ICEB 2021 Proceedings (Nanjing, China)
International Conference on Electronic Business (ICEB)

Winter 12-3-2021

Navigational Guidance – A Deep Learning Approach

Benjamin Yen

Yu Luo

Navigational Guidance – A Deep Learning Approach

Benjamin Yen [1,*]
Yu Luo [2]

*Corresponding author
[1] Associate Professor, The University of Hong Kong, SAR China, benyen@hku.hk
[2] Research Assistant, The University of Hong Kong, SAR China, yuluo@hku.hk

ABSTRACT

The useful navigation guidance is favorable to considerably reducing navigation time. The navigation problems involved with multiple destinations are formulated as the Directed Steiner Tree (DST) problems over directed graphs. In this paper, we propose a deep learning (to be exact, graph neural networks) based approach to tackle the DST problem in a supervised manner. Experiments are conducted to evaluate the proposed approach, and the results suggest that our approach can effectively solve the DST problems. In particular, the accuracy of the network model can reach 95.04% or even higher.

Keywords: Navigation guidance, directed Steiner Tree, graph neural network.

INTRODUCTION

The processes of navigation are omnipresent in modern society. Since most people are reluctant to spend enormous time on the way to the destinations, providing practical navigation guidance is desirable. Good guidance can save substantial time. Therefore it could benefit people by certain occupations, such as delivery men or Uber drivers to whom time cost is crucial. Speedy finishing orders bring more orders and income consequently. On the other hand, excessive steering would incur delay or even retreat in the process of navigation. Furthermore, Uber drivers might not prefer to take an order when seeing the route is tortuous. In terms of the Web, an appropriate path is beneficial for both surfers and website owners by reducing the searching time to retrieve information and promoting the advertisement of websites, respectively.

Providing effective navigation guidance is a challenging task due to the scale and complexity of applications in reality (e.g., city maps and websites) (Yen & Wan, 2010). Worse still, such a task can be further complicated by the diversity and multiplicity of destinations. It is straightforward to use those shortest path algorithms (e.g., Dijkstra algorithm (Dijkstra, 1959)) to acquire the optimal paths for single destination problems, yet the problems become tricky with multiple destinations involved. Basically, randomly exploring multiple destinations could be cumbersome. The major issue then is how to find optimal paths for reaching multiple destinations with minimum navigation time. Therefore, this paper focuses on an approach to analyze the problem as the Directed Steiner Tree (DST) problem (Charikar *et al.*, 1999), an NP-hard combinatorial optimization (CO) problem (Karp, 1972). Recently, deep learning has been proved to be suited for solving various CO problems (Bengio *et al.*, 2021; Mazyavkina *et al.*, 2021; Vesselinova *et al.*, 2020). In general, NP-hard problems on graphs can be formulated as sequential decision-making tasks due to their highly structured property (Joshi *et al.*, 2019), and deep learning can automatically detect useful patterns in the data. Hence, deep learning-based approaches are good candidates to be adopted for approximately solving those CO problems.

In this paper, we propose a deep learning-based approach to solve DST problems. Then the solutions can be served as optimal navigation guidance. The core of our model is Graph Attention Network (GAT) that proposed by Veličković *et al*. (2017). We demonstrate our approach through simulations, and the experiments indicate that the approach can effectively address DST problems and efficiently generate optimal navigation guidance.

The remainder of this paper is organized as follows. A survey of related work is given in section 2. A problem modeling is introduced in section 3. Subsequently, the deep learning (neural graph networks, to be exact) based approach is described in section 4. The numerical experiments are conducted in section 5 to evaluate the performance of the proposed approach. Finally, the paper concludes with a summary in section 6.

LITERATURE REVIEW

Navigation and Guidance

In a broad sense, navigation can refer to any study that involves the determination of position and direction (Hofmann-Wellenhof *et al*., 2003). Most literature on navigation studied position systems of vehicles with multi-sensors (Syed *et al*., 2008; Jost *et al*., 2014; Wang *et al*., 2020; Rose *et al*., 2014; Jiang *et al*., 2017; Vu *et al*., 2012), or navigation in networking (Win *et al*., 2011), whereas the direction aspect is the thing that draws our attention.

