Guiding CP Search for Recurring Vehicle Routing Problems with Learned Heuristics

Marc-Emmanuel Coupvent des Graviers, Jérémy Turi, Christophe Guettier
SAFRAN ELETRONICS & DEFENSE, MASSY, FRANCE
marc-emmanuel.des-graviers, jeremy.turi, christophe.guettier@safrangroup.com

Abstract

Recent research has explored neuro-symbolic methods that combine Machine Learning with traditional combinatorial optimization techniques to enhance problem solving efficiency. However, existing approaches remained limited in their ability to leveraged learned heuristics in their search mechanism. In this work, we introduce an integration schema for learned heuristic guidance with a new Constraint Programming (CP) based solving strategy called Iterative Domain Expansion With Initial Contraction (IDEWIC). We evaluate our approach on real road network extracted from OpenStreetMap. We show that our approach outperforms classical CP approaches and scales to larger instances.

Keywords: Neuro-symbolic, Recurring Vehicle Routing Problem, Constraint Programming, Heuristic Learning, Search Domain

1 Introduction

Recurrent routing problems arise in many real-world applications where vehicles repeatedly serve geographically distributed locations while minimizing travel distance or time. Because such instances often share structural patterns, they are natural candidates for machine learning. Neural Networks (NNs) can exploit these patterns to produce routing solutions, but they lack transparency, interpretability (Rudin 2019), and guarantees of correctness or optimality. Hybrid neuro-symbolic systems address these limitations by combining learning with rule-based solvers that enforce constraints and consistency.

Typical applications include daily mail delivery, waste collection, or fresh food distribution, where one or more vehicles drive on identical road networks each day while the visited destinations vary. Here we are interested in a particular version of single-vehicle routing problem, or Vehicle Routing Problems (VRP), which is defined as follows:

- The environment consists of a remanent road network represented as a graph $G = (V, E)$, where V is the set of vertices (intersections and locations), and E is the set of weighted edges representing roads with associated travel costs L (e.g., distance or travel time).

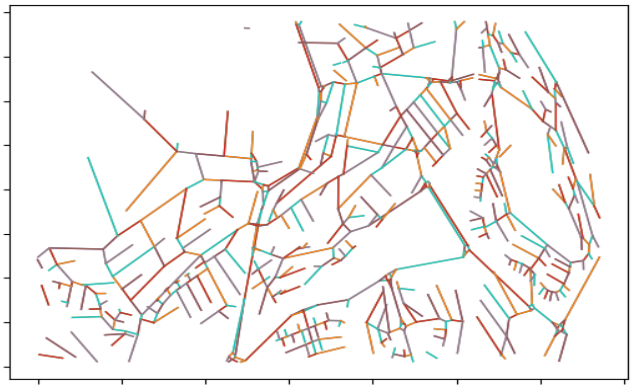

Figure 1: *Intersections graph extracted from the road network with 676 intersections and 3414 localized nodes.*

- Each day, a new subset of locations $M \subseteq V$ is selected for mandatory visitation, with the number of locations varying daily.
- A single vehicle must start from a fixed depot $start \in V$, visit all locations in M exactly once, and return to a fixed depot $end \in V$ while minimizing the total travel cost.

Previous approaches addressing VRP with machine learning often exhibit limited generalization through instance sizes, which consequently requires extensive solving time to create training datasets. The NNs output are also exploited by traditional algorithms such as beam search, which does not guaranty solution optimality, and requires additional algorithmic adaptation when industrial constraints are added in the problem definition.

This paper applies and enhances a hybrid approach using Graph Neural Networks (GNNs) similar to (Joshi et al. 2022), alongside CP solvers to tackle realistic recurring routing problems, while contributing with a new solver guidance mechanism: IDEWIC. The GNN component captures structures in the road network graph, while the CP solver ensures that the generated solutions satisfy all operational constraints. This hybridization enables efficient and trustworthy route optimization, combining data-driven learning with formal constraint reasoning.

2 Related Works

Routing problems (Braekers, Ramaekers, and Nieuwenhuyse 2016) are central in combinatorial optimization, with canonical variants such as the Traveling Salesman Problem Traveling Salesman Problem (TSP) (Steiglitz and Weiner 1968) and the Capacitated VRP Capacitated VRP (CVRP) (Dantzig and Ramser 1959). Both are NP-hard (Karp 1972a) and have been addressed using exact algorithms, heuristics, and metaheuristics (Held and Karp 1961; Nilsson 2003; Johnson and McGeoch 1997). These methods are effective on small to medium instances, but their computational cost grows quickly with problem size.

