Aalto-yliopisto
Perustieteiden korkeakoulu

Predicting Solutions for the Vehicle Routing Problem using Graph Neural Networks

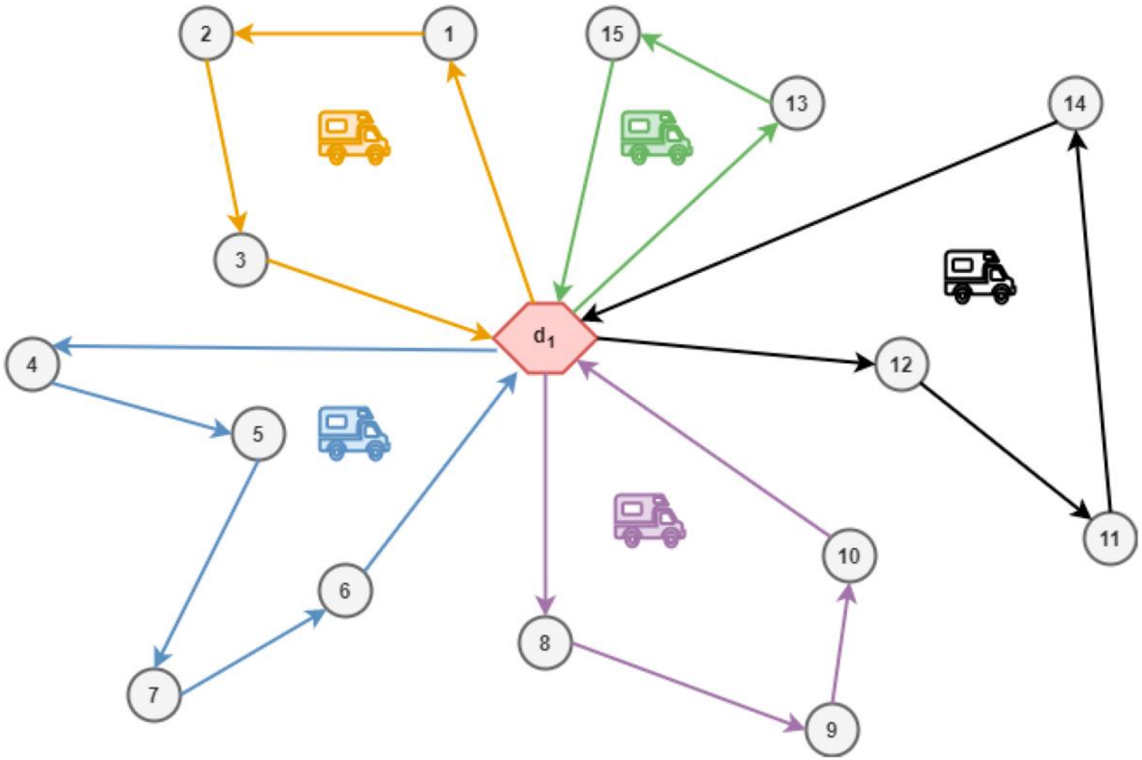
Fredrik Hagström

02.12.2021

Advisor and Supervisor: *Fabricio Oliveira*

Systeemianalyysin laboratorio

The Vehicle Routing Problem



Graph from Kovács et al. 2018

The Vehicle Routing Problem

- One of the most studied problems in combinatorial optimization
- Utilizing optimization algorithms in routing produce savings of 5% to 20% in global transportation costs (Moghdani et al. 2021)
- There are many variants of the VRP, e.g. Capacitated VRP and VRP with Time Windows

The Vehicle Routing Problem

- The VRP is an NP-hard problem, hence computation times for exact algorithms become unreasonable for large sets of customers
- Many heuristic algorithms provide good approximations within reasonable computing times (Sharma et al. 2018)
- Could the computational time and cost be further reduced with a Deep Learning model?

Thesis

- In this thesis we explore a Graph Neural Network (GNN) model for predicting solutions to the VRP.
- The developed GNN is based on the Recurrent Relational Network (RRN) architecture (Palm et al. 2018).
- The GNN outputs an edge probability matrix, on which we perform beam search decoding to yield valid tours.