Some studies focus on navigation for people. Niu *et al*. (2019) fuse the information of two foot-mounted systems to achieve more robust and reliable performance; Renaudin *et al*. (2017) conduct an original transdisciplinary study of the underlying reasons about no universal solution of a navigation system; Mancini *et al*. (2015) propose a safe point-to-point navigation for impaired people owing to high-detailed road graphs; Patil *et al*. (2011) present a novel approach to guide agents towards desired goals using navigation fields. Besides humankind, navigation for mobile robots is one of the popular fields. Shi *et al*. (2007)

developed a novel fuzzy behavioral scheme for navigating an unmanned helicopter in cluttered 3-D spaces; Gabriely and Rimon (2008) introduce a notion of competitiveness suitable for online mobile robot navigation in general planar environments; Liu *et al*. (2021) present a self-improving lifelong learning framework for a mobile robot navigating in different environments that improves a mobile robot's navigation behavior purely based on its own experience.

It is significant to recommend an appropriate path, i.e., guidance, for navigation. This is because not only distinct paths might enormously differ in time, but also a poor path would cause troubles (e.g., collision). Votion and Cao (2019) develop navigation algorithms for cooperative unmanned vehicles in cost-map environments. Loizou (2017) maps an obstacle-cluttered environment to a trivial domain, reducing the navigation task to connecting the images of the initial and destination configurations by a straight line. Devaurs *et al*. (2016) develop sampling-based algorithms for path planning in cost space, efficiently solving high-dimensional search space. Muralidharan and Mostofi (2019) focus on a general path planning problem of a robot on a graph with edge costs, where each node has a Boolean value denoting success or failure with a given probability.

Combinatorial Optimization (CO) and Graph Neural Network (GNN)

This study aims at investigating the optimal navigation guidance in general graph structure (Gross & Yellen, 2006) environments, e.g., city maps and websites. Yen and Wan (2010) report positive results that support the use of the web navigation guidance for one destination web page. However, the information of interest was often scattered on several web pages. It is, therefore, desirable that the optimal navigation guidance could handle the destinations with diversity and multiplicity. Thus, the navigation problem can be formulated as a Directed Steiner Tree (DST) problem (Karp, 1972). Deep learning algorithms have been used to approximate for solving CO problems (Smith, 1999; Bengio *et al*., 2021). Besides, owing to the graph structure, Graph Neural Networks (GNN), deep learning-based methods for graph data, are more suitable for solving them.

Khalil *et al*. (2017) first consider the graph structure and encode problem instances using a graph embedding model (Dai *et al*., 2016). Nowak *et al*. (2017) train a Graph Neural Network (GNN) in a non-autoregressive and supervised manner to output a TSP tour (Wu *et al*., 2016). Joshi *et al.* (2019) leverage different Graph Convolution Networks (Bresson & Laurent, 2017) and improve the performance on top of Nowak et al.'s work. Kool *et al*. (2021) further integrate Joshi et al.'s model with dynamic programming (DP) to Solve Vehicle Routing Problems (VRP). Cappart *et al*. (2021a) train a GAT-based model to learn a policy for Markov decision process (MDP) problems using reinforcement learning and constraint programming.

Along with those work mainly on TSP and VRP, Barrett *et al*. (2020) utilize the framework of Dai et al. on the Maximum Clique (MC) problem. Li *et al*. (2018) integrate a GCN with a guided tree-search in a supervised setting for Maximal Independent Set (MIS), Minimum Vertex Cover (MVC), and MC problems. From the perspective of approximation ratio, Sato *et al*. (2019) first theoretically demonstrate approximation ratios of using GNN for MVC. For more surveys of machine learning for combinatorial optimization in general, please refer to (Bengio *et al*., 2021; Mazyavkina *et al*., 2021; Vesselinova *et al*., 2020; Cappart *et al*., 2021b).

To our knowledge, this work is the first to train Graph Neural Network (GNN) models with the goal of solving DST problems. The closest work to ours might be Joshi *et al.* (2019) and Li *et al*. (2018), both using GNN models to solve different CO problems by supervised learning, but their models are not suited for solving our problems.

PROBLEM DESCRIPTION AND MODELLING

The structure of a map is represented as a directed graph, where each node denotes a site (e.g., a building, a recreation facility, or a residential spot), and each arc represents a road (i.e., path, arc, or edge).

Graph Structure and Traversal

The graph $G=(V,E,w)$ is defined as follows:

Nodes: $V=\{v_1,v_2,...v_n\}$, where v_i denotes a site or a web page, i=1,2,…,n;

Arcs (*Arcs*): $E=\{e_{ij}=[v_i,v_j]$ *or* $(i,j)\,|$ A road or a hyperlink exists from node i to node $j\}$;

Weights: $w: E \to R^+$ is the edge-associated weight that reflects the road length or loading time of a web page.

In the above definition of *G*, the length of e_{ij} is the same as e_{ji} only if it is a bidirectional road, as far as the web is concerned, the loading time will vary as it is related to the size of the next desired page. Traffic and network condition are not considered in this paper.

