IEICE TRANS. COMMUN., VOL.E104–B, NO.1 JANUARY 2021

PAPER

Optimal Planning of Emergency Communication Network Using Deep Reinforcement Learning

Changsheng YIN[†a)], **Ruopeng YANG**[†], **Wei ZHU**[†], **Xiaofei ZOU**[†], ***Nonmembers***, ***and*** **Junda ZHANG**[††], ***Member***

SUMMARY Aiming at the problems of traditional algorithms that require high prior knowledge and weak timeliness, this paper proposes an emergency communication network topology planning method based on deep reinforcement learning. Based on the characteristics of the emergency communication network, and drawing on chess, we map the node layout and topology planning problems in the network planning to chess game problems; The two factors of network coverage and connectivity are considered to construct the evaluation criteria for network planning; The method of combining Monte Carlo tree search and self-game is used to realize network planning sample data generation, and the network planning strategy network and value network structure based on residual network are designed. On this basis, the model was constructed and trained based on Tensorflow library. Simulation results show that the proposed planning method can effectively implement intelligent planning of network topology, and has excellent timeliness and feasibility.

key words: *emergency communication, network planning, reinforcement learning, intelligence*

1. Introduction

Emergency communication networks should be able to effectively to provide timely and reliable communication connection in case of emergency. Different from the traditional communication networks, its unique characteristics include time burst, location uncertainty, business urgency, and transient process [1], the emergency communication network must offer rapid deployment and have certain robustness and scalability to ensure reliable communication services for auxiliary rescue operations. So it has become the key to emergency communication that how to quickly formulate a scientific and effective emergency communication plan. Communication network topology planning is an important part of communication network planning and design. It refers to determining the layout and topology of the communication node with the best comprehensive performance based on the known users' location and service distribution, and according to the existing equipment conditions, and under the given constraints of service coverage and reliability.

At present, there have been many studies on communication network planning problems [2]–[9], mainly using graph theory to model network topology, multi-objective combination optimization considering different communication index constraints, and transforming network planning into optimization problems. Heuristic algorithms such as genetic algorithm (GAs) [3], Tabu search (TS) [4] and simulated annealing (SA) [5] are currently used to solve such optimization problems. In summary, although there have been many studies on communication network planning problems, most of them are based on traditional heuristic algorithms, which have certain requirements on prior experience and historical data to generate initial solution or initial population, which has a great impact on the efficiency of traditional heuristic algorithm. At the same time, there are problems of low effectiveness and being trapped in local optimality. However, methods based on quantitative analysis are not easy to operate and universal, and are difficult to implement effectively. Meanwhile, in recent years, the intelligent technology with deep learning as the core has made great progress, especially the intelligent method represented by deep reinforcement learning is used to solve Atari games, chess game confrontation, and incomplete information real-time strategy games. Numerous achievements have been achieved on the issue that surpass human level [5]–[10], and people have seen the dawn of intelligent decision-making.

Traditional planning algorithms require excessive amounts of prior knowledge and take too long to produce results. Accordingly, this paper draws on the intelligent decision ideas of AlphaZero [5], comprehensively considers the two factors of network coverage and connectivity, and introduces deep reinforcement learning algorithms [6], [7] to solve topology planning in emergency communication networks. First, according to the characteristics of the emergency communication network, learn from chess game ideas, and establish an abstract model of network planning, which realize the abstract mapping of the emergency communication network/node to the chessboard/chess. Then based on the idea of reinforcement learning, Monte Carlo Tree Search (MCTS) [7] combined with self-game was used to generate training sample data to build a residual network (Residual Network, ResNet) [8] Network planning strategy network and value evaluation network to establish a deep neural network model for emergency communication network planning. Based on this, an evaluation function for reinforcement learning is established according to the actual application requirements of the emergency communication network, and the model is trained by using Tensorflow to build a neural network. Finally, the model is obtained for real-time network planning based on the training. The results show that the algorithm is highly feasible, accurate, and

Manuscript received April 16, 2020.
Manuscript publicized June 29, 2020.
[†]The authors are with National University of Defense Technology, Wuhan, 430010 China.
[††]The author is with Naval Aviation University, Yantai, 264000 China.
a) E-mail: yincs1989@163.com
DOI: 10.1587/transcom.2020EBP3061

Copyright © 2021 The Institute of Electronics, Information and Communication Engineers

is more efficient than existing alternatives.