Related Work

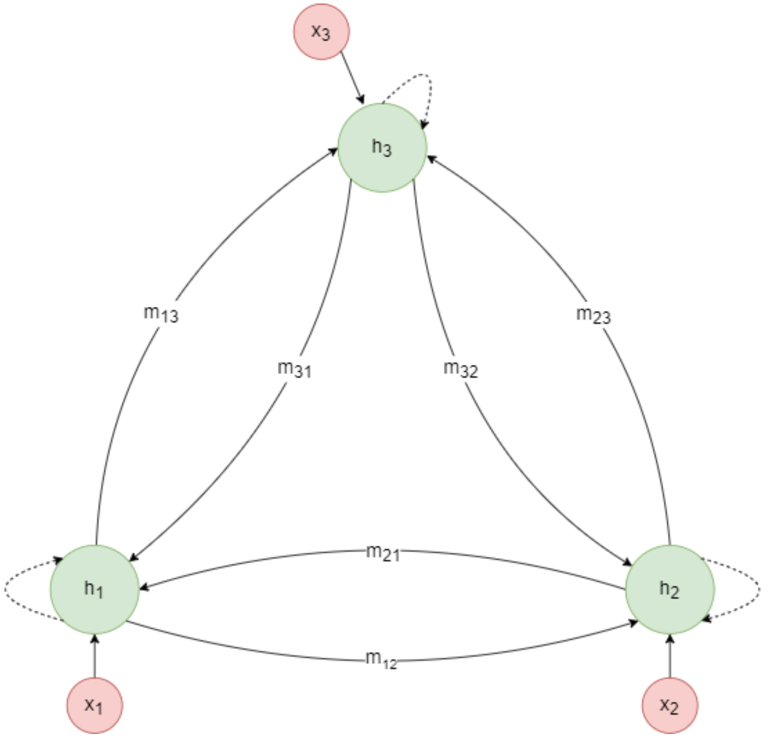
- Interest in utilizing Deep Learning for solving Combinatorial Optimization problems has grown during recent years
- Most research propose hybrid models that combine Deep Learning and traditional models. For instance, learning a heuristic that is used in a local search algorithm
- A few propose end-to-end learning models that output solutions directly from input

Data

- The training and test data are randomly generated instances of the VRP solved using OR-tools, a Google suite that provides powerful solvers for important optimization tasks.
- Three different problem sets:
  - 10 000 instances of VRP with 20 customer nodes and 5 delivery vehicles
  - 10 000 instances of Travelling Salesperson Problem (TSP) with 20 customer nodes
  - 1000 instances of VRP with 50 customer nodes and 5 delivery vehicles

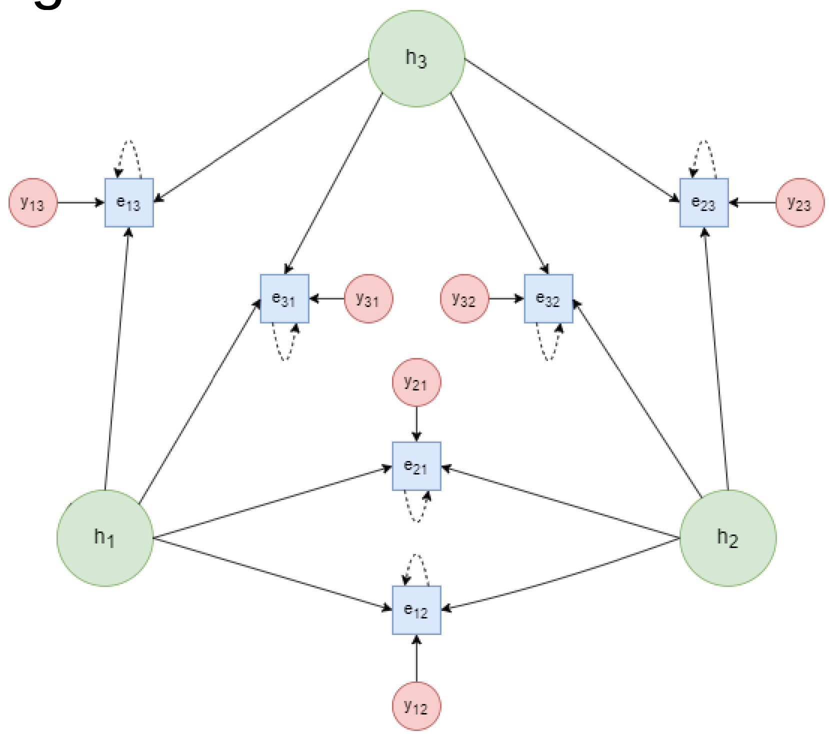
Recurrent Relational Networks

- 1. Update node states



Recurrent Relational Network

- 2. Update edge states.