CP provides another established framework for routing problems, where variables and constraints define a search space explored through propagation and inference (Caseau 1997; O'Neil and Hoffman 2018). Modern CP solvers include techniques such as Conflict-Driven Clause Learning and Unit Clause Propagation (Silva, Lynce, and Malik 2009). However, both CP and classical optimization typically solve each instance independently and do not reuse experience from past solves. Moreover, search strategies combining multiple paradigms make variable and value selection difficult to interpret, which is problematic in real-scale applications (Siboulet et al. 2025) where search can be interrupted, optimization is incremental, or user interaction is required.

Hybrid neuro-symbolic approaches, combining the interpretability of CP with the adaptability of learning-based methods, have been explored (Osanlou et al. 2019), providing learned search heuristics while preserving exactness and completeness. Yet, challenges persist in transferring learned strategies to significantly larger or structurally different problem instances (Cantürk et al. 2024).

3 Search Guidance by Neural Network Scoring

Our method, IDEWIC, integrates a CP solver with a single-pass GNN inference through an ad-hoc problem formulation designed to efficiently find a route solution Ψ. We adopt a non-autoregressive approach: the GNN is run once to estimate edge scores $pred_{i,j}$, which are embedded in the CP model as guidance signals. Choosing CP instead of greedy or beam-search integration (Joshi et al. 2022) allows preserving correctness and optimality, while leveraging GNN predictions to improve solution quality. Our core contribution, IDEWIC, first reduces the search space to the most promising edges, then progressively expands it to guarantee completeness (i.e. Iterative Domain Expansion). The critical an unseen Initial Contraction phase, detailed later, determines the tightest feasible reduction from which expansion can efficiently start.

3.1 Constraint Programming Model

The problem is formulated as a successor-based model, where the solution is expressed through a successor function. Let $s(i)$ denote the successor of location $i \in M$ such that $s(i) = j$ means that location $j \in M$ is visited immediately after location i. With L being the travel costs matrix, the costs between a location and its successor become $L[i, s(i)]$. The solution validity is enforced by circuit constraint, and the optimal tour length is obtained by minimizing $D(s)$:

Variables:

$$s : M \to M \tag{1}$$

Assign the next location to each $i \in M$

Objective:

$$D(\Psi) = \min_s D(s) = \min_s \sum_{i \in M} L[i, s(i)] \tag{2}$$

Minimize total travel distance

Subject to:

$$\texttt{circuit}(s) \tag{3}$$

$$\forall\, i, j \in M,\ i \neq j: \quad C_{i,j} = (pred_{i,j} > Cut) \tag{4}$$

$$\forall\, i \in M: \quad C_{i,s(i)} \vee C_{s(i),i} \tag{5}$$

$$\forall\, i, j \in M,\ j > i: \quad \neg(C_{i,j} \vee C_{j,i}) \Rightarrow s(i) \neq j \wedge s(j) \neq i \tag{6}$$

The circuit constraint (3) ensures the sequence forms a valid tour (i.e. iterative following of $s(i)$ reaches every location). An edge (i, j) decision is reified as a candidate $C_{i,j}$ to be included in the solution if its score $pred_{i,j}$ is above Cut (4). Each edge between a vertex and its successor in the solution need to be among candidates (5) in either direction. Moreover, if a given edge is not candidate in both directions, then the associated vertex cannot be a successor of the other (6). The purpose of introducing Cut variable is described in following chapters.

We implemented this formulation with MiniZinc, using the OR-Tools CP-SAT solver (Perron and Furnon 2024) as backend.

3.2 Guiding the search

To guide the CP solver, we propose a new solving approach: *Iterative Domain Expansion With Initial Contraction (IDEWIC)*. The idea is to first reduce the search space by initially excluding edges whose predicted score falls below Cut, creating a contracted domain that leads the search on the most promising regions. Once, we found a suitable solution, we iteratively relax the parameter Cut, allowing the search space to expand again as needed to ensure completeness and optimality.

Initial Domain Contraction Cutoff-based guidance raises a feasibility issue: restricting the graph to high-scoring edges may eliminate all Hamiltonian routes, and checking this is NP-complete (Karp 1972b). To address this, we introduce an initial domain contraction phase that determines, for each instance, a conservative upper bound $Cut_{\max}$ on the cutoff value Cut, beyond which feasibility cannot be guaranteed.