2. Emergency Communication Network Planning Problem

2.1 Problem Description

This paper studies the network topology planning issues in emergency communication networks, that is, how to find out the best economical backbone node and access node layout locations and topological relationships on the premise of meeting reliability and service distribution constraints when the user node's location are known. The user node uses an ultra-short wave radio to establish a communication link with the access node and access the network, while the access node uses a high-speed radio to establish a communication link with the backbone node and access the backbone network, and the backbone node uses microwave relays to establish a communication link which forms a backbone network.

The emergency communication network is modeled by the structure of the graph. The vertices of the graph represent the user node, the access node and the backbone node respectively, and the edge represents the transmission link, and the undirected graph $G = (V, E)$ is used, and the vertex set $V = \{B, A, U\}$ includes three types of nodes, including N backbone nodes $B = \{B_1, B_2, \cdots, B_N\}$, M access nodes $A = \{A_1, A_2, \cdots, A_M\}$, L users Node $U = \{U_1, U_2, \cdots, U_L\}$, as shown in Fig. 1.

2.2 Mathematical Model

1) Link establishment constraints: It means that if the distance between two nodes is within the communication range, a communication link is established. The user node can only access the network through the access node, and the access node can establish a link with the user node and the backbone node. In addition to the backbone nodes being able to establish links with the access nodes, they can also establish links with each other and form a transmission backbone network. The constraint expressions are as follows.

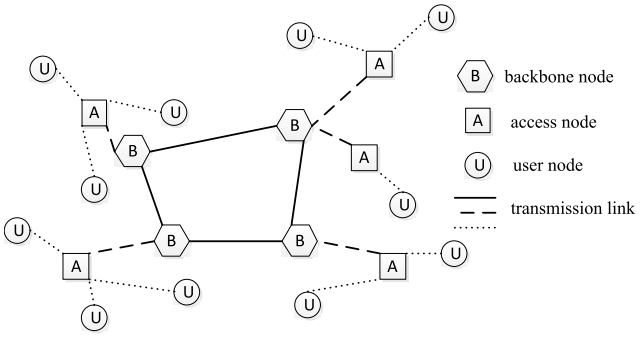


Fig. 1 Emergency communication network topology diagram.

$$E\left(V_i, V_j\right) = \begin{cases} 1, & d_{V_i V_j} \le D \\ 0, & d_{V_i V_j} > D \end{cases} \tag{1}$$

Where $D = \{D_{UA}, D_{AB}, D_{BB}\}$ respectively represent the maximum communication distance between the user node and the access node, the maximum communication distance between the access node and the backbone node, and the maximum communication distance between the backbone nodes.

2) User node access constraints: For any user node, there is at least one access node around it for its access to the network, as shown in Eq. (2).

$$C_{U-A} = \prod_{i=1}^{L}\left(\sum_{j=1}^{M} E(U_i, A_j)\right) \neq 0 \tag{2}$$

Where $E(U_i, A_j)$ indicates whether there is a link between the user node U_i and the access node A_j, and if there is a link, it is 1, otherwise it is 0.

3) Access node access constraints: For any access node, there is at least one backbone node around it for its access to the backbone network, as shown in Eq. (3).

$$C_{A-B} = \prod_{i=1}^{M}\left(\sum_{j=1}^{N} E\left(A_i, B_j\right)\right) \neq 0 \tag{3}$$

Where $E\left(A_i, B_j\right)$ indicates whether there is a link between the access node A_i and the backbone node B_j, and if there is a link, it is 1, otherwise it is 0.

4) Backbone network connectivity constraints: It means that backbone nodes form a connected network, that is, there is a communication link between any backbone nodes, or a path can be formed by relaying through other backbone nodes. Then, the undirected graph $G_B = (V, E)$ composed of all the backbone nodes is a connected graph, that is, for any two backbone nodes B_i and B_j, there are alternating sequences of vertices and edges $\tau = (B_i = v_0 - e_1 - v_1 - e_2 - \cdots - e_k - e_{k+1} = B_j)$.

3. Network Planning Method Based on Deep Reinforcement Learning

In order to solve the problem of deep learning training samples, this paper adopts the network intelligent planning method of emergency communication network based on deep reinforcement learning and game theory. It mainly includes emergency communication network planning element abstraction, MCTS-based sample data generation, deep learning-based model training, and model-based network planning. The specific steps are as follows.

