On Learning Paradigms for the Travelling Salesman Problem

Contributions
End-to-end learning for TSP
We unify recent ideas in learning for Combinatorial Problems through a generic end-to-end pipeline.

Zero-shot generalization
We design controlled experiments to study the impact of architecture and learning paradigms on solving TSP instances larger than those seen in training (when training on fixed sizes).

Chaitanya K. Joshi\textsuperscript{1}, Thomas Laurent\textsuperscript{2}, and Xavier Bresson\textsuperscript{1}
\textsuperscript{1} Nanyang Technological University, Singapore
\textsuperscript{2} Loyola Marymount University, LA, USA
Correspondence: chaitanya.joshi@ntu.edu.sg

Studying the end-to-end pipeline for TSP

Graph Definition
Define TSP as a decision problem:
- Full graphs
- $k$-NN graphs

Graph Embedding
Obtain embeddings of nodes and edges:
- GCN/GAT
- Transformer
- Simple MLP

Solution Decoding
Assign probability to each node or edge:
- Autoregressive
- Non-autoregressive

Graph Search
Select feasible solution set:
- Greedy search
- Beam search
- Sampling

Policy Learning
Train prediction policy end-to-end:
- Supervised Learning
- Reinforce Learning

Generalization, esp. to larger sizes, is difficult.
Sparse graphs lead to faster learning.
Modern GNNs do not have implicit scale-invariance, i.e., cannot generalize zero-shot.
Inductive bias: AR decoding generalizes better than NAR.
Step-by-step AR is slower than NAR.
Search/sampling is a trade-off between performance and inference time.
SL overfits to specific graph sizes.
Comparatively, RL generalizes better.

GNN-based solvers generalize poorly to TSPs larger than training size.

Prediction heatmaps
Visualizing probability to be on TSP tour for each edge before performing graph search:

Visualizing poor generalization

Related Work

What is needed for zero-shot generalization?

Acknowledgement: XB is supported in part by NRF Fellowship NRF2017-10.