Research on terminal logistics distribution based on reinforcement learning and pointer network

Chunxin Dong[a], Jintian Ge[*bc], Hongmei Yan[c], Shuai Mi[a]
[a]School of Information Science and Engineering, University of Jinan, Jinan 250022, Shandong, China; [b]Industry School of Standardization, University of Jinan, Jinan 250002, Shandong, China; [c]Business School, University of Jinan, Jinan 250002, Shandong, China

ABSTRACT

The path planning problem at the end of the logistics is an important task for logistics companies today. Due to the continuous expansion of the logistics scale and the requirements of customer delivery, there is a higher requirement for the path planning algorithm. This paper proposes a distribution path planning algorithm based on pointer network. For the training based on supervised learning, the final results will depend on the quality of the solution in the training data, and the optimal solution is difficult to obtain. In this paper, we propose a reinforcement learning method to train the model, which solves the problem of high cost of training samples and improves model accuracy. Because the input of the terminal logistics distribution problem has the characteristics of sequence independence, this paper studies the multi-head attention mechanism. By capturing the multiple features in the input sequence information, the speed of network convergence is accelerated. The experimental results and analysis indicate that the model proposed in this paper has great improvement in solving speed and good performance in effect.

Keywords: Terminal logistics distribution, pointer network, deep reinforcement learning, multi-headed attention

1. INTRODUCTION

Terminal logistics distribution is to meet the end of the distribution link customers for the purpose of the goods to customers, so as to solve the centralized scale effect of logistics and customer dispersed demand. The most important scene of terminal logistics distribution is express delivery and instant delivery, the latter includes takeaway, fresh, shopping malls, retail, with the end of the market demand for logistics distribution is also growing, but also accompanied by customers ' desire for low-cost and fast delivery, which brings new challenges. Due to the characteristics of terminal logistics, such as scattered service locations and distribution locations, there are still many problems in the actual distribution process leading to low quality of distribution services. How to reduce the cost of terminal logistics distribution and improve the speed of distribution services is now an urgent problem to be solved.

The terminal logistics distribution path planning problem is usually NP-hard problem. For solving NP-hard problem, the existing algorithms can be classified into two categories: exact solution and approximate solution. The exact solution is to obtain the global optimal solution. The branch and bound methods are used to search the global search space. The cost is long time and consumes more resources, but the optimal solution can be found. The objective of the approximate solution is to find a better solution close to the optimal solution, which cannot be proved to be the optimal solution, but it takes a short time and consumes less resources. The traditional approximate solution is heuristic algorithm, such as genetic algorithm[1], simulated annealing algorithm[2], particle swarm optimization algorithm[3], ant colony algorithm[4] and so on. These heuristic algorithms obtain the approximate solution of a combinatorial optimization problem by simulation. However, these algorithms need to design algorithms artificially and specifically for each different problem. Once the description of the problem changes slightly, these algorithms need to be modified accordingly. And the existing algorithms can effectively solve such problems when the problem size is small, but with the problem size increasing, the calculation time of the algorithm may increase exponentially. In contrast, the machine learning method is suitable for many tasks, and new heuristic methods can be found. Therefore, less manual engineering is needed than the solver for only one task optimization.

In this paper, aiming at the shortcomings of existing algorithms, a terminal logistics distribution strategy is proposed based on deep learning technology. Taking the terminal logistics distribution problem model as an example, a pointer network[5]

* se_gejt@uijn.edu.cn

Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12506, 1250644
 · 0277-786X · doi: 10.1117/12.2662605

is constructed to learn and solve the problem. Different from the traditional algorithm, the traditional algorithm is only aimed at the optimal solution of the problem, and this method can learn the strategy of problem solving. Aiming at the shortcomings of supervised learning, the training method of reinforcement learning is used to replace the original supervised learning. Since supervised learning needs a large number of known optimal solutions, the pointer network faces the problem that the training set is difficult to find. Reinforcement learning generates training data by itself, and can be trained without problem labels. More importantly, the pointer network uses the solution obtained by the traditional algorithm as a label to train, so the upper limit of its performance is the traditional algorithm, which limits its performance. In the traditional pointer network model, the problem is regarded as a sequence correlation, but for the path problem, the problem sequence is irrelevant to the solution. Therefore, this paper proposes to use the multi-headed attention mechanism[6] to understand the input sequence from multiple perspectives and capture the multiple features in the input sequence information.