3.1 Feature Abstraction

Firstly, the abstraction of the emergency communication network planning is carried out, including gridding the planning area into a similar chessboard. The user node, the access

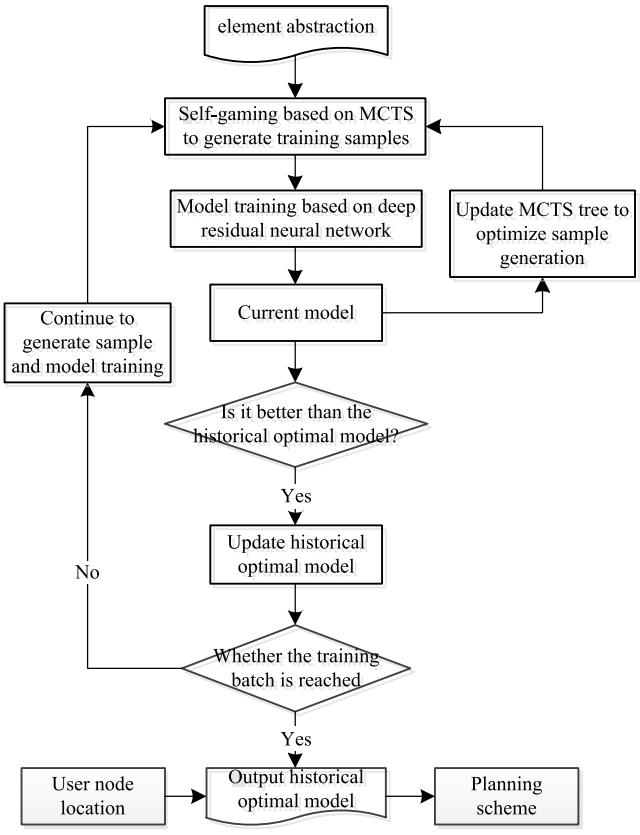


Fig. 2 Schematic diagram of network planning method based on deep reinforcement learning.

node and the backbone node are regarded as chess pieces, wherein the user node position is the initial state of the chessboard, which is different from the general chess class. There is only one player, and the player can play the game including the access node and the backbone node. The winning and losing condition is that after the agreed number of access nodes and backbone nodes are completed, if the constraint in the 2.2 mathematical model is satisfied at the same time. (2)–(4), it is determined that the player wins, otherwise the player loses. At the same time, in order to ensure the actual deployment benefits, whether the number of subordinates is smaller is used to judge the model's pros and cons, and the model is updated and iterated.

3.2 MCTS-Based Training Sample Generation

Intensive learning is used to continuously generate chess game data through self-game. First of all, in order to ensure the diversity of the sample data, first, the initial chess surface of each game, that is, the layout of the user nodes are randomly arranged, and the second is that the strategy model adopted by the game always adopts the latest model generated by the training process to avoid falling into the local space. Then use the model's strategy output and valuation output as the search direction, use the Monte Carlo tree search algorithm to explore the possible action space, and continue to carry out the game according to the probability of returning, until the end of the game, if it wins, the game will be played. Data is entered into the training sample set. Specific steps are as follows:

1) Select: Assume that the current state of the search tree is S, the selection action is a, the connected edge between nodes is $e(s,a)$, and each edge stores a quad set, that is, the number of traversals $N(s,a)$, the cumulative value of the action $W(s,a)$, action mean $Q(s,a)$ and prior probability $P(s,a)$. It is assumed that in each step t from the root node s_0 to the leaf node s_1, it is necessary to determine the action a_t according to the quaternion set stored in the current time search number, that is, select the action with the largest action value in the current state s_t, specifically The calculation method is as follows:

$$a_t = argmax\left(Q\left(s_t,a\right)+U\left(s_t,a\right)\right) \tag{4}$$

$$U\left(s_t,a\right) = c_{puct}P\left(s_t,a\right)\frac{\sqrt{\sum_b N\left(s_t,b\right)}}{1+N\left(s_t,a\right)} \tag{5}$$

$$P\left(s_t,a\right) = \left(1-\epsilon\right)P\left(s_t,a\right)+\epsilon\eta \tag{6}$$

Among them, c_{puct} is the super parameter for balance exploration and utilization, $\sum_b N(s_t,b)$ is all times of the state s_t, $p(s_t,a)$ is the probability of action a in the model strategy output, η To enhance the robust noise, ϵ is the inertia factor.