A simple example with identical weights in Figure 1 is presented to illustrate the traversal process. The preferences of navigation can be diversified that there could be multiple destination nodes, especially equal or greater than 3. As shown in Figure 1, it is non-trivial to determine an optimal path from the root node *A* to the destination nodes *G*, *J*, and *K*.

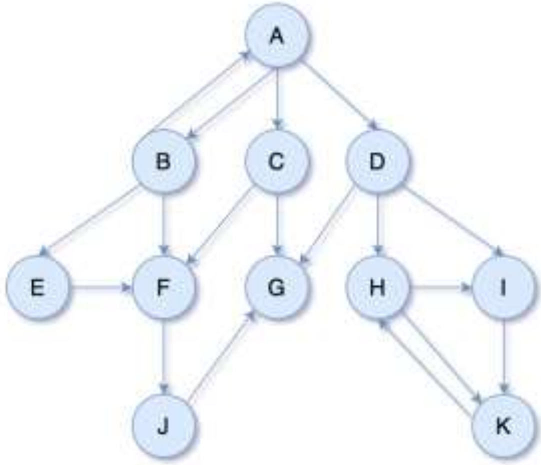

Figure 1: Single source to multiple destinations.

Steiner Problem in Directed Graph

The Directed Steiner Tree (DST) problem is defined as follows: Given a directed graph $G=(V,E,w)$ with *n* nodes and *m* arcs, a node *r* called root and a set *X* of *k* nodes called terminals (destinations) and non-negative weights *w* associated with arcs, the DST problem is to find a minimum cost directed tree rooted at *r* spanning all destinations.

In fact, the DST problem has been widely studied in the literature, whereas no polynomial approximation algorithm achieves a constant ratio contrary to the undirected version (Watel & Weisser, 2016). One of the renowned heuristic methods for DST problem is proposed by Charikar *et al*. (1999), a greedy non-trivial algorithm achieving $l(l-1)k^{\frac{1}{l}}$ factor based on Zelikovsky's l-restricted Steiner tree (Zelikovsky, 1997), where *n* is the number of nodes in the graph, *k* is the number of destinations, and *l* denotes every leaf node being at most *l* edges away from the root node. Due to the high time complexity of Charikar *et al.*'s algorithm, Hsieh *et al.* (2006) developed a faster approximation algorithm for directed Steiner trees based on it. Takahashi and Matsuyama (1980) proposed a straightforward greedy approach to find the shortest destination from the "selected tree." Since Charikar et al.'s algorithm, the state-of-the-art polynomial-time approximation algorithm for the DST problem is presented by Grandoni *et al*. (2019). Linear programming-based methods are also widely applied to DST problems, such as (Rothvoß, 2011). For the exact solution of the DST problem, the directed version of the original Dreyfus-Wagner algorithm (Dreyfus & Wagner, 1972), a dynamic programming algorithm, can be used to obtain it in exponential time.

DEEP LEARNING APPROACH

Both the approximation and exact algorithms have intrinsic limitations. Takahashi-Matsuyama algorithm (Takahashi & Matsuyama, 1980) is fast, but the performance (solution quality) is not satisfactory sometimes; the performance of the Charikar algorithm (Charikar *et al.,* 1999) is acceptable but at the sacrifice of time and space. Even though the approximation ratio of the Grandoni algorithm (Grandoni *et al.,* 2019) is the best to date, its computation time of it remains an issue.

In light of two essential requirements for real-time and minimum cost, we propose a deep learning approach based on Graph Attention Network to provide navigation guidance for multi-destination problems. The guidance data (i.e., optimal paths) is generated from the directed variant of the Dreyfus-Wagner algorithm (Dreyfus & Wagner, 1972). Then path information as well as graph information, including node features and edges, are fed into the neural network model to train. The output of each node denotes the probabilities of nodes that belong to the optimal path. After adequate training, the neural network can provide identical navigation guidance paths with those generated by the Dreyfus-Wagner algorithm given unseen destination sets, which reflects good generalization.