2. TERMINAL LOGISTICS DISTRIBUTION

The essence of the terminal logistics distribution problem can be seen as a traveling salesman problem. The problem can be described as follows: a distributor needs to go to several demand points to distribute goods. Starting from the distribution center, the distributor needs to go through and can only go through all demand points once before returning to the distribution center. The distributor needs to choose a shortest path from all possible paths.

The mathematical model is as follows. Given the two-dimensional coordinates of n demand points and these n demand points, the distance between any two demand points in this demand point is also fixed. Starting from the distribution center, a distributor must traverse the remaining $n-1$ demand points, and finally return to the distribution center. This should traverse the n demand points in what order to minimize the total travel distance.

In order to better describe this problem, the i demand point is expressed by x_i, and the distance from the i demand point to the j demand point x_j is expressed by d_{ij}. Then the mathematical model of the terminal logistics distribution problem can be expressed as follows:

$$X = \{x_1, x_2, \mathrm{L}\ , x_n\} \tag{1}$$

$$minD = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} \tag{2}$$

where equation (1) represents the access sequence, $minD$ represents the distance of the optimal path, and n represents the number of demand points.

3. Neural Network Architecture

3.1 Pointer network

Considering that the output of the end logistics problem is undoubtedly a sequence composed of demand points, and the input (demand point set or distance matrix) can be organized as a sequence (point or edge sequence), the seq2seq model[7] is used to solve this problem.

Pointer network is mainly used to solve combinatorial optimization class problems, which can learn the probability of the output sequence from the input sequence data, i.e., from the solution of the problem sequence problem. The structure diagram of the pointer network is shown in Figure 1, x_n and y_n denote the input sequence.

In Figure 1, the left white RNN is used to process the input sequence to generate the encoding vector, where the encoding vector is used to generate the output sequence and another right gray RNN using the probabilistic association rule. The conditional probability of the output is calculated by softmax, and the calculation formula is as follows:

$$u_j^i = v^T \tanh\left(W_1 e_j + W_2 d_i\right), j \in \left(1, 2, \mathrm{L}\ , n\right) \tag{3}$$

$$P\left(C_i \mid C_1, C_2, \ ..., C_{i-1}, P; \theta\right) = softmax\left(u^i\right) \tag{4}$$

The v, W_1, W_2 are the parameters obtained after model training, and u_j^i is the input pointer calculated by tanh. Equation

(4) expresses the conditional probability of the output element calculated by softmax.

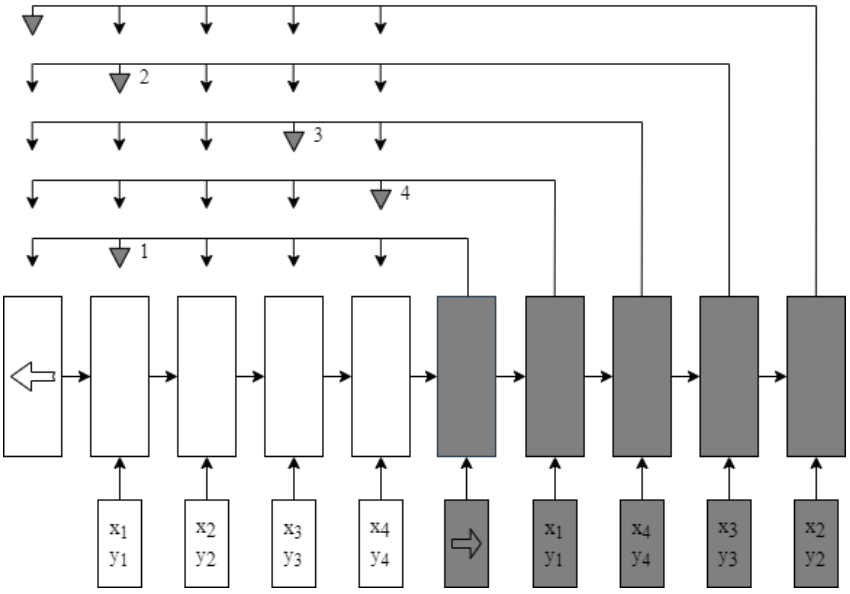

Figure 1. Pointer network model.

3.2 Deep reinforcement learning

Due to the combinatorial optimization problems, it is difficult to obtain the optimal solution. If the supervised learning method is used, the large amount of sample data required for training cannot be obtained, and the terminal logistics distribution problem has a relatively simple reward mechanism. Reinforcement learning is an effective network training method.

