Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 199 (2022) 681–686

Procedia Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021)

The neural network methods for solving Traveling Salesman Problem

Yong Shi[a,b,c,d], Yuanying Zhang[b,c,e]*

[a]School of Economics and Management, University of Chinese Academy of Sciences, Beijing, 100190, China
[b]Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing, 100190, China
[c]Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing, 100190, China
[d]College of Information Sciences and Technology, University of Nebraska at Omaha, NE 68182, USA
[e]School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, 100190, China

Abstract

Traveling Salesman Problem(TSP) is a main attention issue at present. Neural network can be used to solve combinatorial optimization problems. In recent years, there have existed many neural network methods for solving TSP, which has made a big step forward for solving combinatorial optimization problems. This paper reviews the neural network methods for solving TSP in recent years, including Hopfield neural network, graph neural network and neural network with reinforcement learning. Using neural network to solve TSP can effectively improve the accuracy of the approximate solution. Finally, we put forward the prospect of solving TSP in the future.

Keyword: Traveling Salesman Problem; Neural network; Hopfield neural network; Graph neural network; Reinforcement learning.

1. Introduction

Traveling Salesman Problem(TSP) is a famous NP hard problem in combinatorial optimization [1]. And there is no algorithm that can find the optimal solution in polynomial time. The specific problem description is that a traveler wants to travel to n cities, and he is required to travel to each city only once and then return to the city he started from, making the whole distance covered the shortest.

The mathematical model of the TSP is in Eqs. (1)-(5).

* Corresponding author. Tel.: +17739367895.
E-mail address: 510786116@qq.com.

10.1016/j.procs.2022.01.084

For weighted graphs G=(V, E), objective function is

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{1}$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = 1, i \in V \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} = 1, j \in V \tag{3}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \forall S \subset V, 2 \leq |S| \leq n-1, x_{ij} = \{0,\ 1\}, i,j \in E \tag{4}$$

Here Equations (2) and (3) guarantee that only one arc as the starting point and only one arc as the end point in the loop, and Equation (4) guarantees that no subloop solutions will be generated.

The algorithms for solving TSP include exact algorithm and approximate algorithm [2]. The exact algorithm includes branch and bound method [3], dynamic programming method [4]. The approximate algorithms include genetic algorithm [5], ant colony algorithm [6], simulated annealing algorithm [7], particle swarm optimization algorithm [8] and their hybrid algorithm [9-11]. With the increase of the solution scale, the time needed to find the optimal solution is explosive growth, that is, the algorithm complexity is too high. Moreover, some approximate algorithms such as ant colony algorithm, its convergence is slow and easy to fall into local optimal solution [12], which can lead to the low optimization efficiency.

At present, no one has worked out a perfect algorithm to solve the TSP problem. However, it can only produce some approximate solutions that can be as close to the perfect solution as possible in a reasonable running time. In recent years, with the continuous development of deep learning with neural network, many new approximate algorithms using neural network have emerged to solve TSP. Compared with traditional algorithms, the error rate of solution has been significantly improved [13]. The main work of this paper is to enumerate the neural network methods for solving TSP, and point out the direction for future research.

2. Neural network methods for solving TSP

Neural network is an algorithm of machine learning. It can solve TSP by machine learning which trains neural networks with different structures, making it possible to solve TSP more accurately. we will introduce the algorithm of solving TSP through the following kinds of neural networks.

2.1. Hopfield neural network

In 1985, Hopfield designed the fully connected network which is later known as Hopfield neural network [14]. He simulated the TSP of 10 cities and 30 cities respectively. Moreover, it was the first algorithm that used neural network to solve the optimization problem of TSP. As shown in Fig. 1, the algorithm idea is to convert the objective function into the energy function of the neural network, and minimize the energy function in the process of running the neural network so as to obtain the local optimal solution. Hopfield defined the energy function and the equations of motion of the TSP problem in Eqs. (5) and (6).

$$E = \frac{A}{2}\sum_{x=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N} X_{xj}X_{xj} + \frac{B}{2}\sum_{i=1}^{N}\sum_{x=1}^{N}\sum_{y=x}^{N} X_{xi}X_{yi} + \frac{C}{2}\left(\sum_{x=1}^{N}\sum_{i=1}^{N} X_{xi} - N\right)^2 + \frac{D}{2}\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{i=1}^{N} d_{xy}X_{xi}\left(X_{y,i+1} + X_{y,i-1}\right) \tag{5}$$

$$\frac{dU_{xi}}{dt} = -\frac{\partial E}{\partial X_{xi}} = -A\left(\sum_{i=1}^{N} X_{xi} - 1\right) - B\left(\sum_{y=1}^{N} X_{yi} - 1\right) - D\sum_{y=1}^{N} d_{xy}X_{y,i+1} \tag{6}$$

Here A, B and C are all positive, which is equivalent to the three penalty terms plus the target function.