In terms of neural network architecture, since the graph is a type of non-Euclidean structure data, unlike image, video, text, etc. those can be represented in the Euclidean space, it is challenging and unsuitable to adopt Multilayer Perceptron (MLP) or convolution neural networks (CNNs) (LeCun & Bengio, 1995). Recently, many studies have extended deep learning on graph data called graph neural networks (GNNs) and proposed various network architectures exclusively for graphs (Wu *et al*., 2020). Among them, we do not take spectral-based convolutional graph neural networks (ConvGNNs) into consideration, owing to their undirected and fixed graph scope (Wu *et al*., 2020). Graph Attention Network (GAT) (Velickovic *et al.*, 2017), one of the spatial-based ConvGNNs, is suitable for our problem. GAT assumes contributions of neighbor nodes to the central node are neither identical like GraphSage (Hamilton *et al.*, 2017) nor pre-determined like GCN (Kipf & Welling, 2016). In fact, it accords with the logic of the aforementioned directed Dreyfus-Wagner algorithm. Each neighbor node has a disparate impact on the central node as the weight of edges is different when finding the shortest path. Furthermore, to emphasize the influence of weights, a GAT layer is accompanied by a linear layer that is fed in node features only. Then their outputs are passed through a Batch Normalization layer and followed by the ReLU activation function. We construct several layers to repeat such a process, guarantee enough parameters and make it fully trained. Since our problem is a binary node classification problem that nodes within the guidance are labeled as 1 while others 0, the final output layer should apply a sigmoid function to bound it in the interval [0, 1]. The whole neural network architecture is presented as follows (Figure 2).

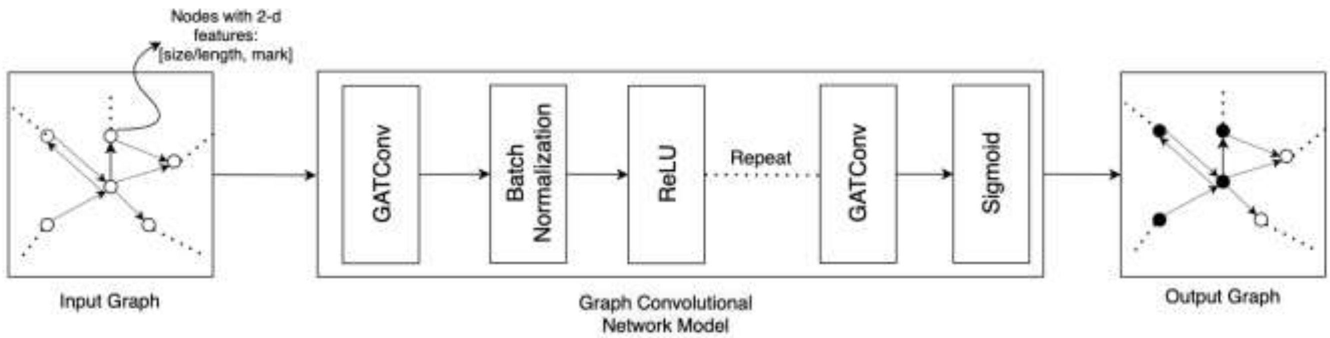

Figure 2: The model architecture.

To sum up, we feed the paths generated from an exact solution of the Directed Steiner Tree problem as well as corresponding graph data into GAT networks to pre-train, thus achieving real-time and minimum cost navigation guidance to suggest. Please refer to Section 5.2 for more details about the proposed model.

EXPERIMENT RESULTS AND DISCUSSION

In this section, numerical experiments are conducted to assess the performance of the proposed deep learning approach for navigation guidance in various scenarios. For node classification problems, the common metric is accuracy. However, the class distribution of the dataset for navigation guidance problems is uneven, i.e., the number of nodes on the optimal paths is relatively small. Besides, if one node is misclassified in the graph, either making the path impassable or superfluous, the path result is not acceptable. Hence, a more rigorous metric is required as misclassification of even one node would influence the guidance path. Therefore, instead of from the perspective of a single node, we define that for each data sample that it is classified as correct only if all nodes in the graph are labeled correctly. The metric referred to as “path accuracy” is the rate of the number of correct data samples against the total number of data sets.

Graph Generation

The graph, represented by $G=(V,E,w)$ as described in section 3.1, is generated from the following procedure:

(1) Set the number of nodes $n=|V|$.
(2) Randomly generate outdegree and point-to-neighbors for each node. The distribution of outdegree can follow power law to satisfy characteristics of certain graphs, e.g., the web (Broder *et al.*, 2000; Pennock *et al.*, 2002), BA graphs (Albert & Barabási, 2002), or random graph theory (Erdős & Rényi, 2011).
(3) Randomly assign the weights of edges from a Gaussian distribution.
(4) Randomly select *k* destinations that can be reached from the root node *r*.

After the graph generation is complete, the directed Dreyfus-Wagner algorithm (Dreyfus & Wagner, 1972) can be used to generate optimal paths according to pairs of root nodes and destinations.