This paper uses policy gradient-based reinforcement learning to optimize the parameters θ of the network. For a specific set of requirements S, the optimization objective of the network is to minimize the path expectation $J(\theta|S) = E_{\pi\sim p_{\theta(\cdot|S)}} L(\pi|S)$, which represents the expected cumulative loss obtained by action according to strategy π when the problem demand set S is determined. This algorithm directly uses the Reinforcer algorithm[8] to optimize the actor network:

$$V_\theta J(\theta \mid S) = E_{\pi\sim p\theta(\mathrm{gS})}[(L(\pi \mid s) - b(S))\nabla_\theta \log P\theta(\pi \mid S)] \tag{5}$$

where $b(S)$ represents the baseline function independent of strategy π, and estimates the loop length expectation to reduce the variance of the gradient. A simple and popular choice for the baseline function $b(S)$ is the moving exponential average of the rewards obtained during the training period, although better results can be obtained by directly averaging the moving exponential, the use of parameter baseline to estimate the loop length expectation $E_{\pi\sim p\theta(\cdot|S)} L(\pi|S)$ can usually improve the learning effect.

Only based on the strategy gradient optimization, the training speed is too slow, so using the actor-critic structure[9] and the actor-critic structure is a method that combines the advantages of value optimization and strategy gradient optimization. Compared with the traditional policy gradient model, it has faster round update and faster convergence speed. The network structure is shown in Figure 2.

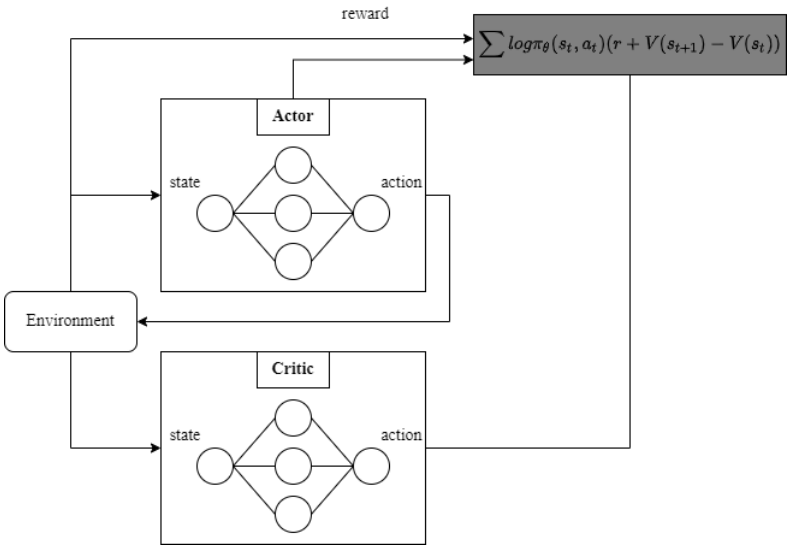

Figure 2. Actor-Critic structure diagram.

The network of commentators is used as an auxiliary network and is parameterized by θ_v. Its function is to determine the expected loop length of the current strategy p_θ under the given input sequence s. The proposed critic network architecture is a RNN with LSTM[10], where the prediction is based on the final state. The network of reviewers uses stochastic gradient descent to train with mean square error:

$$l(\theta_v) = \frac{1}{B}\sum_{i=1}^{B} \| b_{\theta_v}(s_i) - L(\pi_i \mid S_i) \|_2^2 \tag{6}$$

As a result, Figure 3 shows the structure of the enhanced pointer network model proposed in this paper. Compared with Figure 1, the critic network is added to it, and the action and current state obtained by the actor network are used as inputs to get the estimate of the value of the current state.