- 3. Compute output of each edge.
- 4. Repeat for n iterations.

Beam search

- Beam search is a limited-width breadth-first search, where we iteratively expand the *b* most likely partial tours until every node in the graph has been visited (Joshi et al. 2019).
- The probability of a partial tour π' can be expressed using the chain rule of probability:

$$\mathrm{p}(\pi') = \prod_{j \sim i \in \pi'} p_{ij}$$

where node j comes after node i in the tour and p_{ij} is the probability of the edge between the nodes appearing in the tour.

Beam search

- Two strategies:
  - **”Vanilla” beam search**: Returns the most probable complete tour. Faster, but often doesn’t yield the shortest tour. Mainly used for fast validation during model training.
  - **Shortest beam search**: Computes the tour length of all *b* complete tours found during beam search and returns the shortest. Used for final validation of trained model.

Average optimality gap

- The performance of the model predictions compared to the test data is evaluated using average optimality gap.
- For m test instances, the average optimality gap (aog) is computed as:

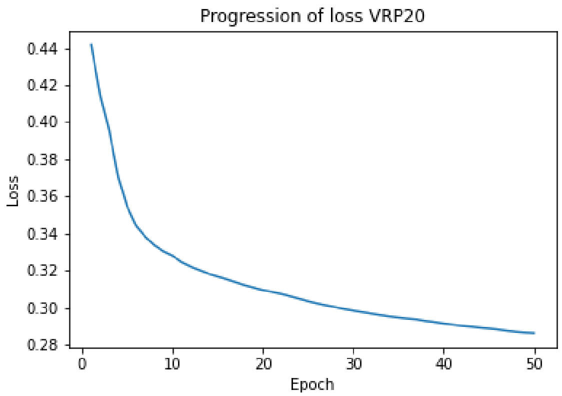
$$aog = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{l_i}{\hat{l}_i} - 1\right)$$

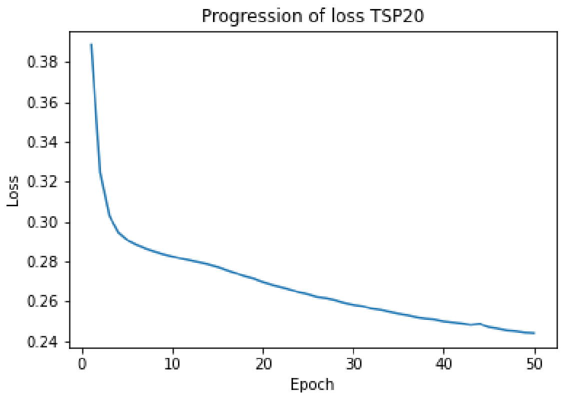
where l_i is the predicted tour length and $\hat{l}_i$ is the target tour length.

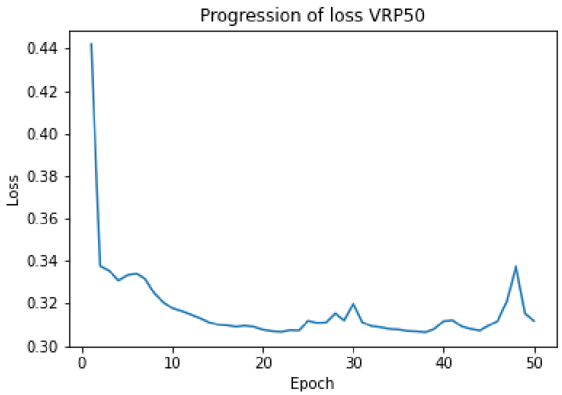
Results

- We examine the following results:
  - Progression of loss during training.
  - Progression of average optimality gap during training (computed with vanilla beam search)
  - Performance of fully trained model (computed using shortest beam search)
  - The performance of each iteration of the RRN.
  - The effect of beam size b.
  - Sensitivity analysis of different configurations of the RRN.