The contraction process begins from the fully expanded domain in which all edges are available. The solver then iteratively increases the cutoff Cut, contracting the domain by excluding edges whose predicted score falls below this value. This contraction continues until the resulting domain

no longer supports a feasible Hamiltonian route. The largest cutoff for which a feasible solution still exists is recorded as $Cut_{\max}$. This phase is independent of route quality and serves solely to identify the boundary of the feasible contracted region. Because feasibility is guaranteed when the domain is fully expanded, a valid $Cut_{\max}$ is always obtained, even if the contraction process is halted early.

Iterative Domain Expansion Once $Cut_{\max}$ has been established, the solver enters the second phase, which performs path length minimization with complementary iterative domain expansion. The search begins with the contracted domain defined by $Cut_{\max}$, restricting exploration to the high-scoring edges predicted by the GNN. This initial contraction leads the solver to the most promising region of the domain, often allowing it to identify high quality solutions rapidly.

To guarantee optimality and completeness, the domain is then gradually expanded. The cutoff Cut is iteratively relaxed, reintroducing edges that were previously excluded due to lower predicted scores.

Implementation We implemented this strategy via a score-cutoff parameter Cut and associated constraints:

- In the first phase of *Initial Contraction*, the solver is tasked to maximize Cut while its search strategy is constrained with `int_search([Cut], indomain_min)`, which forces the solver to start with low values of Cut. By limiting the solver time in this phase, we ensure that the search remain in areas where the solver finds Hamiltonian path in reasonable time.
- In the second phase of *Domain Expansion*, the solver is tasked to minimize route length while its search strategy is constrained with `int_search([Cut], indomain_max)` and $Cut \leq Cut_{\max}$. Expansion is controlled through MiniZinc's `indomain_max` strategy applied to Cut, which prioritizes high-cutoff configurations early in the search and delays the inclusion of low-scoring edges. As Cut eventually reaches zero, the full domain is restored, ensuring that the global optimum is always recoverable regardless of the accuracy of the GNN predictions.

Together, iterative domain contraction and expansion balance efficiency and robustness: the solver benefits from targeted search when neural guidance is reliable, yet retains complete exploration when needed. The integration of this cutoff-based domain contraction/expansion with MiniZinc's search annotations keeps the approach solver-agnostic and enables consistent evaluation across different CP backends.

4 Preliminary Results

Solver performance is evaluated on multiple problem instances, but presented only for instances of 80 mandatory waypoints, which is out of our learning instances size range (from 10 to 30). However those instances differ in their optimal path length. To enable meaningful comparisons across instances of heterogeneous optimal length, solution quality is assessed using the normalized gap to optimality, defined in (7).

$$Performance = \frac{D(\Psi) - D(\Psi_{Opt})}{D(\Psi_{Opt})} \tag{7}$$

Here, $D(\Psi)$ denotes the length of the candidate solution and $D(\Psi_{Opt})$ the length of the optimal solution. For performance assessment, $D(\Psi_{Opt})$ is obtained through multi-threaded solving without time limitation, ensuring that the benchmark values represent true optima. Since this setup guarantees that the solver will eventually reach the optimal solution (i.e., with zero gap to optimality), our primary interest lies not in the final outcome but in the solver's search dynamics.

Accordingly, we focus on the effect of guidance by tracking the quality of intermediate solutions over solving time, thereby characterizing specifically the convergence behavior of the guided search.

4.1 Initial Domain Contraction Results

The effectiveness of the *Domain Contraction* phase is evaluated based on its ability to produce a valid upper bound value for $Cut_{\max}$ within a reasonable time. This bound should be as high as possible to minimize the number of edges considered during the initial stages of the guided search, while still ensuring the existence of a Hamiltonian route within the selected subset. The highest possible value, denoted Cut_{opt}, corresponds to the maximum cutoff that still permits such a path.

We conducted experiments on instances generated within the Reykjavik city area (see Figure 1). We measured the evolution of the best valid cutoff as a function of solving time depicted in Figure 2. Interestingly, the evolution of the valid cutoff over time consistently exhibits a two-phase behavior. Cutoff values quickly improve in the first phase. The slope is consistent for every instance, but depends on instance size as checked independently. Further improvements become increasingly costly in the second phase, where cutoff gains diminish significantly and eventually saturate. This phenomenon can be explained post hoc as follows: as the cutoff increases, the number of considered edges decreases, thereby reducing the search space and making it easier to find a Hamiltonian circuit in the early stages. Once the edge density drops below a critical point, however, the graph becomes sparse enough that the existence of a Hamiltonian circuit is no longer structurally guaranteed. At this stage, increasing the cutoff value becomes significantly harder and requires extensive backtracking, which leads to the observed exponential growth in computation time. This observation highlights the central role of the initial domain contraction: its purpose is precisely to configure the subsequent optimization within a reduced search space, while still keeping the problem in a state where finding valid solutions remains relatively easy. As a result, although the optimal value Cut_{opt} may be theoretically unreachable within practical time limits, a near-optimal and valid upper bound $Cut_{\max}$ can be reliably computed within a predictable and accept-

able timeframe, which is required for real-time or resource-constrained applications.