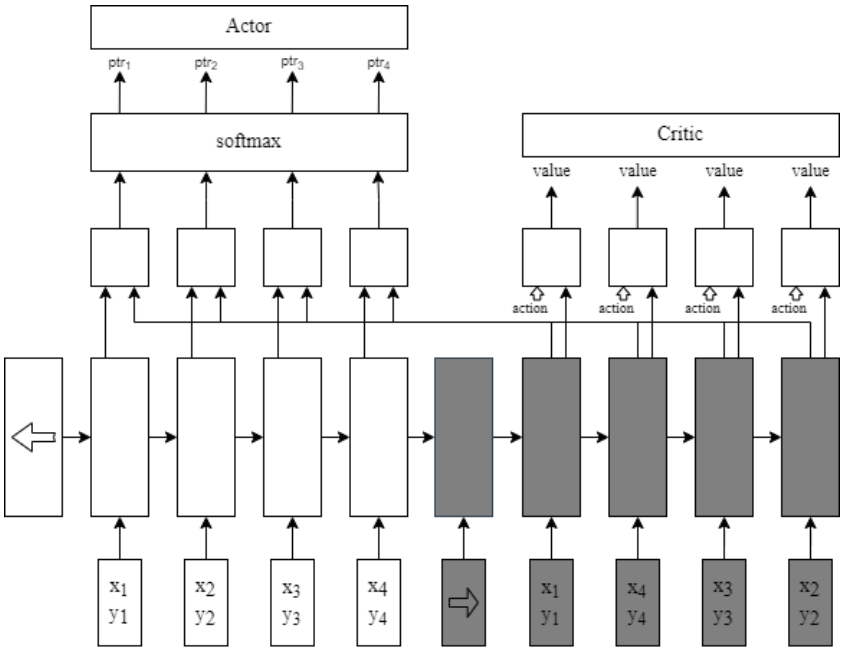

Figure 3. Reinforcement pointer network model.

The actor network part of the pointer network added to reinforcement learning uses the structure of the pointer network to simulate the action strategy $p(C_i|C_1,\ldots,C_{i-1},P)$, as follows:

$$u_j^i = v^T \tanh\left(W_1 e_j + W_2 d_i\right), j \in (1,2,\ldots,n) \tag{7}$$

$$P(C_i \mid C_1,\mathrm{L}\ ,C_{i-1},P) = softmax\left(u_j^i\right) \tag{8}$$

where $softmax$ standardizes the vector u_i (length n) as the output distribution on the input dictionary, v, W_1 and W_2 are the parameters learned by the model. In pointer networks, instead of e_j, the node of the encoder, to spread additional information to the decoder, u_j^i is used as a pointer to the input element.

3.3 Multi-head attention

Compared with the traditional sequence problem, the solution of the terminal logistics distribution problem is independent of the order of the input sequence, so the multi-headed attention mechanism is used to understand the input sequence from multiple perspectives and capture the multiple features in the input sequence information.

The multi-head attention mechanism performs multiple self-attention calculations. Using the feature of query (Q) = key (K) = value (V) can fully capture the long-distance dependence of data within the sequence, and give different attributes different weights. In this paper, the zoom product attention mechanism is adopted. Multiple attention uses different weight matrices W_i^Q, W_i^K and W_i^V to perform h sublinear transformation for Q, K and V, and obtains different packet vector attention h_i^{head}. Finally, the global features are combined and output, which allows efficient learning of key information in different representation subspaces.

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d}k}\right)V \tag{9}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{10}$$

$$Multi(Q,K,V) = Concat\left(head_1,\mathrm{L}\ ,head_h\right)W^O \tag{11}$$

$Attention$ represents the calculation of attention, $head_i$ represents the calculation of the i attention head; W_i^Q, W_i^K, W_i^V represent the weight matrix of matrix Q,K,V; W^O is the corresponding weight matrix after the calculation of h attention heads.

4. EXPERIMENTAL RESULTS

4.1 Date sets

Because of the small sample size of the traditional data set, we sampled the data in a uniform distribution on a two-dimensional space in the range of [0, 1], and used 1,000,000-point set data for the training set and 10,000-point set data for the validation set. In order to facilitate the comparison of three scale data: TSP20, TSP50 and TSP100.

4.2 Setting

In this paper, the experiment is implemented using pytorch. In the experiment, mini-batch was used for training. Each batch adopted 128 sequences. The hidden layer of encoder network, decoder network and critic network were LSTM with 128 hidden units, and the two coordinates of each point were embedded in 128-dimensional space. The proposed model is trained using the Adam optimizer, and the initial learning rates of TSP {20, 50} and TSP100 are 10^{-3} and 10^{-4}, respectively, which are reduced to 0.96 times per 5000 steps. The parameters are randomly and uniformly initialized within [-0.08, 0.08] and the L2 parametric of the gradient is set to 1.0.