2) Expand and evaluate: If the search tree is currently in a non-leaf node, the expansion continues, and if the leaf node s_l is reached, the evaluation is performed. According to the current neural network training based model f_θ, the strategy output p_l and the evaluation output v_l are obtained, and the quaternion set of the edge $e\left(s_l,a\right)$ is initialized, that is, $N\left(s_t,a\right)$, $W\left(s_t,a\right)$ and $Q\left(s_t,a\right)$ is 0. The current chess data is then entered into the neural network model to evaluate the current panel.

3) Backup: After the search tree completes the expansion and evaluation, the search structure is started from the leaf node based on the connection side information of each node in the search tree, and is transmitted back to the root node node by node, and the information of the quaternion set information on each side is updated. The update algorithms $N\left(s_t,a_t\right)$, the action cumulative value $W\left(s_t,a_t\right)$, and the action average $Q\left(s_t,a_t\right)$ are as follows:

$$N\left(s_t,a_t\right) = N\left(s_t,a_t\right)+1 \tag{7}$$

$$W\left(s_t,a_t\right) = W\left(s_t,a_t\right)+v_t \tag{8}$$

$$Q\left(s_t,a_t\right) = \frac{W\left(s_t,a_t\right)}{N\left(s_t,a_t\right)} \tag{9}$$

Where v_t is the estimated output of the neural network $f_\theta\left(s_t\right)$. As the number of simulations increases, the action value Q will gradually become stable, and it is not directly related to the strategic output p_t of the neural network.

4) Play: According to the steps 3.2.1 to 3.2.3 above, after 400 Monte Carlo tree searches, using the history information such as the number of accesses of each leaf node stored in each side of the search tree, the simulated annealing algorithm can be used to obtain all the drop probability distributions $\pi\left(a|s_0\right)$, ie:

$$\pi(a|s_0) = \frac{N(s_0,a)^{\frac{1}{\tau}}}{\sum_b N(s_0,b)^{\frac{1}{\tau}}} \tag{10}$$

Among them, τ is the simulated annealing parameter, its function is to avoid the same game opening in each game, and effectively expand the diversity and effectiveness of the search sample.

The above method is used to continuously perform the drop. After each step, the extended child nodes and subtree information of the current search tree are retained until the game ends. Only when the final result of the game is that the player wins, all the corresponding chess data and the probability distribution of the drop under the corresponding chess face are taken as the training sample set and the evaluation set.

5) Sample data expansion: Using the chess board to have the equivalent properties of rotation and mirror flipping, the expansion of the sample data is realized by the rotation and mirroring of the chess surface data. The original sample data is rotated by using $n \times 45^o$, where $n = \{0, 1, \ldots, 7\}$, and a total of 8 times of the original data is generated by this method, and these sample data are collectively used as a neural network. Training input.

3.3 Model Training Based on Deep Residual Neural Network

Using ResNet-based deep neural network, training samples are used to train the strategic value network of network planning. After the fixed batch, the strategy value network model is updated and compared with the historical optimal model, while the optimal model and the latest model are saved. model. In order to ensure the diversity of the exploration space, the latest policy value network of training and updating is applied to the self-game stage in the MCTS to generate better self-game sample data, thereby realizing the loop embedded of the self-game sample data generation and the neural network training process. Set, accelerate the convergence of the training model.

1) Training data description method: The sample data is described by means of a binary feature plane. A total of five *Width*×*Height* planes are included. The first three planes are used to represent the positions of the user nodes, the access nodes and the backbone nodes. The position of the node is 1, otherwise it is 0. The fourth one represents the last step of the player's position. Only one position of the plane is 1, that is, the position of the drop, and the rest of the position is all zero. The fifth plane represents the last step type. If it is an access node, the entire plane is all 0, and if it is a backbone node, it is all 1.