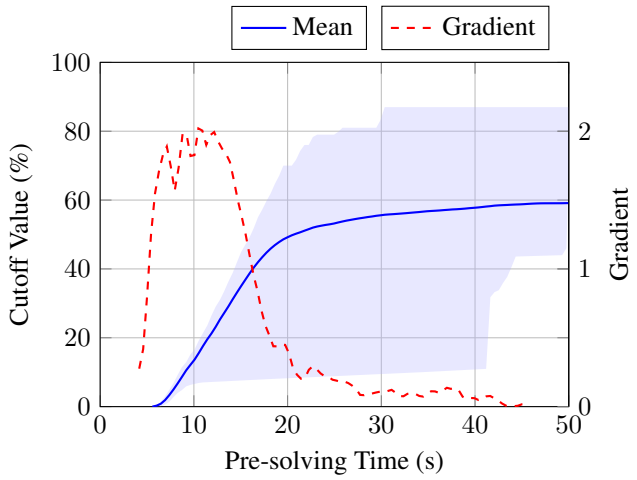

Figure 2: *Cutoff bound, according to pre-solving time, for 100 instances embedding 80 mandatory points in Reykjavik. The complete range (from min to max), average value and gradient of intermediate cutoff value are represented.*

4.2 Iterative Domain Expansion Results

To evaluate the effectiveness of the *Domain Expansion* phase, we compare its intermediate results with those of the unguided solver over time. We evaluated our method on the Reykjavik area. Figure 3 reports solution quality trajectories for both solvers on 100 problem instances with 80 waypoints. Because guided search begins once the initial domain contraction has established a valid threshold, guided results appear after initial domain contraction (25 seconds).

Once initiated the guided solver rapidly outperforms the unguided baseline across all instances. This sharp improvement demonstrates that neural network guidance effectively steers the search toward promising edges, yielding higher-quality solutions more quickly overall. Using instance of 80 waypoints also demonstrates that the advantage of guided search is efficient well beyond training distribution, reinforcing the practical value of our approach.

5 Conclusion

We presented a hybrid neuro-symbolic approach combining machine learning with CP to address recurring routing problems on real road networks. The key contribution allows to speed up solving by improving the solver ability to navigate the search space. For that, we propose a two phase approach, namely IDEWIC: an initial search domain reduction followed by search domain expansions during an iterative solution optimization. This is possible because a GNN is able to score the most promising edges for a given routing instance, and this by learning from past experiences, instead of treating each problem independently as in classical operational research or CP approaches.

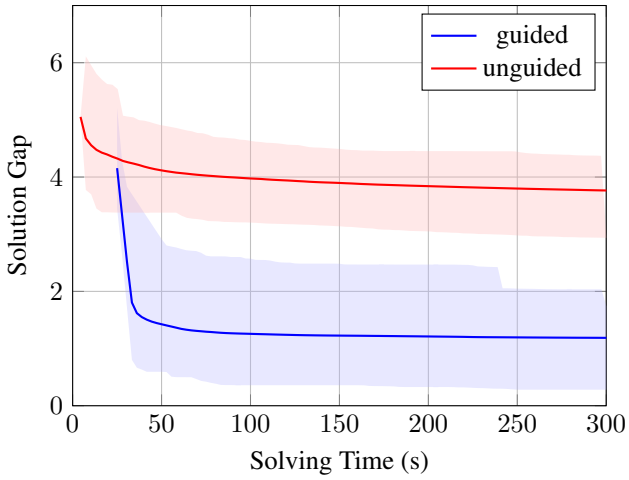

Figure 3: *Results for both guided and unguided solving for instances of 80 waypoints in Reykjavik city area for 100 problem instances. The average and complete range (i.e. from min to max) of intermediate solution quality are represented. Intermediate solution for guided solver starts after 25 seconds of initial domain contraction*

The results show that IDEWIC systematically improves intermediate solution quality, even with the overhead of the first *Domain Contraction* phase. Experimental results suggest that the IDEWIC approach provides consistent improvements over the unguided baseline. The method appears to scale effectively to larger problem instances and to generalize across multiple geographic areas. Taken together, these findings indicate that neural network guidance can enhance solver efficiency and solution quality beyond its training distribution, while maintaining robustness across diverse settings. Integration of the guidance can be directly done at the modelling level, without requiring complex search strategy at the solver level, making the approach easier to implement for other use cases. However, this approach strongly relies on the ability of the GNN to provide accurate edge scoring.