Neural Network Configuration

Based on the network architecture described in section 4, there are a total of 12 layers of GAT and Linear including input layer and output layer, followed by Batch Normalization (BN) and ReLU sequentially except the output layer. Thus, there are 11 BN layers and 11 ReLU functions in total, as well as 1 sigmoid function at the final output layer. For each GAT layer, the hidden channel and head are 128 and 4, adding up to 512 parameters. In the training stage, we set batch size as 32, chose Adagrad as the optimizer, and the learning rate as 0.01. Adagrad is suited for uneven distribution problems, and these hyperparameters are obtained through trials and experience. The loss function is binary cross-entropy. All experiments are carried out with a single GPU NVIDIA Tesla P100-PCIE 16G and programmed in Python.

Experiments

We first study on a fixed graph, i.e., the graph scale, node features, edge connection, and edge weights are pinned down, and explore the influence of destination number on metric path accuracy. The initial node number and the destination number are 100 and 3, respectively.

To start with, we set destination number *k* as a constant number 3 as initial. Randomly select three destinations in the fixed graph except for the root node, and generate corresponding optimal paths. Totally we generate 40000 such data samples and split them as training dataset (70%) and test dataset (30%). The training dataset is approximately 18% of all permutations of 3 destinations, i.e. $C_3^{99} = 156849$. It is reasonable that more training data would make the neural network learn better, but we found 18% of all possible permutations is enough to train a workable model. The loss and path accuracy curves are as follows. See the figures below.

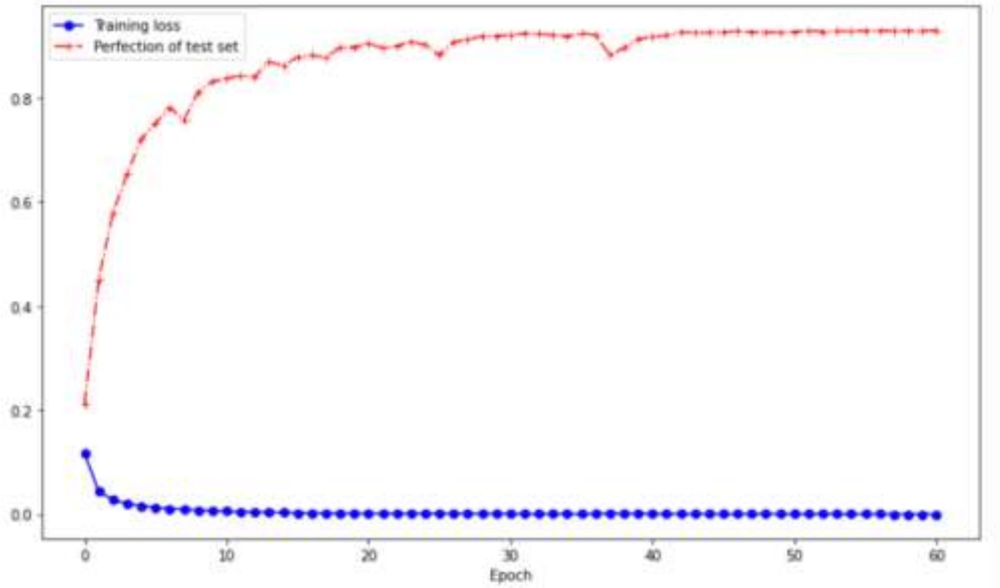

Figure 3: Path accuracy of the test set and training loss.

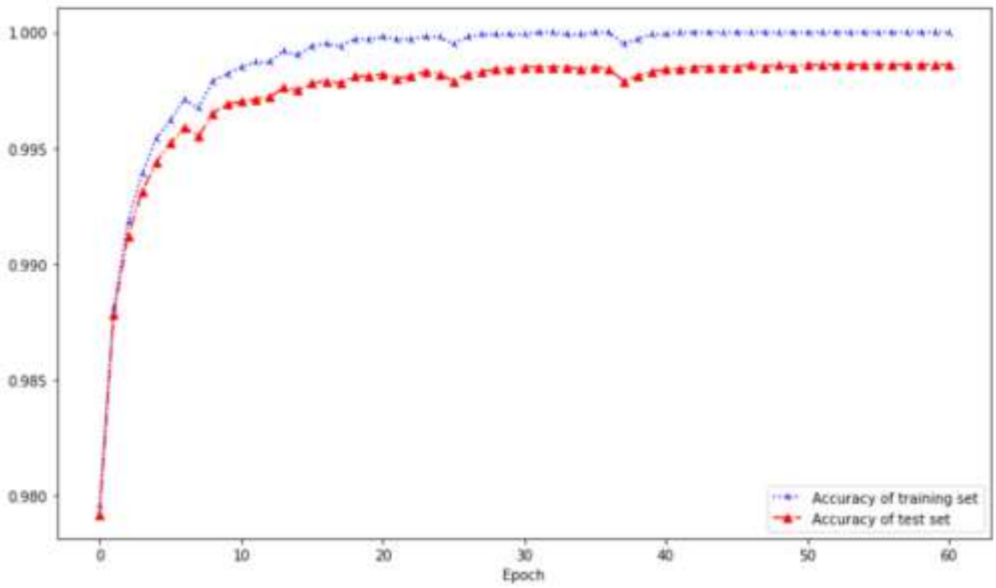

Figure 4: Path accuracy of training and test set.