Here is a two-layer Hopfield neural network in Fig. 1. The zero layer is the input of the network, and the first layer is the node of the neuron. The neuronal input includes the external input and the feedback of neuronal output in Eq. (7).

$$z_j = \sum_{i}^{N} w_{ij}y_i + x_j \tag{7}$$

Here w_{ij} is weight, $y_i = [y_1, y_2, \cdots, y_N]$ is the neuronal output and $x_j = [x_1, x_2, \cdots, x_N]$ is the external input.

The difference between Hopfield network and other neural networks is that its network weights are not determined through repeated learning, but are directly given, and then the state of the network is updated according to the operating equation of the network, and finally the local optimal solution is reached. Hopfield found that this neural network worked very well for problems smaller than 30 cities. However, It did not apply when the TSP is larger than 30 cities.

Although Hopfield network solves TSP almost perfectly, it also has some shortcomings, which have been optimized by many researchers. Luo [15] improved the third term of the energy function in view of its poor convergence; In addition, Qiao et al [16] also improved the energy function by adding a correction term to the energy function and establishing a new hysteretic noisy frequency conversion sinusoidal chaotic neural network (HNFCSCNN) to solve TSP which received a satisfactory result.

Li et al [17] improved the connection weight according to its shortcoming that it was easy to fall into the local minimum. They changed the connection weights based on the performance of the objective function. García [18] adopted the divide-and-conquer method to improve the performance of Hopfield network in TSP.

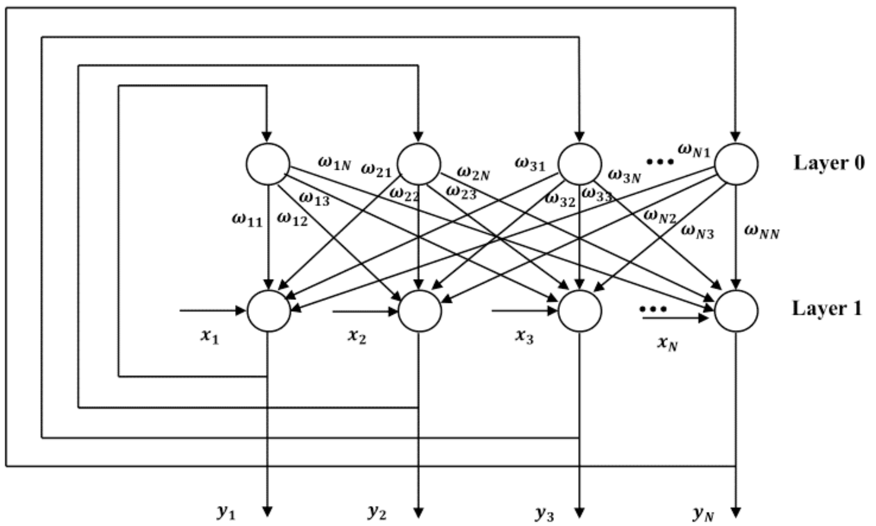

Fig. 1. Two layer of Hopfield network

2.2. Graph neural network

Graph has a complex structure, and it is not easy to combine neural network with graph data. In recent years, a large number of studies have been devoted to the study of graph neural network. The basic idea of graph-neural network is to aggregate the information of each node and its surrounding nodes, and graph neural network has been proved to be able to solve TSP [19].