Overall, this hybrid approach provides simultaneously practical performance improvements and strong theoretical guarantees: generated solutions remain valid according to problem definition, and optimality can always be reached given enough time in contrast with purely ML-based approaches where solutions do not yield guarantees. This is a step toward more reliable and transferable ML-guided optimization methods in realistic routing contexts. While the single-vehicle use case is a reasonable first step, practical logistics applications typically involve fleets of vehicles (CVRP-like problems). Therefore, extending the current approach in this way seems like a natural step. There properties are critical in high-stakes domains such as logistics, finance, and industrial automation (Varshney and Alemzadeh 2016), and are explicitly highlighted in the European Union AI Act, which mandates transparency and interpretability for high-risk AI systems (Article 13)[1].

[1]https://artificialintelligenceact.eu/article/13/

6 Acknowledgements

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them. This work was supported by the European Union under the EDF Project FaRADAI (grant number 101103386).

References

Braekers, K.; Ramaekers, K.; and Nieuwenhuyse, I. V. 2016. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99: 300–313.

Cantürk, F.; Varol, T.; Aydoğan, R.; and Özener, O. O. 2024. Scalable Primal Heuristics Using Graph Neural Networks for Combinatorial Optimization. *J. Artif. Int. Res.*, 80.

Caseau, L. 1997. *Solving Small TSPs with Constraints*, 316–330.

Dantzig, G. B.; and Ramser, J. H. 1959. The Truck Dispatching Problem. *Management Science*, 6(1): 80–91.

Held, M.; and Karp, R. M. 1961. A dynamic programming approach to sequencing problems. In *Proceedings of the 1961 16th ACM National Meeting*, ACM '61, 71.201–71.204. New York, NY, USA: Association for Computing Machinery. ISBN 9781450373883.

Johnson, D. S.; and McGeoch, L. A. 1997. The traveling salesman problem: A case study in local optimization. *Local search in combinatorial optimization*, 1(1): 215–310.

Joshi, C. K.; Cappart, Q.; Rousseau, L.-M.; and Laurent, T. 2022. Learning the Travelling Salesperson Problem Requires Rethinking Generalization. ArXiv:2006.07054 [cs, stat].

Karp, R. 1972a. Reducibility Among Combinatorial Problems. volume 40, 85–103. ISBN 978-3-540-68274-5.

Karp, R. M. 1972b. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, 85–103. Springer.

Nilsson, C. 2003. Heuristics for the Traveling Salesman Problem.

Osanlou, K.; Bursuc, A.; Guettier, C.; Cazenave, T.; and Jacopin, E. 2019. Optimal Solving of Constrained Path-Planning Problems with Graph Convolutional Networks and Optimized Tree Search. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3519–3525. Macau, China: IEEE. ISBN 978-1-72814-004-9.

O'Neil, R. J.; and Hoffman, K. 2018. Exact methods for solving traveling salesman problems with pickup and delivery in real time. *Optimization-Online. org: http://www. optimization-online. org/DB_HTML/2017/12/6370. html. Accessed*, 9.

Perron, L.; and Furnon, V. 2024. OR-Tools.

Rudin, C. 2019. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. arXiv:1811.10154.

Siboulet, E.; Godet, R.; Bit-Monnot, A.; Graviers, M.-E. C. D.; Guettier, C.; and Lacroix, S. 2025. A Flow Based Planning Method for Multi-Agent Progression with Deployable Agents and Communication Constraints. *Proceedings of the International Conference on Automated Planning and Scheduling*, 35(1): 348–357.

Silva, J.; Lynce, I.; and Malik, S. 2009. Conflict-Driven Clause Learning SAT Solvers. *Frontiers in Artificial Intelligence and Applications*, 185.

Steiglitz, K.; and Weiner, P. 1968. Some Improved Algorithms for Computer Solution of the Traveling Salesman Problem.

Varshney, K.; and Alemzadeh, H. 2016. On the Safety of Machine Learning: Cyber-Physical Systems, Decision Sciences, and Data Products. *Big data*, 5.