2) Neural network structure: The structure of the neural network is as shown in the figure. First, a four-layer common convolutional network is used, and 32, 64, 128, and 256 3×3 filters are constructed using the Relu function, respectively, and then divided into a planning strategy network (Policy). Network) and Value Network two branches, the strategy network branch uses four 1×1 dimensionality reduction filter, a fully connected layer, using the softmax function to output the selection probability P of each node in the planning space, value network branch Use two 1×1 dimensionality reduction filters, one fully connected layer, and use the tanh function to output a range of [0, 1] to score C, i.e. $f_\theta(s) = (P, C)$.

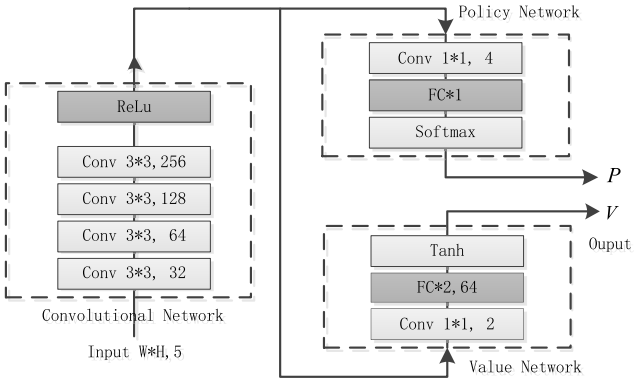


Fig. 3 Neural network structure.

3) Training objectives: It can be seen from the above that the input of the strategy network and the value network is a description of the current situation, the output is the probability of each actionable in the current situation and the score of the current situation, and the training for the strategy value network is based on the sample data generated by the game. Therefore, the training goal is to make the probability of the action output by the strategy network approach the probability of the MCTS output, and let the value score of the output of the value network more accurately predict the true final result, that is, the loss function is minimized by training, and the loss is lost. The function is as follows:

$$Loss = (C' - C)^2 - \pi^T log P + g \parallel \theta \parallel \tag{11}$$

3.4 Model-Based Network Topology Planning

Network topology planning is based on the neural network model generated by training for offline planning. The model output during the training process has saved all the parameters of the neural network. In the planning stage, a binary feature plane containing the initial situation information such as the position of the user node is established, which is used as the input of the neural network, and the output of the neural network is Network planning result.

4. Simulation

4.1 The Training and Analysis of Algorithm

The experimental platform computer CPU is Inter Core i5-8210, the memory is 8 G, the hard disk is 1 TB, the system is Ubutun18, the specific code of the program is written by python language and open source package, and the neural network is built by using Tensorflow library. Considering the computing power of the computer, this paper rasterizes

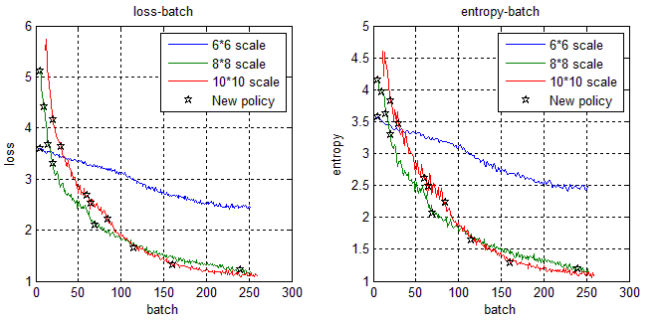


Fig. 4 Relationship between loss, entropy and training batch.

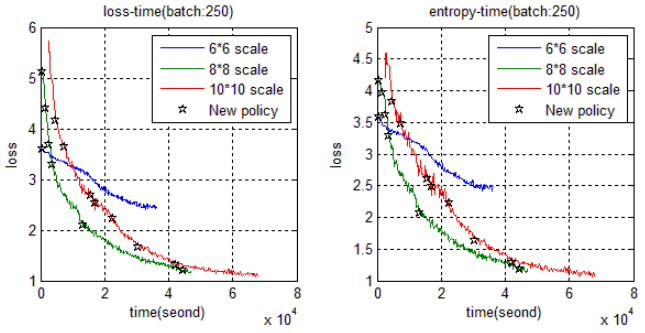


Fig. 5 Relationship between loss, entropy and training time.

the area with a scale of 0.5 km per grid. With a flat terrain of 3 km×3 km, 4 km×4 km, 5 km×5 km as the background, the communication area grid is 6×6, 8×8, 10×10. The maximum communication distance $D_{UA} = D_{AB} = 0.5$ Km (1 grid), $D_{BB} = 2$ Km (4 grids), and the number of user nodes is 4. The termination condition of the training is 250 rounds of training. Through the application of the model check, three models of the model can be effectively used to plan the emergency communication network after completing 250 training sessions.

