Appendix for Integration of Global and Local Features for Large-scale TSP via Reinforcement Graph Fusion

Primary Keywords: *None*

Per-Instance Search

Utilizing a heatmap that highlights promising regions within the search space, discrete solutions can be obtained through various methods such as beam search (Joshi, Laurent, and Bresson 2019), sampling (Kool, van Hoof, and Welling 2019), guided tree-search (Li, Chen, and Koltun 2018), dynamic programming (Kool et al. 2022), and Monte Carlo Tree Search (MCTS) (Fu, Qiu, and Zha 2021). In this study, we primarily employ sampling and MCTS as the per-instance search strategies.

Experimental Details

Hardware

We adhere to the hardware configuration proposed by (Fu, Qiu, and Zha 2021). The three conventional algorithms operate on an Intel Xeon Gold 5118 CPU @ 2.30GHz due to their source codes' lack of GPU support. To ensure equitable assessment, the learning-based approaches execute on a GTX 1080 Ti GPU during testing. MCTS runs on an Intel Xeon Gold 6230 80-core CPU @ 2.10GHz, utilizing 64 threads for TSP-500 and TSP-1000, and 16 threads for TSP-10000.

Hyperparameters

The number of sub-graphs is set as $\lceil (|V|/|V_{sub}|) \times 5 \rceil$ in the process of sampling sub-graphs, where $|V_{sub}|$ is the number of points for sub-graphs. For Att-GCN, the $|V_{sub}|$ was set to 50 on all datasets in order to be consistent with the original method. And for POMO, we choose the pre-trained model for TSP-100 provided by the original method, setting $|V_{sub}|$ to 100. For the warm-up module, the iterations corresponding to the three datasets is set to 50, 25 and 5, the main consideration here is the time consumption problem of warm-up, which will be given in the ablation experiments for detailed description. And the learning rate β for back-propagation is set to 20.

Reproduction

We implement the DTTGF for the Traveling Salesman Problem (TSP) using PyTorch(Paszke et al. 2019). Our TSP code is available in the supplementary material.. The test instances used in our experiments are provided by (Fu, Qiu, and Zha 2021).

References

Fu, Z.; Qiu, K.; and Zha, H. 2021. Generalize a Small Pre-trained Model to Arbitrarily Large TSP Instances. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 7474–7482. AAAI Press.

Joshi, C. K.; Laurent, T.; and Bresson, X. 2019. An Efficient Graph Convolutional Network Technique for the Travelling Salesman Problem. *CoRR*, abs/1906.01227.

Kool, W.; van Hoof, H.; Gromicho, J. A. S.; and Welling, M. 2022. Deep Policy Dynamic Programming for Vehicle Routing Problems. In Schaus, P., ed., *Integration of Constraint Programming, Artificial Intelligence, and Operations Research - 19th International Conference, CPAIOR 2022, Los Angeles, CA, USA, June 20-23, 2022, Proceedings*, volume 13292 of *Lecture Notes in Computer Science*, 190–213. Springer.

Kool, W.; van Hoof, H.; and Welling, M. 2019. Attention, Learn to Solve Routing Problems! In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Li, Z.; Chen, Q.; and Koltun, V. 2018. Combinatorial Optimization with Graph Convolutional Networks and Guided Tree Search. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 537–546.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Neural Information Processing Systems,Neural Information Processing Systems*.