Hybrid Search with Graph Neural Networks for Constraint-Based Navigation Planning

Marc-Emmanuel Coupvent des Graviers,[1] Kevin Osanlou,[2] Christophe Guettier,[1] Tristan Cazenave [2]

[1] Safran Electronics & Defense
[2] LAMSADE, Université Paris Dauphine - PSL, CNRS, Paris, France
marc-emmanuel.des-graviers@safrangroup.com

Abstract

Route planning for autonomous vehicles is a challenging task, especially in dense road networks with multiple delivery points. Additional external constraints can quickly add overhead to this already-difficult problem that often requires prompt, on-the-fly decisions. This work introduces a hybrid method combining machine learning and Constraint Programming (CP) to improve search performance. A new message passing-based graph neural network tailored to constraint solving and global search is defined. Once trained, a single neural network inference is enough to guide CP search while ensuring solution optimality. Large-scale experiments using real road networks from cities worldwide are presented. The hybrid method is effective in solving complex routing problems, addressing larger problems than those used for model training.

Introduction

Many real-world routing situations, such as goods delivery, industrial logistics or medical transports involve driving on the same road network repeatedly, day after day. This is particularly the case for autonomous driving vehicles, where on-line planning and scheduling is embedded to deal with frequent missions updates and scarce on-board resources (batteries, fuel, cargo space…).

The capability to learn from past experiences on the long term is an interesting human skill. Mastering this capability can greatly benefit artificial intelligence, including to tackle automated planning and scheduling problems. Machine learning is a powerful asset that is increasingly used in this domain. Not only it makes learning from past routing and planning experience possible, but it also generalizes learned information from previously-experienced situations to similar, yet unseen, situations. Recently, alongside major successes in natural language processing and computer vision, deep learning techniques have been applied to graph-structured data (Chen et al. 2020). More specifically, Graph Neural Networks (GNN) and their variants (Cappart et al. 2021) have been investigated for routing problems. Once trained, these networks can be hybridized with operational research algorithms such as beam search, Monte Carlo Tree Search (Mazyavkina et al. 2021), local search and CP (Osanlou et al. 2019).

However, existing GNN-based methods have the following different numbered downsides. **(1)** Numerous works use randomly generated TSP datasets for training and validation, while limited attention is given to actual performance on real-world graphs. **(2)** Although GNNs are agnostic to graph size (i.e.the same GNN can process graphs with any number of nodes and connections), several works only consider datasets with fixed-size graph. **(3)** Generalization to larger problems is often considered as a side topic. **(4)** In auto-regressive approaches, necessary computer resources often make embedding into autonomous vehicles infeasible. **(5)** Lastly, most of the search algorithms hybridized with GNNs can neither guarantee proof of completeness nor optimal solution.

The hybrid architecture with GNN and CP, proposed in the work, addresses the above issues. The GNN is independent of input graph dimensions, connectivity, and problem instance size. It is trained and validated on several real-world graphs. The GNN output is used as a non-auto-regressive probe for incremental variable selection. We also provide a constraint based formulation of this incremental search strategy. The resulting model offers a significant increase in performance, notably on intermediate solutions found for problem instances larger than those encountered during GNN training.

Our Approach

Our approach defines a hybrid solving architecture based on learning and constraint solving. At first, a GNN is introduced to learn search features as edge scores over the graph, using several problem instances. Then, the GNN outputs are used as a non-auto-regressive probe for incremental variable selection with a constraint propagation schema. The CP search exploit those scores to formally build the exploration strategy. This strategy is expressed using a pure constraint based formulation. Similar hybridisation schema combining search and learning is reported in (Peng, Choi, and Xu 2021), that strongly decouples the search algorithm from the learning pre-processing. However, incremental problem solving has not been considered so far.

This hybrid approach can be summarized as:

1. Select a real road network graph from OpenStreetMap and extract a connected sub-graph.

2. Generate training, testing and validation dataset instances by randomly selecting mandatory points.
3. Annotate datasets without supervision with exact problem solving.
4. Train a GNN to score each edge between mandatory points, with high scores provided to edges belonging in the optimal order.
5. Use the GNN to infer edge scores in one inference pass (i.e. non-autoregressive).
6. Use edge scores to guide a CP solver to find the optimal order faster.
7. Compare performance of unguided and GNN-guided constraint model on the validation dataset .