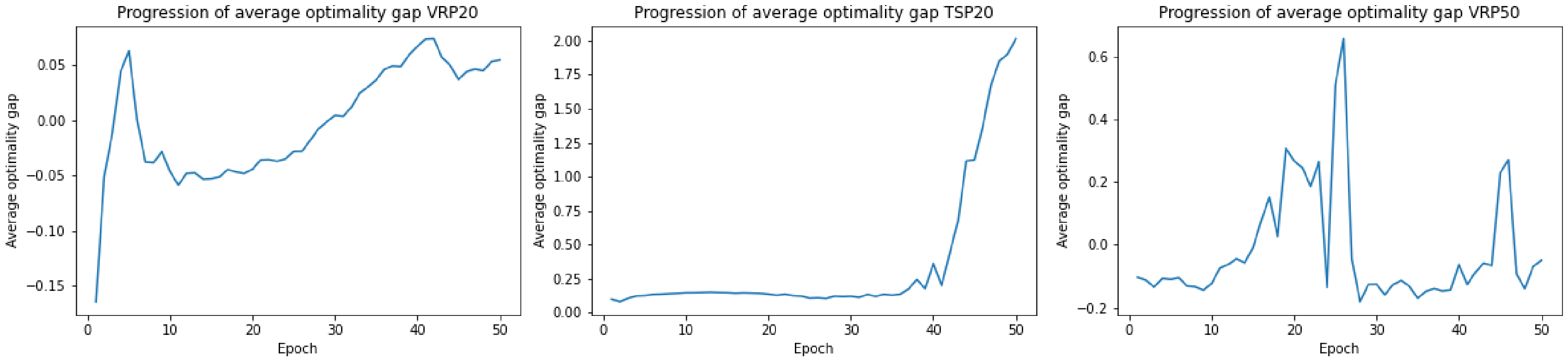
Progression of loss







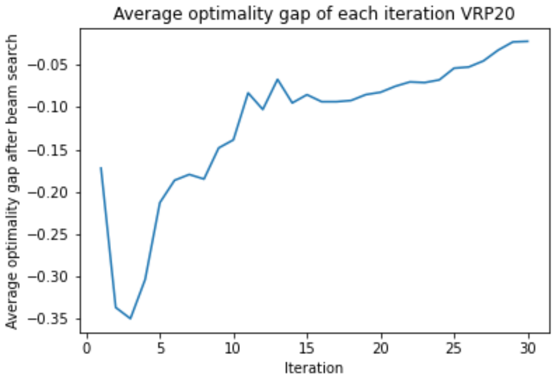
Progression of average optimality gap

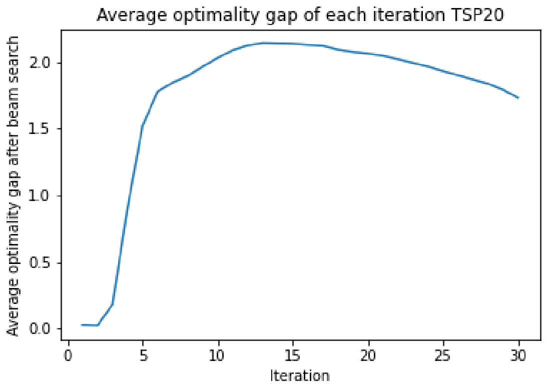


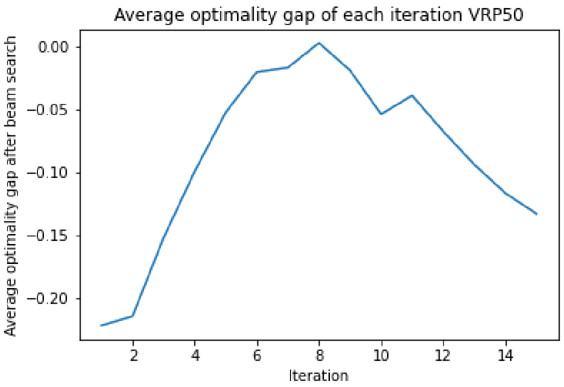
Comparing beam search performance using RRN output vs no additional info

