Learning TSP Requires Rethinking Generalization

Chaitanya K. Joshi
Institute for Infocomm Research, A*STAR, Singapore

Joint work with Q. Cappart, L. M. Rousseau (Polytechnique Montreal), T. Laurent (LMU), X. Bresson (NTU)
Travelling Salesperson Problem (TSP)

• Most extensively studied NP-hard problem with wide practical applications.

• Engine of discovery for advances in applied mathematics...and deep learning?
Summary

- Experimental review of SotA deep learning-based combinatorial optimization solvers, with TSP as a benchmark.

- We can learn to solve trivially small instances close to optimality, but extrapolating to larger and realistic problem instances is a challenging and open problem.
Deep Neural Networks can **perfectly memorize** very complex and random training data.

Inspired the community to study **generalization** and **extrapolation** performance **beyond training** data.

**Inspired by**

**UNDERSTANDING DEEP LEARNING REQUIRES RE-THINKING GENERALIZATION**

Chiyuan Zhang*  
Massachusetts Institute of Technology  
chiyuan@mit.edu

Samy Bengio  
Google Brain  
bengio@google.com

Moritz Hardt  
Google Brain  
mrtz@google.com

Benjamin Recht†  
University of California, Berkeley  
brecht@berkeley.edu

Oriol Vinyals  
Google DeepMind  
vinyals@google.com
Non-learnt Heuristics for TSP

- Cheap approximate solutions.
- No/few theoretical guarantees.
- Handcrafted.

Motivation

Furthest Insertion Heuristic (GIF)
Learnt Heuristics for TSP Novel NP-hard Problems

- Cheap approximate solutions.
- No/few theoretical guarantees.
- Handcrafted.
Motivation

**Learnt Heuristics for TSP Novel NP-hard Problems**

Recent works showing this is possible for trivially small TSP instances...
Learnt Heuristics for TSP Novel NP-hard Problems

Recent works showing this is possible for trivially small TSP instances...

![Bar chart showing average tour length for different heuristics and algorithms.](chart.png)
Research Question

How do we scale to practical sizes beyond few hundreds of nodes?
We were unable to outperform the simple insertion heuristic when directly training on 10+ Million TSP200 samples for 500 hours on university-scale hardware...
Option 2: Transfer Learning from small instances

Alternative: learn **efficiently** from **trivially small TSPs** + transfer the learnt policy to larger graphs in a **zero-shot fashion** or via fast finetuning.
Which Architectures, Learning Paradigms and Inductive Biases enable strong Zero-shot Generalization to large TSP instances?
Step 1: A unified view of recent advances

End-to-end Neural Combinatorial Optimization Pipeline
Measuring generalization across TSP sizes

Step 2: A fair and controlled experimental setup

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
<th>Insertion Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSP Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(1) Compare models fairly: fixed #params, epochs, computation.

(2) Generalization performance: test TSPs beyond training range.

(3) Insertion heuristic baseline: quantify ‘good’ generalization.
Our Findings
• Sparse k-NN graphs > Fully connected graphs.
• Maintain consistent graph diameter across TSP sizes.
• GNN-Sum, Transformers (GAT): most expressive.
• GNN-Max/Mean: agnostic to node degree => better generalization.
• **Autoregressive decoding (AR):** sequential inductive bias => strong generalization!
• NAR: **fast**, but poor extrapolation.
Reinforcement Learning (RL): better under greedy decoding.
Supervised Learning (SL): more amenable to beam search decoding.
What’s next?

Expressive + **scale invariant GNN architectures** for COPs.

More powerful **classical search techniques** s.a. MC Tree Search, post-hoc Local Search.

Novel **transfer learning** and **meta-learning** techniques for extrapolation.
Pre-print and Code are online

- **GitHub:** [github.com/chaitjo/learning-tsp](http://github.com/chaitjo/learning-tsp)
- **Blog:** [chaitjo.com/neural-combinatorial-optimization/](http://chaitjo.com/neural-combinatorial-optimization/)
Thank you!