radial and branched topology. Cell labels are support counts, and cells with $n \leq 5$ are hidden.

The paired comparisons show that the two meta-heuristics are competitive with the MILP-radial reference when evaluated on the same problem instances. HGS-CVRP is almost indistinguishable from the selected MILP-radial incumbents in most matched cases, while LKH-3 remains close but has a broader positive tail and a median increase of $0.555\,\%$. These results support the use of HGS-CVRP as a strong radial-topology solver for the WFCRP.

The MILP metadata also show that problem difficulty is not determined solely by turbine count. The median remaining gap increases with turbine count, but capacity has a non-monotonic effect: intermediate capacities create a larger set of competitive tree configurations, increasing the search effort required to close the gap. Branched MILP instances are somewhat harder than radial ones on average, especially in the mid-capacity range, and runtime increases sharply as problem size grows from 80 to 100 turbines.

The topology and embedding analyses quantify two practical design effects. Allowing branching usually reduces total embedded cable length, albeit modestly. The variance of that advantage, however, points to the need for project-specific weighing against the impacts of branching on equipment, installation and reliability. The study of length increases caused by feeder detours confirms the findings of our previous work (Souza de Alencar et al., 2026b) on a larger sample and extends the conclusion to radial topologies.

The analyses presented here are only a few of the many possible perspectives for exploring the dataset. Some interesting questions which the data may support answering are: how to predict the effort needed to reach a desired solution gap before starting the MILP solver?; can a model significantly reduce the search space (and, hence, problem difficulty) by predicting

© Author(s) 2026. CC BY 4.0 License.

WIND ENERGY SCIENCE DISCUSSIONS

which links are unlikely to be optimal and pruning them?; how to build and assess a quality certificate for an inexact solution based on empirical data instead of mathematical correctness?

The database should therefore be used as a structured experimental corpus rather than as a single leaderboard. Solver comparisons should be paired by $(nodeset, capacity)$ and matched by topology where relevant; machine-learning splits should respect site-family relationships. Within those limits, *OptiWindNet RouteSets* can be used as an open benchmark for WFCRP optimization, surrogate modeling and future algorithm development.

Code and data availability. The digital artifacts for the dataset and associated scripts are deposited on *Zenodo* under the CC-BY-4.0 license. The dataset is distributed as the single SQLite file `optiwindnet-routesets-r26.05-v4.sqlite` (Souza de Alencar, 2026a) and the scripts used to generate the figures presented here are deposited as an online citable supplement (Souza de Alencar, 2026b). The framework that produced the solutions is available as the Python package `optiwindnet` (Souza de Alencar et al., 2026a), and is distributed under the MIT license. It is also published in the *Python Package Index* (PyPI) and in the *conda-forge* repositories.

Author contributions. MSA: data curation, software, analysis, writing original draft. TG, NAC: supervision, writing, review and editing.

Competing interests. At least one of the (co-)authors is a member of the editorial board of Wind Energy Science.

Acknowledgements. This work was supported by Danmarks Frie Forskningsfond (grant no. 1127-00188B) under the project "Integrated Design of Offshore Wind Power Plants". Compute was provided by DTU's HPC cluster. Generative AI assistance was used for drafting and editorial revision; all scientific claims, data analysis, and final text were reviewed and revised by the authors.

https://doi.org/10.5194/wes-2026-124
Preprint. Discussion started: 28 July 2026
© Author(s) 2026. CC BY 4.0 License.

WIND ENERGY SCIENCE DISCUSSIONS

References

BVG Associates: Wind farm costs – Guide to an offshore wind farm, https://guidetoanoffshorewindfarm.com/wind-farm-costs, 2019.

Bynum, M. L., Hackebeil, G. A., Hart, W. E., Laird, C. D., Nicholson, B. L., Siirola, J. D., Watson, J.-P., and Woodruff, D. L.: Pyomo – Optimization Modeling in Python, vol. 67 of *Springer Optimization and Its Applications*, Springer, third edn., https://doi.org/10.1007/978-3-030-68928-5, 2021.

Cazzaro, D. and Pisinger, D.: Balanced Cable Routing for Offshore Wind Farms with Obstacles, Networks, 80, 386–406, https://doi.org/10.1002/net.22100, 2022.

Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual, https://docs.gurobi.com/projects/optimizer/en/12.0/index.html, 2024.

Helsgaun, K.: LKH-3: An Extension of LKH-2 for Solving Constrained Traveling Salesman and Vehicle Routing Problems, http://akira.ruc.dk/~keld/research/LKH-3/, accessed: 2026-05-03, 2026.

IBM: Mathematical Program Solvers - IBM CPLEX, https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer, 2024.

Leifer, C.: Peewee: a small, expressive ORM, https://github.com/coleifer/peewee, python ORM supporting SQLite, MySQL, MariaDB, and PostgreSQL. Accessed: 2026-05-03, 2026.

Papadimitriou, C. H.: The Complexity of the Capacitated Tree Problem, Networks, 8, 217–230, https://doi.org/10.1002/net.3230080306, 1978.

Pérez-Rúa, J.-A. and Cutululis, N. A.: Electrical Cable Optimization in Offshore Wind Farms—A Review, IEEE Access, 7, 85 796–85 811, https://doi.org/10.1109/ACCESS.2019.2925873, 2019.

Souza de Alencar, M.: OptiWindNet RouteSets, https://doi.org/10.5281/zenodo.20053479, 2026a.

Souza de Alencar, M.: Code and Computational Artifacts for the PhD Thesis "Wind Farm Collection System Optimization for Integrated Design", https://doi.org/10.5281/zenodo.20140812, 2026b.

Souza de Alencar, M.: HybGenSea: Python wrapper for HGS-CVRP, https://github.com/mdealencar/HybGenSea, python wrapper for the Hybrid Genetic Search solver for Capacitated Vehicle Routing Problems. Accessed: 2026-05-03, 2026.

Souza De Alencar, M., Göçmen, T., and Cutululis, N. A.: GNN-based Surrogate Modeling for Collection Systems Costs, Journal of Physics: Conference Series, 2767, 082 018, https://doi.org/10.1088/1742-6596/2767/8/082018, 2024.

Souza de Alencar, M., Arasteh, A., and Friis-Møller, M.: OptiWindNet by DTU Wind Energy, https://doi.org/10.5281/zenodo.18412871, 2026a.

Souza de Alencar, M., Göçmen, T., and Cutululis, N. A.: Flexible Cable Routing Framework for Wind Farm Collection System Optimization, European Journal of Operational Research, 329, 1037–1051, https://doi.org/10.1016/j.ejor.2025.07.069, 2026b.

Taylor, P., Yue, H., Campos-Gaona, D., Anaya-Lara, O., and Jia, C.: Wind Farm Array Cable Layout Optimisation for Complex Offshore Sites—A Decomposition Based Heuristic Approach, IET Renewable Power Generation, 17, 243–259, https://doi.org/10.1049/rpg2.12593, 2023.

Vidal, T.: Hybrid Genetic Search for the CVRP: Open-source Implementation and SWAP* Neighborhood, Computers & Operations Research, 140, 105 643, https://doi.org/10.1016/j.cor.2021.105643, 2022.

Yi, X., Scutariu, M., and Smith, K.: Optimisation of Offshore Wind Farm Inter-Array Collection System, IET Renewable Power Generation, 13, 1990–1999, https://doi.org/10.1049/iet-rpg.2018.5805, 2019.