	Average optimality gap (RRN computed edge probabilities)	Average optimality gap (Uniform edge probabilities)
VRP 20	-34,4 %	16,7 %
TSP 20	1,9 %	112,3 %
VRP 50	-23,4 %	157,3 %

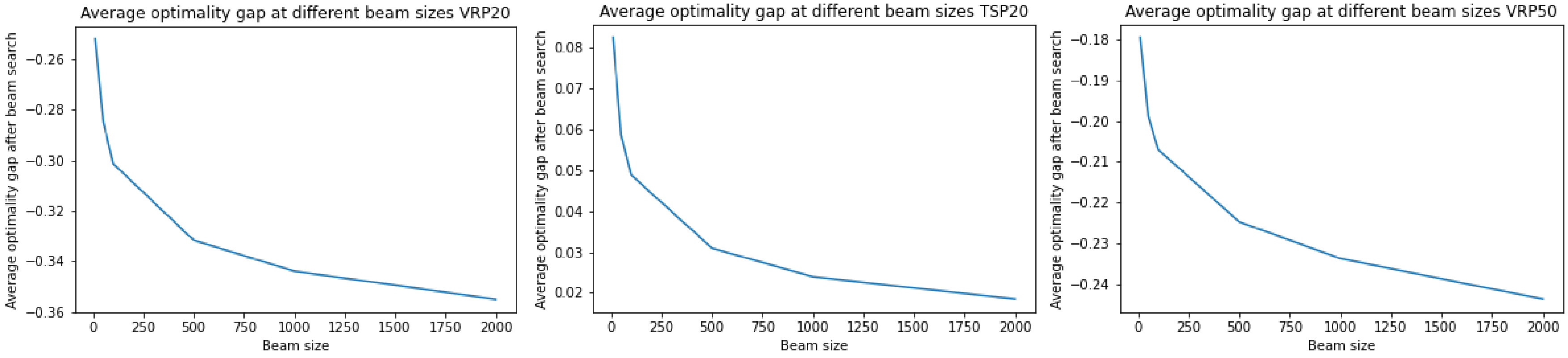
How many iterations are needed in the RRN?