Prates [20] constructed a graph neural network and trained the model into an effective information transfer algorithm. After several iterations of edges and vertices embedded in weights, then judged whether existed a route that cost$< C$. Joshi [21] proposed a non-autoregressive deep learning method. The Graph ConvNet introduced by Bresson [22] is used to approximate the solution of TSP. The special feature of this method is that the output is the probability of each edge becoming the optimal solution, and the optimal solution is obtained by traversing the whole TSP through beam search. Different from other graph neural networks, Hu et al [23]. constructed a bidirectional graph neural network. In the decoding part, the bidirectional message passing layer was used to predict the probability of the next city visit, and the optimal solution was found by combining the best first search.

2.3. Neural network with reinforcement learning

Reinforcement learning is mainly used to solve sequential decision making problems [24]. Compared with other machine learning, reinforcement learning has no "supervisor" but only "reward" signals. It obtains learning information and updates model parameters by receiving rewards (feedback) from the environment for actions.

In recent years, reinforcement learning has developed rapidly, and it is also very common to solve TSP.
The notable neural network is the Pointer network proposed by Vinyals [25] in 2015, which introduced the attention mechanism and took attention as a Pointer, and then selected a member of the input sequence as the output. It has been proved to be able to solve TSP in combination with beam search. Ma et al [26] used the graph

pointer network proposed by Vinyals and trained the hierarchical graph pointer network with reinforcement learning, and searched for the optimal solution through the hierarchical strategy and reward mechanism. Bello [27] also trained a graph point network with negative travel length as the reward signal, and used the strategy gradient method to optimize the parameters of the repeated current neural network. Deudon et al [28] replaced Bello's LSTM with critic and combined with 2-opt heuristic, then obtained more satisfactory results when applied to TSP.

In addition to using graph pointer networks to combine reinforcement learning, some other networks or systems have been proposed to combine reinforcement learning. For the graphs with complex node context relationships, Dai et al [29] used the Structure2Vec architecture system to embed the vertices of the graph into the feature information, and uses Q-learning training and greedy insertion method to place each new vertex in the local optimal position within the partially formed tour, thus gradually constructing the TSP tour. Bresson [30] used a transformer framework for encoding, which trained with reinforcement learning and decoded with beam search. The transformer solver performed better than other previously proposed TSP solvers.

3. Conclusion

So far, We have reviewed the neural network methods to solve TSP, including Hopfield neural network, graph neural network and neural network with reinforcement learning, and then we put forward suggestions for future research directions:

- Large-scale TSP is still the focus of current research. The computing resources always grow rapidly with the increase of scale. So whether we can improve the precision of large-scale TSP solution by parallelizing neural network or deepening network can become a research direction in the future.
- The complexity of graph data is a big challenge for future research. It is interesting that how to break through the limit of deep learning framework and combine it with reinforcement learning and other algorithms to solve TSP.
- Although graph neural networks are very powerful, whether they can solve other combinatorial optimization problems which called the generalization ability of these graph neural networks remains to be studied.

Acknowledgements

I would like to express my gratitude to all those who helped me during the writing of this thesis. My thanks would go to my beloved family for their loving considerations and great confidence in me all through these years. I also owe my sincere gratitude to my teachers and my fellow classmates who gave me their help and time in listening to me and helping me work out my problems during the difficult course of the thesis. This work was supported by the National Natural Science Foundation of China [No. 11671379, No. 11991021, No. 91546201, No. 71331005, No. 71932008], UCAS, China Grant [No. Y55202LY00].

References

[1] François A, Cappart Q, Rousseau L M. How to Evaluate Machine Learning Approaches for Combinatorial Optimization:Application to the Travelling Salesman Problem. *arXiv preprint arXiv:1909.13121v1,* 2019.

[2] Laporte G. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational* Research, 1992, 59(2):231-247.

[3] Balas E, Toth E. Branch and bound methods for the traveling salesman problem, John Wiley & Sons, Chichester; 1985, p. 361-401.

[4] Chauhan C, Gupta R, Pathak K. Survey of Methods of Solving TSP along with its Implementation using Dynamic Programming Approach. *International Journal of Computer Applications*, 2012, 52(4):12-19.