As shown in Figures 3 and 4, the training loss drops to almost 0.0000, and the accuracy of the training set reaches 99.99% respectively after enough epochs. It implicates that the neural network has made full use of the training dataset. For the generalization, the path accuracy, the metric we pay attention to, of the test dataset is around 95.04% finally.

Then, we try to add nodes to a graph to imitate the growing process of the graph, e.g., new web pages are created in a website or road segments are expanding. As revealed by our experiments, the adding step of from 10% to 50% of the original number of nodes does not affect re-training and even boosts the former final path accuracy. In the case of 20%, the eventual path accuracy increases to 96.7% when the total node of a graph increases to 200. We can expand it to the scale of 1000 smoothly, and the final path accuracy is 98.10%.

Table 1: Experiment results when the graphs are growing.

Situation	Scale *n*	Destination *k*	Path Accuracy
The Basic Graph	100	3	95.04%
Addition	200	3	96.7%
Addition	1000	3	98.10%

CONCLUSION

In this paper, we introduce a deep learning-based approach for approximately solving the Directed Steiner Tree problems using Graph Attention Networks via a supervised and no-autoregressive manner. Our approach trains a network model to predict the probability of each node whether it belongs to the optimal solution. A rigorous metric beyond node classification accuracy, path accuracy, is adopted to evaluate the performances. The experiment results show that our approach can effectively solve the DST problems with good scalability, and it implies the potential to utilize Graph Neural Networks to solve Combinatorial Optimization problems over directed graphs.

REFERENCES

Albert, R. & Barabási, A.L. (2002). Statistical mechanics of complex networks. *Reviews of modern physics*, *74*(1), 47-97. doi: https://doi.org/10.1103/RevModPhys.74.47

Barrett, T., Clements, W., Foerster, J., & Lvovsky, A. (2020). Exploratory combinatorial optimization with reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(4), 3243-3250. doi: https://doi.org/10.1609/aaai.v34i04.5723

Bengio, Y., Lodi, A., & Prouvost, A. (2021). Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, *290*(2), 405-421. doi: https://doi.org/10.1016/j.ejor.2020.07.063

Bresson, X. & Laurent, T. (2017). Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*.

Broder, A., Kumar, R.i, Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., & Wiener J. (2000). Graph structure in the Web. *Computer Networks*, *33*, 309–320. doi: 10.1016/S1389-1286(00)00083-9

Cappart, Q., Chételat, D., Khalil, E., Lodi, A., Morris, C., & Veličković, P. (2021a). Combinatorial optimization and reasoning with graph neural networks. *arXiv preprint, arXiv:2102.09544*.

Cappart, Q., Moisan, T., Rousseau, L. M., Prémont-Schwarz, I., & Cire, A. A. (2021b). Combining reinforcement learning and constraint programming for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(5), 3677-3687. arXiv:2006.01610

Charikar, M., Chekuri, C., Cheung, T., Dai, Z., Goel, A., Guha, S., & Li, M. (1999). Approximation algorithms for directed Steiner problems. *Journal of Algorithms*, *33*, 73-91. doi: https://doi.org/10.1006/jagm.1999.1042

Dai, H., Dai, B., & Song, L. (2016). Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning* (pp. 2702-2711). PMLR. http://proceedings.mlr.press/v48/daib16.html

Devaurs, D., Siméon, T., & Cortés, J. (2016). Optimal path planning in complex cost spaces with sampling-based algorithms. *IEEE Transactions on Automation Science and Engineering*, *13*(2), 415-424. doi: 10.1109/TASE.2015.2487881

Dijkstra, E.W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*, *1*, 269-271.