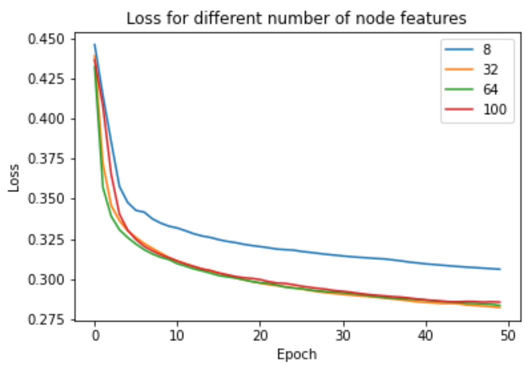


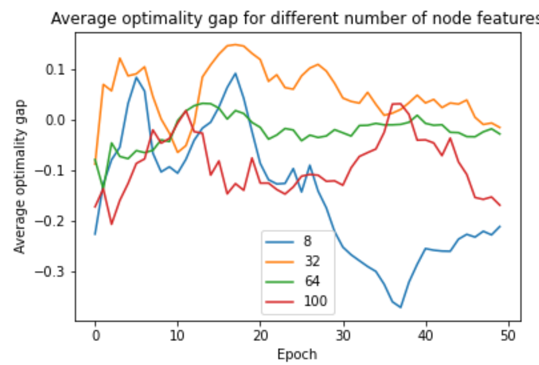


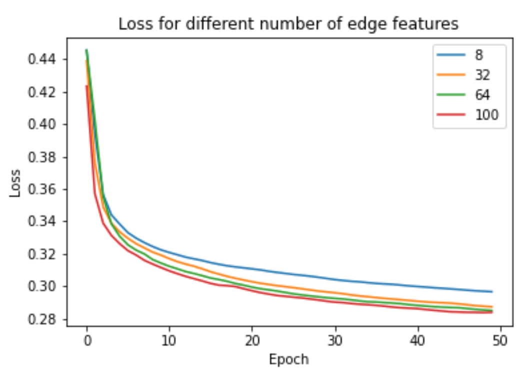
Effect of beam size

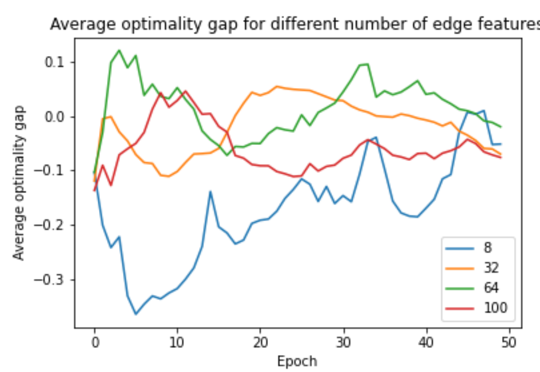


Sensitivity analysis

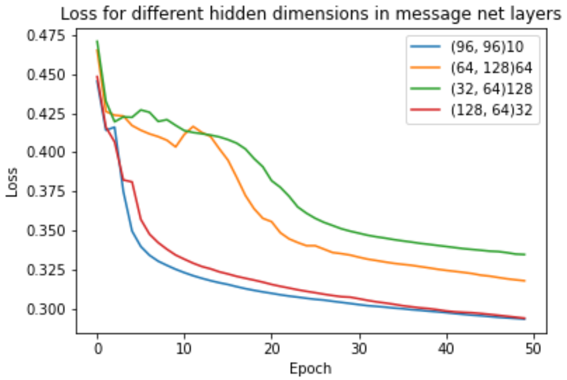


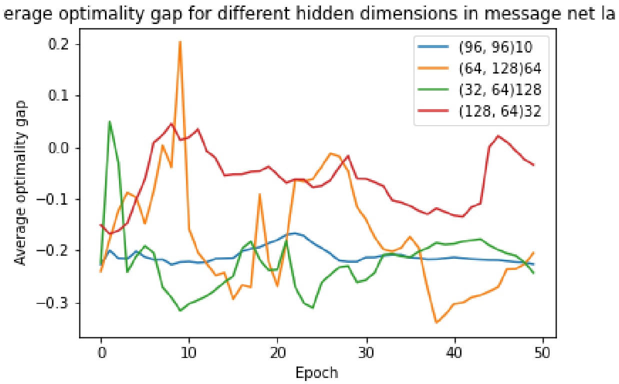


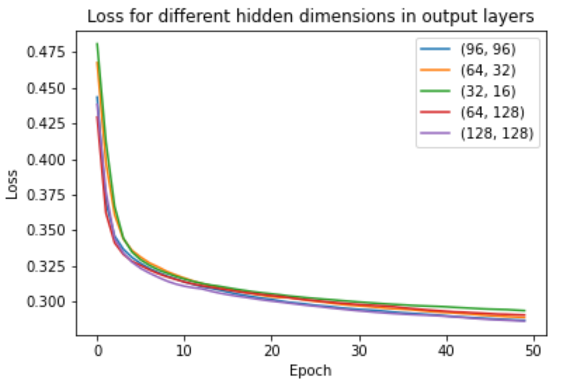


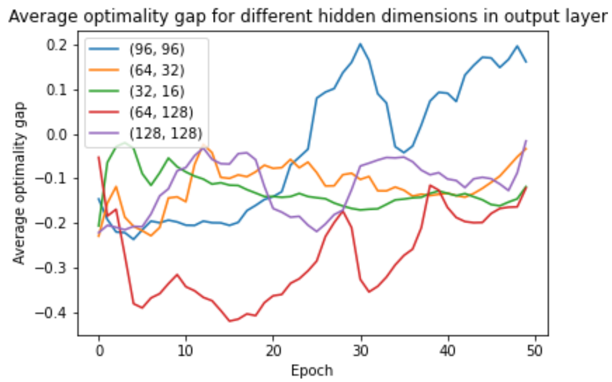


Sensitivity analysis



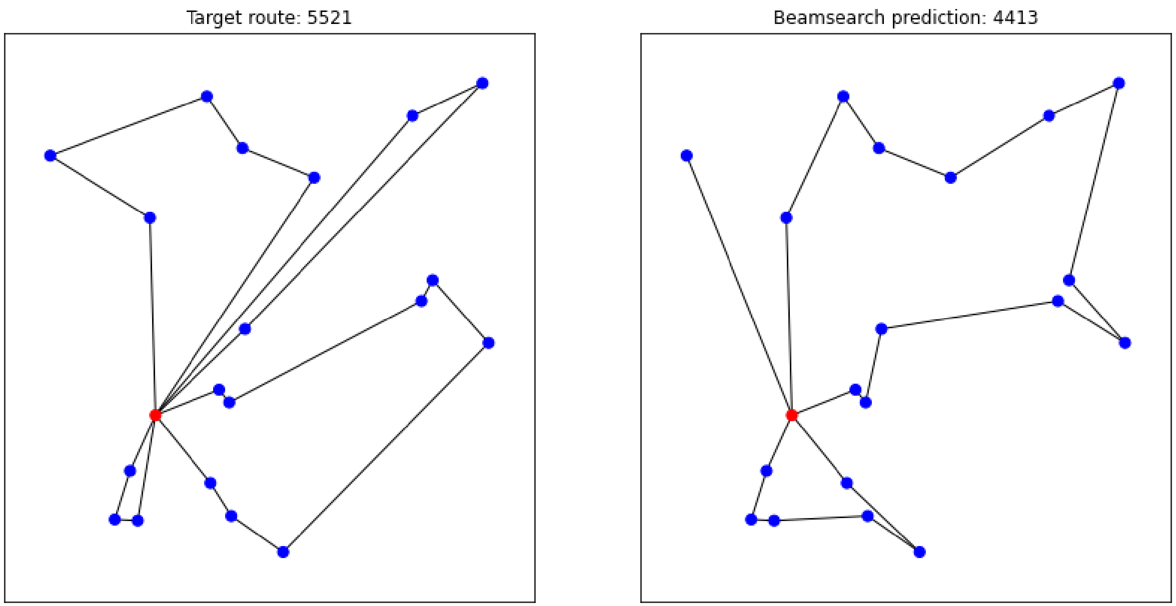






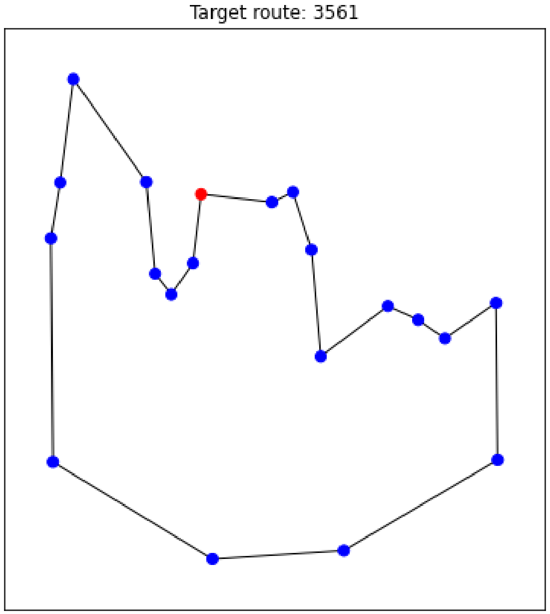
Example tours

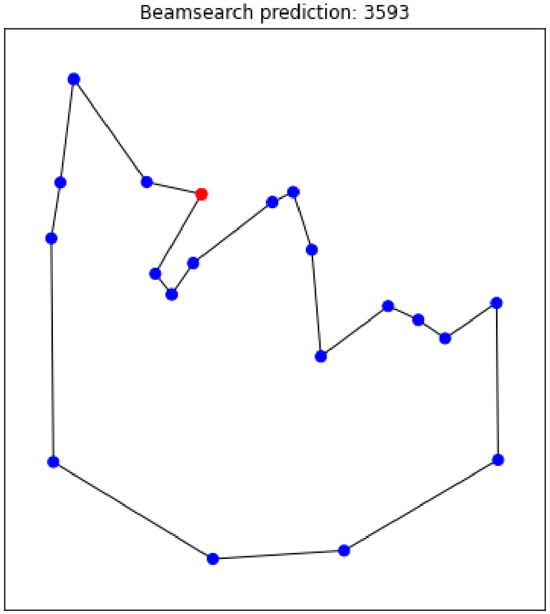
- VRP 20



Example tours

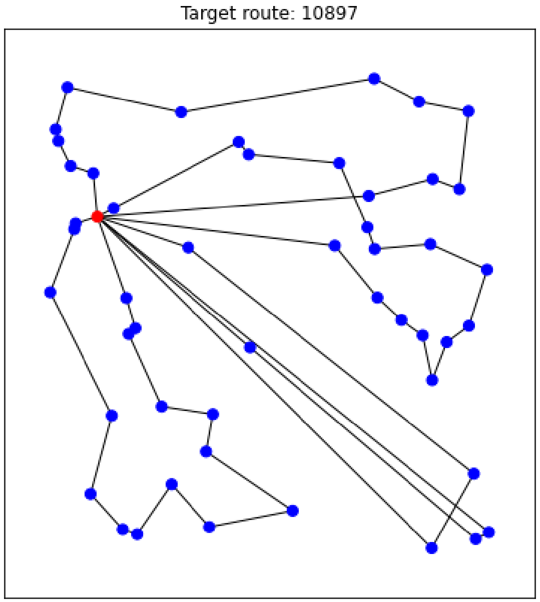
- TSP 20

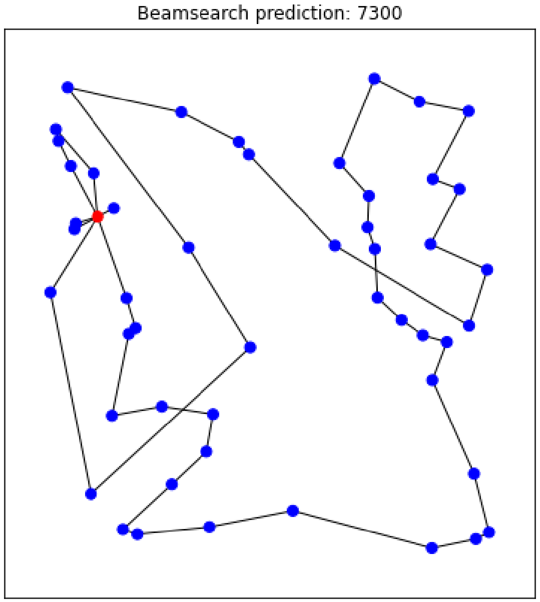




Example tours

- VRP 50





Conclusions

- The model manages to outperform the solutions generated by OR-tools for the VRP 20 and VRP 50, but not for the TSP 20. This could be because OR-tools manages to produce near-optimal solutions for the TSP, but has a harder time finding optimal solutions for the more complex VRP.