Our work differs from related works mostly because of the following assumptions :

Hypothesis 1: *Optimally solving a set of simple problem instances in road network graph may be used to learn how to guide a solver on new bigger instances in the same graph.*

Keeping the same graph can be seen a strong limitation in some applications. However, the scenarios we face involve slowly evolving graphs and a large set of missions defined inside each graph. The cost of training a GNN on a given graph is compensated by numerous mission planning requests in the same graph; and the overall cost is expected to be lower than if solving every mission ignoring past experience.

Hypothesis 2: *GNNs with Sigmoid activation works better than GCN with ReLU for route planning.*

GCN embeds node-wise normalization by neighbor number, which is inconsistent with existing handcrafted shortest path finding algorithms.

Hypothesis 3: *Learning in non-complete road graphs is more efficient than learning in the graph of mandatory points*

We refrain from seeing the problem only as a TSP variant. We assume partial connections between road intersections embed richer topological information than a fully connected TSP between mandatory points.

Hypothesis 4: *Shortest path cost to start and end nodes can be used as input vertex features, instead of absolute vertex position.*

This makes our approach invariant by translation and rotation. However, one drawback is that transfer learning using pre-trained GNN models is not feasible in this context.

Hypothesis 5: *Backpropagation should apply only to mandatory vertex features during model training.*

We are interested in scoring edges among mandatory vertices only. Backpropagating through non-mandatory vertices will add unnecessary noise to the training process.

Hypothesis 6: *Solution correctness and optimality can be guaranteed using CP instead of beam search.*

In our CP approach, GNN guidance impacts solving time only. Moreover, the CP paradigm eases future adaptation to problem variations, compared to ad-hoc handcrafted solving algorithms.

Discussion and Further Works

We propose a hybrid solving approach for single-vehicle route planning, combining GNN learning with a CP search algorithm. The approach can cope with recurrent route planing problems of interest for autonomous vehicles. We use an edge-wise GNN architecture, agnostic to graph order and problem size. Several hyper parameters have been investigated during training. An interruptible and incremental search method is proposed, and formalized with constraints. It exploits edge scores from GNN inferences to select variables within the search tree. Experiments have been carried out over real world data for both training and testing purposes. The 6 presented hypothesises have been verified in practice, exhibiting the feasibility of the approach.

On real scale problems, it is shown that the learning schema generalizes in the following three ways. Test and train losses are similar with no visible overfitting. The GNN can be trained on different static graphs using similar hyper-parameters. It also infers properly on problem sizes larger than the training ones. As a result, the GNN can be used as an efficient probe to guide the search algorithm, even on large scale instances. The complete CP search algorithm, guided by the GNN model, avoids heavy tuning requirements of meta-heuristics and related hyperparameters. Lastly, the search neither requires sophisticated restarts nor other intrusive mechanisms (backjumping or hooks). Other advantages have been discussed in terms of data frugality and trustworthiness for embedding purposes. Annotated data and implementation extracts are made available on Github [1].

As a follow up to this research, future works could also involve extensions to multi-agent planning problems, with specific constraint-like coordination, taking into account time-windows.

References

Cappart, Quentin et al. (Apr. 2021). *Combinatorial optimization and reasoning with graph neural networks*. en. arXiv:2102.09544 [cs, math, stat].

Chen, Fenxiao et al. (2020). "Graph Representation Learning: A Survey". en. In: *APSIPA Transactions on Signal and Information Processing* 9.1. arXiv:1909.00958 [cs, stat]. ISSN: 2048-7703, 2048-7703.

Mazyavkina, Nina et al. (2021). "Reinforcement learning for combinatorial optimization: A survey". In: *Computers & Operations Research* 134, p. 105400.

Osanlou, Kevin et al. (Nov. 2019). "Optimal Solving of Constrained Path-Planning Problems with Graph Convolutional Networks and Optimized Tree Search". en. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Macau, China: IEEE, pp. 3519–3525. ISBN: 978-1-72814-004-9.

Peng, Yun, Byron Choi, and Jianliang Xu (June 2021). "Graph Learning for Combinatorial Optimization: A Survey of State-of-the-Art". en. In: *Data Science and Engineering* 6.2, pp. 119–141. ISSN: 2364-1185, 2364-1541.

[1] see https://github.com/oihojehtuohfdsgpokzre/cp-gnn