Dreyfus, S. E. & Wagner, R. A. (1972). The Steiner problem in graphs. *Networks*, *1*, 195-207. https://doi.org/10.1002/net.3230010302

Erdös, P., & Rényi, A. (2011). On the evolution of random graphs. In *The structure and dynamics of networks* (pp. 38-82). Princeton University Press. https://doi.org/10.1515/9781400841356.38

Gabriely, Y. & Rimon, E. (2008). CBUG: A quadratically competitive mobile robot navigation algorithm. *IEEE Transactions on Robotics*, *24*(6), 1451-1457. doi: 10.1109/TRO.2008.2006237

Grandoni, F., Laekhanukit, B., & Li, S. (2019). O (log2 k/log log k)-approximation algorithm for directed steiner tree: A tight quasi-polynomial-time algorithm. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (pp. 253-264). https://dl.acm.org/doi/abs/10.1145/3313276.3316349

Gross, J. T. & Yellen, J. (2006). *Graph Theory and Its Applications* (2nd ed.), Boca Raton, FL: CRC Press.

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 1025-1035).

Hofmann-Wellenhof, B., Legat, K., & Wieser, M. (2003). Navigation: Principles of Positioning and Guidance. Springer.

Hsieh, M.-I., Wu, E. H.-K., & Tsai, M.-F. (2006). FasterDSP: A faster approximation algorithm for directed Steiner tree problem. *Journal of Information Science and Engineering*, *22*, 1409-1425.

Jiang, W., Li, Y., & Rizos, C. (2017). A multisensor navigation system based on an adaptive fault-tolerant GOF algorithm. *IEEE Transactions on Intelligent Transportation Systems*, *18*(1), 103-113. doi: 10.1109/TITS.2016.2562700

Joshi, C. K., Laurent, T., & Bresson, X. (2019). An efficient graph convolutional network technique for the travelling salesman problem. *arXiv preprint arXiv:1906.01227.*

Jost, T., Wang, W., Fiebig, U., & Pérez-Fontán, F. (2014). A wideband satellite-to-indoor channel model for navigation applications. *IEEE Transactions on Antennas and Propagation*, *62*(10), 5307-5320. doi: 10.1109/TAP.2014.2344097

Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations* (pp. 85-103). Springer, Boston, MA. https://doi.org/10.1007/978-1-4684-2001-2_9

Khalil, E., Dai, H., Zhang, Y., Dilkina, B., & Song, L. (2017). Learning combinatorial optimization algorithms over graphs. *Advances in Neural Information Processing Systems, 30*, 6348-6358. arXiv:1704.01665

Kipf, T. N. & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. Published as a conference paper at ICLR 2017. *arXiv preprint arXiv:1609.02907*.

Kool, W., Hoof, H. van, Gromicho, J., & Welling, M. (2021). Deep policy dynamic programming for vehicle routing problems. *arXiv preprint arXiv:2102.11756.*

LeCun, Y. & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), 255-258.

Li, Z., Chen, Q., & Koltun, V. (2018). Combinatorial optimization with graph convolutional networks and guided tree search. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (pp. 537-546). arXiv:1810.10659

Liu, B., Xiao, X., & Stone, P. (2021). A lifelong learning approach to mobile robot navigation. *IEEE Robotics and Automation Letters*, *6*(2), 1090-1096. doi: 10.1109/LRA.2021.3056373

Loizou, S. G. (2017). The navigation transformation. *IEEE Transactions on Robotics*, *33*(6), 1516-1523. doi: 10.1109/TRO.2017.2725323

Mancini, A., Frontoni, E., & Zingaretti, P. (2015). Embedded multisensor system for safe point-to-point navigation of impaired users. *IEEE Transactions on Intelligent Transportation Systems*, *16*(6), 3543-3555. doi: 10.1109/TITS.2015.2489261

Mazyavkina, N., Sviridov, S., Ivanov, S., & Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. *Computers and Operations Research*, *134*, 105400. https://doi.org/10.1016/j.cor.2021.105400

Muralidharan, A. & Mostofi, Y. (2019). Path planning for minimizing the expected cost until success. *IEEE Transactions on Robotics*, *35*(2), 466-481. doi: 10.1109/TRO.2018.2883829

Niu, X., Li, Y., Kuang, J., & Zhang, P. (2019). Data fusion of dual foot-mounted IMU for pedestrian navigation. *IEEE Sensors Journal*, *19*(12), 4577-4584. doi: 10.1109/JSEN.2019.2902422

Nowak, A., Villar, S., Bandeira, A. S., & Bruna, J. (2017). A note on learning algorithms for quadratic assignment with graph neural networks. *Proceedings of the 34 th International Conference on Machine Learning, Sydney, Australia*. *arXiv preprint arXiv:1706.07450*.