As shown in Fig. 4, in the figure, with the increase of the number of training stations, the loss value and entropy value of the three scales can be continuously decreased, and the initial loss value of the $6 * 6$ scale due to the small search space. And the entropy value is the smallest, and the new strategy is first generated. Although the entropy and loss values decrease slowly, the strategy is the first to be stable, that is, the convergence is faster; for 8×8 and 10×10 In larger cases, the initial loss value and entropy value will be larger, and as the number of training stations increases, the loss value and entropy value decrease continuously, and a better strategy is continuously generated, but even the loss value and entropy value have been It has dropped below 2, although its entropy and loss values have dropped rapidly, but its strategy is still constantly updated. Even after training 250 stations, it has not stabilized, and its convergence speed is relatively slow.

As shown in Fig. 5, in the case that the number of training stations is 250, as the training time increases, the loss value and entropy value of the three scales can be continuously decreased, and the 6×6 scale is due to the search space. Small, the training speed is the fastest, it takes only 1 hour and 4 minutes to complete 250 training sessions, and its strategy is no update after 5 minutes, and the actual model application verification shows that the strategy can meet the network planning requirements; In the case of larger 8×8 and 10×10, the initial loss value and entropy value will be larger, and as the training time increases, the loss value and entropy value decrease continuously, and a better strategy is continuously generated, but even After the training time reaches 1 hour, the loss value and entropy value are still decreasing, and the resulting strategy model is not ideal in practical applications, and its convergence time is relatively longer.

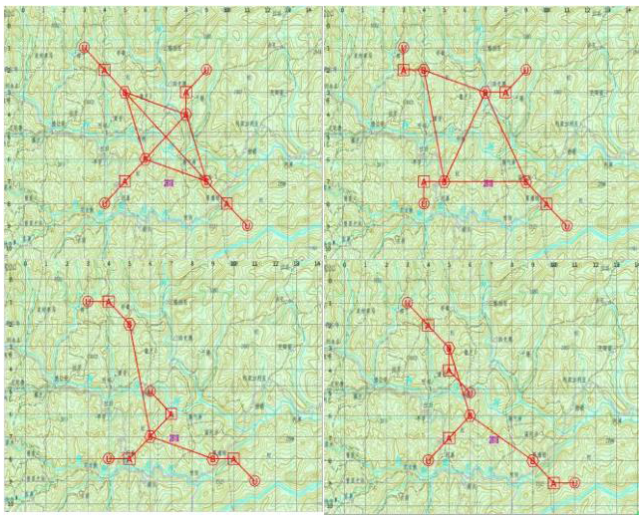


Fig. 6 Planning results in different initial situations.

4.2 Comparison with Heuristic Algorithm

In order to ensure the flexibility and effectiveness of the planning, in the planning implementation phase, multiple sets of feasible planning schemes are generated for the commander to select, and the results are planned for the 10×10 scale. As shown in Fig. 6, The generated network plan meets the requirements. In order to compare with the traditional heuristic algorithm, this paper selects three typical heuristic algorithms, genetic algorithm (GAs) [3], Tabu search (TS) [4] and simulated annealing (SA) [5], to compare with the algorithm proposed in this paper. For three different geographic scales, the comparison of the time of the four algorithms is shown in Table 1.

In two scenarios with different geographic scales, the planning time of the algorithm in this paper is far less than the other three heuristic algorithms. This is because traditional heuristic algorithms are based on a certain initial solution, and the quality of the initial solution has a great impact on the efficiency of the algorithm. In the problem of emergency communication network planning in this paper, the solution is feasible only when the number of backbone

Table 1 Planning time for different algorithms.

scale	1 set of plans				2 set of plans			
	DRL	GAs	TS	SA	DRL	GAs	TS	SA
6×6	12.4	426	878	692	16.9	426	878	692
8×8	19	716	923	848	24.1	716	923	849
10×10	22.6	987	1214	1169	28.9	987	1217	1170

nodes and access nodes does not exceed a given number and the network constraints 2–4 in 2.2 are satisfied at the same time. Considering the actual economic benefits, the pros and cons of the program are evaluated according to the number of deployed nodes. Under the condition of no human intervention, the solution space is very large, and there are few feasible solutions. Therefore, if there is no human experience, the randomly generated initial solution is basically an infeasible solution. At the same time, the heuristic algorithm is an online model. When the position and number of users change, the model needs to be redesigned and calculated. And the deep reinforcement learning algorithm proposed in this paper is an offline model. For emergency communication network planning with characteristics such as suddenness and uncertainty, the algorithm of this paper is more time-sensitive and adaptable.