4.3 Result analysis

4.3.1 Training Results. The training results are shown in Figures 4 and 5. On the training set and verification set of two-dimensional random uniform distribution data, as the training time increases, the average path length calculated by the algorithm decreases in stages, and the algorithm will not change much in a short time. After accumulating a large number of data, the effect will be greatly improved, and finally tends to converge.

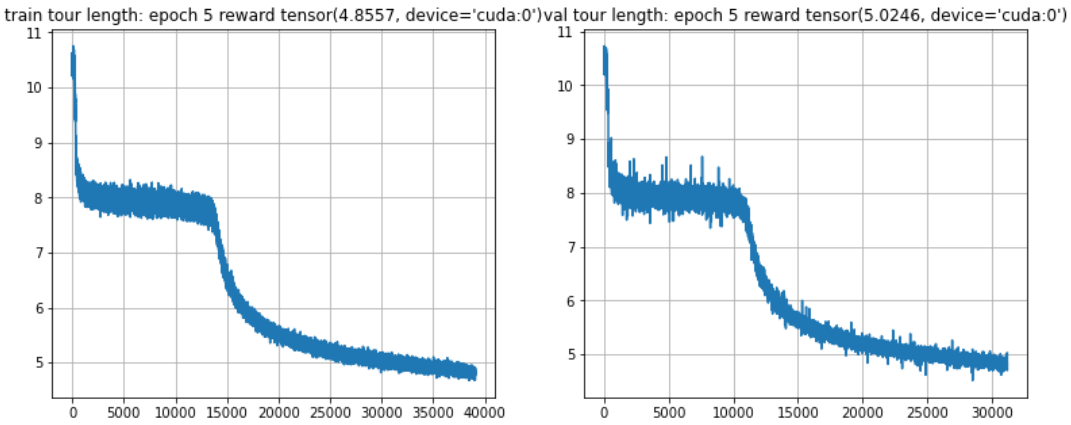

Figure 4. Tsp20 training effect diagram.

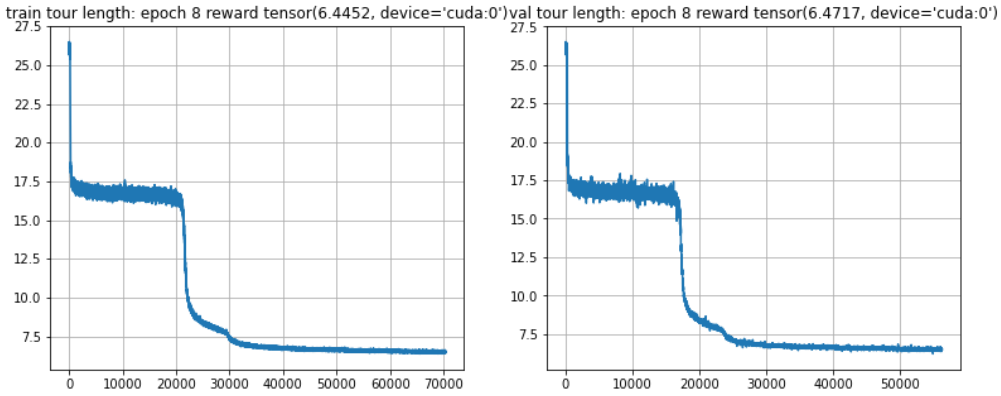

Figure 5. Tsp50 training effect diagram.

After using multiple attention, the model can understand the input sequence from multiple perspectives, capture multiple features in the input sequence information, and improve the convergence speed and effect. In the case of the same number of training samples, the training effect of two periods has reached the first five periods of improvement, and the final convergence effect has increased by 11%, as shown in Figure 6.