Patil, S., Berg, J. van den, Curtis, S., Lin, M. C., & Manocha, D. (2011). Directing crowd simulations using navigation fields. *IEEE Transactions on Visualization and Computer Graphics*, *17*(2), 244-254. doi: 10.1109/TVCG.2010.33

Pennock, D. M., Flake, G. W., Lawrence, S., Glover, E. J., & Giles, C. L. (2002). Winners don't take all: Characterizing the competition for links on the web. In *Proceedings of the National Academy of Sciences*, *99*(8), 5207-5211. https://doi.org/10.1073/pnas.032085699

Renaudin, V., Dommes, A., & Guilbot, M. (2017). Engineering, human, and legal challenges of navigation systems for personal mobility. *IEEE Transactions on Intelligent Transportation Systems*, *18*(1), 177-191. doi: 10.1109/TITS.2016.2563481

Rose, C., Britt, J., Allen, J., & Bevly, D. (2014). An integrated vehicle navigation system utilizing lane-detection and lateral position estimation systems in difficult environments for GPS. *IEEE Transactions on Intelligent Transportation Systems*, *15*(6), 2615-2629. doi: 10.1109/TITS.2014.2321108

Rothvoß, T. (2011). Directed Steiner tree and the Lasserre hierarchy. *arXiv preprint arXiv:1111.5473*.

Sato, R., Yamada, M., & Kashima, H. (2019). Approximation ratios of graph neural networks for combinatorial problems. *Advances in Neural Information Processing Systems*, *32*, 4081-4090. arXiv:1905.10261

Shi, D., Collins, E. G., & Dunlap, D. (2007). Robot navigation in cluttered 3-D environments using preference-based Fuzzy behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *37*(6), 1486-1499. doi: 10.1109/TSMCB.2007.904581

Smith, K. A. (1999). Neural networks for combinatorial optimization: a review of more than a decade of research. *INFORMS Journal on Computing*, *11*(1), 15–34. https://doi.org/10.1287/ijoc.11.1.15

Syed, Z. F., Aggarwal, P., Niu, X., & El-Sheimy, N. (2008). Civilian vehicle navigation: Required alignment of the inertial sensors for acceptable navigation accuracies. *IEEE Transactions on Vehicular Technology*, *57*(6), 3402-3412. Doi: 10.1109/TVT.2008.921616

Takahashi, H. & Matsuyama, A. (1980). An approximate solution for the Steiner problem in graphs. *Mathematica Japonica*, 24, 573-577. http://pascal-francis.inist.fr/vibad/index.php?action=getRecordDetail&idt=PASCAL8030328467

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. Published as a conference paper at *ICLR 2018*. arXiv preprint arXiv:1710.10903.

Vesselinova, N., Steinert, R., Perez-Ramirez, D. F., & Boman, M. (2020). Learning combinatorial optimization on graphs: A survey with applications to networking. *IEEE Access*, 8, 120388-120416. doi: 10.1109/ACCESS.2020.3004964

Votion, J. & Cao, Y. (2019). Diversity-based cooperative multivehicle path planning for risk management in costmap environments. *IEEE Transactions on Industrial Electronics*, 66(8), 6117-6127. doi: 10.1109/TIE.2018.2874587

Vu, A., Ramanandan, A., Chen, A., Farrell, J. A., & Barth, M. (2012). Real-time computer Vision/DGPS-Aided inertial navigation system for lane-level vehicle navigation. *IEEE Transactions on Intelligent Transportation Systems*, *13*(2), 899-913. doi: 10.1109/TITS.2012.2187641

Wang, D., Xu, X., Yao, Y., Zhang, T., & Zhu, Y. (2020). A novel SINS/DVL tightly integrated navigation method for complex environment. *IEEE Transactions on Instrumentation and Measurement*, 69(7), 5183-5196. doi: 10.1109/TIM.2019.2955187

Watel, D. & Weisser, MA. (2016). A practical greedy approximation for the directed Steiner tree problem. *Journal of Combination Optimization*, 32(4), 1327–1370. https://link.springer.com/article/10.1007/s10878-016-0074-0

Win, M. Z., Conti, A., Mazuelas, S., Shen, Y., Gifford, W. M., Dardari, D., & Chiani, M. (2011). Network localization and navigation via cooperation. *IEEE Communications Magazine*, 49(5), 56-62. doi: 10.1109/MCOM.2011.5762798

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., & Macherey, K. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning* Systems, 32(1), 4-24. doi: 10.1109/TNNLS.2020.2978386

Yen, B. P. C. & Wan, Y. W. (2010). Design and evaluation of improvement method on the web information navigation – A stochastic search approach. *Decision Support Systems*, *49*(1), 14-23. https://doi.org/10.1016/j.dss.2009.12.004

Zelikovsky, A. (1997). A series of approximation algorithms for the acyclic directed Steiner tree problem. *Algorithmica*, *18*(1), 99-110. https://link.springer.com/article/10.1007/BF02523690