5. Conclusion

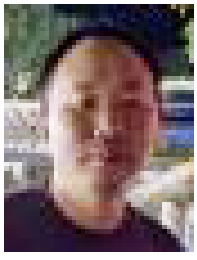
In order to carry out emergency communication network planning more efficiently and intelligently, this paper proposes an emergency communication network topology planning method based on deep reinforcement learning. The mathematical model of emergency communication network planning is constructed. The method of MCTS and deep reinforcement learning is used to design the residual network module based on convolution. The planning data generated from the self-game are used as the input of the neural network, and the intelligent planning model is obtained through the training and learning of the neural network. The experimental results show that the proposed planning method can effectively solve the communication network topology planning problem, and does not depend on historical planning data and human intervention, and has high autonomy and flexibility. The next step will be to study the efficiency of network intelligent planning algorithms in larger planning spaces and higher precision, and further improve the effectiveness and applicability of the algorithm.

References

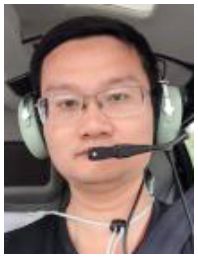
[1] F. Chiti, R. Fantacci, L. Maccari, D. Marabissi, and D. Tarchi, "A broadband wireless communication system for emergency management," IEEE Wireless Commun., vol.15, no.3, pp.8–14, 2008.
[2] M. Abd-El-Barr, "Topological network design: A survey," J. Netw. Comput. Appl., vol.32, no.3, pp.501–509, 2009.
[3] M. Abd-El-Barr, A. Zakir, S.M. Sait, and A. Almulhem, "Reliability and fault tolerance based topological optimization of computer networks — Part II: Iterative techniques," IEEE Pacific Rim Conference, pp.736–739, Victoria, BC, Canada, Aug. 2003.
[4] L. He and N. Mort, "Hybrid genetic algorithms for telecommunications network back-up routing," BT Technol. J., vol.18, no.4, pp.42–50, 2000.
[5] V. Grout, S. Cunningham, and R. Picking, "Practical large-scale network design with variable costs for links and switches," Int. J. Comput. Sci. Netw. Secur., vol.7, no.7, pp.113–125, 2007.
[6] D.N. Le, N.G. Nguyen, N.H. Dinh, N.D. Le, and V.T. Le, "Optimizing gateway placement in wireless mesh networks based on ACO algorithm," Int. J. Comput. Commun. Eng., vol.2, no.2, pp.143–147, 2013.
[7] A. Kamar, S.J. Nawaz, M. Patwary, M. Abdel-Maguid, and S.-U.-R. Qureshi, "Optimized algorithm for cellular network planning based on terrain and demand analysis," Proc. International Conference on Computer Technologies and Development, pp.359–364, 2010.
[8] Y. Zhou, "Research on node deployment and topology optimization strategy in FSO-based 5G backhaul networks," Beijing University of Posts and Telecommunications, 2019.
[9] W. Wu, "Research on topology planning for multi-interface multi-channel wireless mesh networks," Southeast University, 2013.
[10] Z.H. Zhou, Machine Learning, Tsinghua University Press, Beijing, 2016.
[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol.521, no.7553, pp.436–444, 2015.
[12] J. Ferret, R. Marinier, M. Geist, and O. Pietquin, "Credit assignment as a proxy for transfer in reinforcement learning," [EB/OL]. [2019-7-18]. https://arxiv.org/abs/1907.08027v1
[13] M. Jaderberg, W.M. Czarnecki, I. Dunning, L. Marris, G. Lever, A.G. Castañeda, C. Beattie, N.C. Rabinowitz, A.S. Morcos, A. Ruderman, N. Sonnerat, T. Green, L. Deason, J.Z. Leibo, D. Silver, D. Hassabis, K. Kavukcuoglu, and T. Graepel, "Human-level performance in 3D multiplayer games with population-based reinforcement learning," Science, vol.364, no.6443, pp.859–865, 2019.
[14] R.S. Sutton and A.G. Barto, Reinforcement Learning: An Introduction, Massachusetts Institute of Technology Press, Cambridge, USA, 1998.
[15] H.H. Van, A. Guez, and D. Silver, "Deep reinforcement learning with double Q learning," Proc. AAAI Conference on Artificial Intelligence, pp.2094–2100, 2016.
[16] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G.V. Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," Nature, vol.550, no.7676, pp.354–391, 2017.
[17] K. Shao, Y. Zhu, and D. Zhao, "StarCraft micromanagement with reinforcement learning and curriculum transfer learning," IEEE Trans. Emerg. Topics Comput. Intell., vol.3, no.1, pp.73–84, 2019.
[18] C. Clark and A.J. Storkey, "Training deep convolutional neural networks to play go," Proc. 32nd International Conference on International Conference on Machine Learning, vol.37, pp.1766–1774, 2015.
[19] S.Q. Liu, G. Lever, J. Merel, S. Tunyasuvunakool, N. Heess, and T. Graepel, "Emergent coordination through competition," [EB/OL]. [2019-2-21]. https://arxiv.org/abs/1902.07151
[20] M. Fortunato, M. Tan, R. Faulkner, et al., "Generalization of reinforcement learners with working and episodic memory," Proc. Advances in Neural Information Processing Systems, pp.12448–12457, 2019.
[21] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G.V. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," Nature, vol.529, no.7587, pp.484–489, 2016.
[22] X.B. Peng, G. Berseth, K. Yin, and M. Van De Panne, "DeepLoco: Dynamic locomotion skills using hierarchical deep reinforcement learning," ACM Trans. Graph., vol.36, no.4, pp.1–13, 2017.