[5] Xie S L, Yan-Gu Z, Guang L I. Solving TSP Problem by Genetic Algorithm. *Journal of Wenzhou Teachers College*, 2002.
[6] Liang M Y . An ant colony optimization method for generalized TSP problem. *Progress in Natural Science*, 2008.
[7] Wang Z, Geng X, Shao Z. An Effective Simulated Annealing Algorithm for Solving the Traveling Salesman Problem. *Journal of Computational and Theoretical Nanoscience*, 2009, 6(7):1680-1686.
[8] Wang K P, Huang L, Zhou C G, et al. Particle swarm optimization for traveling salesman problem; 2003.
[9] Guillermo, Cabrera G, et al. A Hybrid Particle Swarm Optimization - Simulated Annealing Algorithm for the Probabilistic Travelling *Salesman* Problem. *Studies in Informatics and Control*, 2012, 21(1):49-58.
[10] Petr Stodola, Karel Michenka, Jan Nohel, Marian Rybanský. Hybrid Algorithm Based on Ant Colony Optimization and Simulated *Annealing* Applied to the Dynamic Traveling Salesman Problem. *Entropy*, 2020, 22(8).
[11] Dong L. , et al. Research on improved particle-swarm-optimization algorithm based on ant-colony-optimization algorithm. *29th Chinese Control And Decision Conference (CCDC) IEEE*, 2017.
[12] Song X, Bing L, Yang H. Improved Ant Colony Algorithm and its Applications in TSP. IEEE Computer Society, 2006.
[13] Pihera J, and N. Musliu. Application of Machine Learning to Algorithm Selection for TSP. *IEEE International Conference on Tools with Artificial Intelligence IEEE,* 2014.
[14] Hopfield J J, Tank D W. "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, 1985, 52(3):141-152.
[15] Luo Y.F. Design and Improvement of Hopfield network for TSP. *Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science*, 2019, p. 98-102.
[16] Qiao J , Hu Z , Li W . Hysteretic noisy frequency conversion sinusoidal chaotic neural network for traveling salesman problem. *Neural Computing & Applications*, 2018, p.1-15.
[17] Rong, L, J. Qiao, W. Li . A modified hopfield neural network for solving TSP problem. *2016 12th World Congress on Intelligent Control and Automation (WCICA) IEEE*, 2016.
[18] García L, Talaván P.M, Yáñez J. Improving the Hopfield model performance when applied to the traveling salesman problem, *Soft Comput*. 2016, p. 1–15.
[19] Joshi C K, Ca Pp Art Q, Rousseau L M, et al. Learning TSP Requires Rethinking Generalization. *arXiv preprint arXiv:2006.07054v1*, 2020.
[20] Prates M O, Avelar P H, Lemos H, Lamb L, Vardi M. Learning to solve np-complete problems-a graph neural network for the decisiontsp," *arXiv preprint arXiv:1809.02721*, 2018.
[21] Joshi C K, Laurent T, Bresson X. An Efficient Graph Convolutional Network Technique for the Travelling Salesman Problem. *arXiv preprint* arXiv:1906.01227v2, 2019.
[22] Bresson X, Laurent T. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.
[23] Yujiao, Hu et al. A bidirectional graph neural network for traveling salesman problems on arbitrary symmetric graphs. *ScienceDirect. Engineering Applications of Artificial Intelligence 97*, 2021.
[24] Otterlo M V, Wiering M. Reinforcement Learning and Markov Decision Processes. *Springer Berlin Heidelberg*, 2012.
[25] Vinyals O, Fortunato M, Jaitly N. Pointer Networks. *Computer Science*, 2015, 28.
[26] Ma Q , Ge S , He D , et al. Combinatorial Optimization by Graph Pointer Networks and Hierarchical Reinforcement Learning. *arXiv preprint arXiv:1911.04936v1,* 2019
[27] Bello I, Pham H, Le Q V, et al. Neural Combinatorial Optimization with Reinforcement Learning. *arXiv preprint arXiv:1611.09940v32016,* 2015
[28] Deudon M, Cournut P, Lacoste A, et al. Learning Heuristics for the TSP by Policy Gradient. International conference on integration of constraint programming, artificial intelligence, and operations research. *Springer*, 2018.
[29] Hanjun Dai, Elias B. Khalil, Yuyu Zhang, Bistra Dilkina, Le Song. Learning Combinatorial Optimization Algorithms over Graphs. *In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*. 2017, p. 6348–6358
[30] Bresson X , Laurent T . The Transformer Network for the Traveling Salesman Problem. *arXiv preprint arXiv:2103.03012v1*, 2021.