- Even though the RRN doesn’t generalize well to unseen test data, it provides a good starting seed for beam search to find good solutions.
- The best solutions are achieved with the output from the first few iterations of the RRN. Combined with the irregular progression over iterations, this suggests that the relational reasoning of the network doesn’t add significant improvements, at least for such small problem sizes. Other GNN architectures might provide better results.
- Finding better quality data could improve the generalization of the model, but is infeasible in practice. This is the greatest weakness of using a supervised learning based model.

Future research

- The natural progression would be to develop the model in the reinforcement learning framework. This would eliminate the need for finding optimal data to train on.
- The RRN could be developed by increasing the number of hidden layers in the message and output functions.
- Other GNNs could be explored for comparison.

Literature and References

- Palm et al. 2018. Recurrent Relational Networks
- Joshi et al. 2019. An Efficient Graph Convolutional Network Technique for the Travelling Salesman Problem
- Zhou et al. 2020. Graph neural networks: A review of methods and applications
- Bengio et al. 2020. Machine Learning for Combinatorial Optimization: a Methodological Tour d’Horizon
- Sharma et al. 2018. Vehicle routing problem: recent literature review of its variants
- Moghdani et al. 2021. The green vehicle routing problem: A systematic literature review
- Kovács et al. 2020. Fitness Landscape Analysis and Edge Weighting-Based Optimization of Vehicle Routing Problems