[23] B. Scherrer, M. Ghavamzadeh, V. Gabillon, et al., "Approximate muddied policy iteration and its application to the game of tetris," J. Machine Learning Research, vol.16, no.1, pp.1629–1676, 2015.
[24] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," Nature, vol.518, no.7540, pp.529–533, 2015.
[25] V. Mnih, A.P. Badia, M. Mirza, et al., "Asynchronous methods for deep reinforcement learning," Proc. 33rd International Conference on Machine Learning, vol.48, pp.1928–1937, 2016.
[26] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," Proc. Advances in Neural Information Processing Systems (NIPS), pp.1097–1105, 2012.
[27] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricaed Boltzmann machine for collaborative filtering," Proc. ACM International Conference Proceeding Series, pp.791–798, 2007.
[28] D. Silver and T. Hubert, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," Science, vol.362, no.6419, pp.1140–1144, 2018.
[29] M. Jaderberg, W.M. Czarnecki, I. Dunning, L. Marris, G. Lever, A.G. Castañeda, C. Beattie, N.C. Rabinowitz, A.S. Morcos, A. Ruderman, N. Sonnerat, T. Green, L. Deason, J.Z. Leibo, D. Silver, D. Hassabis, K. Kavukcuoglu, and T. Graepel, "Human-level performance in 3D multiplayer games with population-based reinforcement learning," Science, vol.364, no.6443, pp.859–865, 2019.
[30] B. Wu, Q. Fu, J. Liang, P. Qu, X. Li, L. Wang, W. Liu, W. Yang, and Y. Liu, "Hierarchical macro strategy model for MOBA game AI," [EB/OL]. [2018-12-19]. https://arxiv.org/abs/1812.07887v1
[31] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks, vol.61, pp.85–117, 2015.

Changsheng Yin received master degree in Information and Communication Engineering from National University of Defense Technology in 2013. He now with National University of Defense Technology and a Ph.D. student.



Ruopeng Yang is now with National University of Defense Technology and a Professor, Ph.D. supervisor.



Wei Zhu is now with National University of Defense Technology and Associate Professor, Master Tutor.

Xiaofei Zou is now with National University of Defense Technology and a PhD student.

Junda Zhang received Ph.D. degree in Information and Communication Engineering from National University of Defense Technology in 2018. He is now with National University of Defense Technology as an instructor.