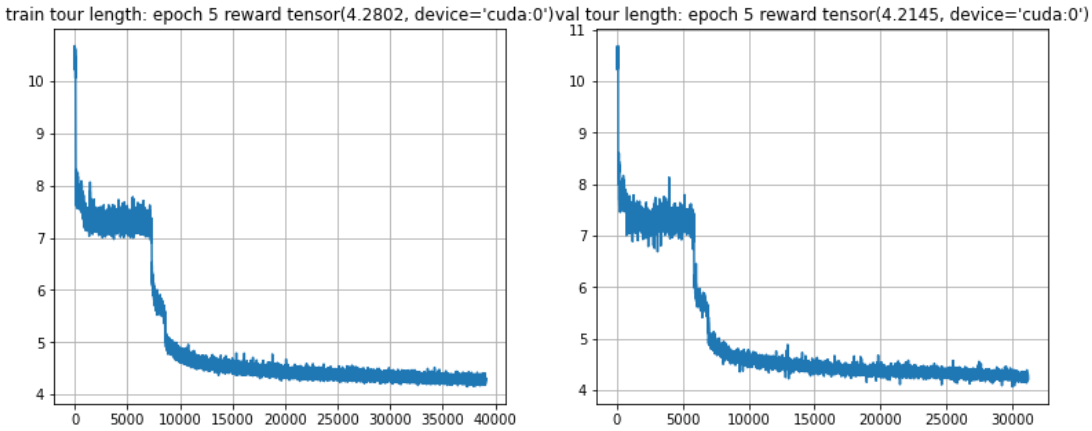

Figure 6. Training effect diagram after multi-attention improvement.

4.3.2 Test Results. In order to show the effect of the proposed model and algorithm more intuitively, this section will use the representative algorithms in the following literature as a benchmark, and compare with the proposed algorithm. (1) Pointer networks for supervised learning: use supervised learning to deploy and train the pointer networks proposed by Vinyals et al. (2) Christofides algorithm: In polynomial time, the solution found by the algorithm is guaranteed to be within the optimal solution range of 1.5 times. (3) General solver: Using a vehicle routing solver from (Google, 2016), it improves Christofides ' solution through simple local search operators and metaheuristic. (4) The most advanced open-source TSP-specific solver: using Concorde and LK-H local search algorithms, using the latest version of LKH-3 in 2018, as shown in Table 1.

Table 1. Comparison of experimental results.

Date set	Pointer networks	Christofides algorithm	General solver	LK-H	Ours
TSP20	3.91	4.37	3.91	3.88	3.88
TSP50	6.69	7.04	6.26	6.14	6.12
TSP100	11.32	9.93	8.23	7.97	7.91

On the two-dimensional random uniform distribution data test set, the enhanced pointer network proposed in this paper has achieved the optimal results in all algorithms. The same results are obtained with the traditional local search algorithm LK-H on small scale problems. In large-scale problems, it surpasses the traditional local search algorithm LK-H, and is far superior to other network algorithms.

4.3.3 Solution Speed Analysis. The solution speed of the algorithm is also an important indicator. Although network algorithms have great disadvantages in training time, they have faster response speed in response time. The network structure of the proposed enhanced pointer network is completely consistent with the ordinary pointer network, so it has better response speed than the traditional local search algorithm, as shown in Table 2.

Table 2. Comparison of solving speed.

	Pointer networks	LKH-3	Ours
TSP50	14s	37s	14s
TSP100	44s	122s	46s

5. CONCLUSION

In this paper, reinforcement learning and pointer network are used to solve the problem of terminal logistics distribution, and the two-dimensional Euclidean graphs of 20, 50 and 100 nodes are tested and compared. The experiments show that the large-scale data can meet the real-time requirements at the present stage, and have good performance in terms of effect, with the advantages of “offline training and online solution”.

REFERENCE

[1] Mirjalili, S., “Genetic algorithm,” Evolutionary Algorithms and Neural Networks, 43-55 (2019).
[2] Aarts, E., Korst, J. and Michiels, W., “Simulated annealing Search Methodologies,” 187-210 (2005).
[3] Wang, D., Tan, D. and Liu, L., “Particle swarm optimization algorithm: An overview,” Soft Computing 22, 387-408 (2018).
[4] Blum, C., “Ant colony optimization: Introduction and recent trends,” Physics of Life Reviews 2, 353-373 (2005).
[5] Vinyals, O., Fortunato, M. and Jaitly, N., “Pointer networks,” Advances in Neural Information Processing Systems 28, (2015).
[6] Vaswani, A., et al., “Attention is all you need,” Advances in Neural Information Processing Systems 30, (2017).
[7] Sutskever, I., Vinyals, O. and Le, Q. V., “Sequence to sequence learning with neural networks,” Advances in Neural Information Processing Systems 27, (2014).
[8] Williams, R. J., “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” Machine Learning

8, 229-256 (1992).
[9] Grondman, I., et al., “A survey of actor-critic reinforcement learning: Standard and natural policy gradients,” IEEE Transactions on Systems, Man, and Cybernetics, Part C 42, 1291-1307 (2012).
[10] Yu, Y., et al., “A review of recurrent neural networks: LSTM cells and network architectures,” Neural Computation 31, 1235-1270 (2019).