Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 4361650, 13 pages
https://doi.org/10.1155/2021/4361650

Research Article

Research on the Difficulty of Mobile Node Deployment's Self-Play in Wireless Ad Hoc Networks Based on Deep Reinforcement Learning

Huitao Wang,[1] **Ruopeng Yang**,[1] **Changsheng Yin**,[1] **Xiaofei Zou**,[1] **and Xuefeng Wang**[2]

[1]*College of Information and Communication, National University of Defense Technology, No. 45 JieFang Park Road, Wuhan Hubei, China*
[2]*College of Army Logistics, No. 20 North 1st Road, University Town, Shapingba District, Chongqing, China*

Correspondence should be addressed to Huitao Wang; wangjane@163.com

Received 16 January 2020; Revised 1 February 2021; Accepted 24 February 2021; Published 9 March 2021

Academic Editor: KI-IL Kim

Deep reinforcement learning is one kind of machine learning algorithms which uses the maximum cumulative reward to learn the optimal strategy. The difficulty is how to ensure the fast convergence of the model and generate a large number of sample data to promote the model optimization. Using the deep reinforcement learning framework of the AlphaZero algorithm, the deployment problem of wireless nodes in wireless ad hoc networks is equivalent to the game of Go. A deployment model of mobile nodes in wireless ad hoc networks based on the AlphaZero algorithm is designed. Because the application scenario of wireless ad hoc network does not have the characteristics of chessboard symmetry and invariability, it cannot expand the data sample set by rotating and changing the chessboard orientation. The strategy of dynamic updating learning rate and the method of selecting the latest model to generate sample data are used to solve the problem of fast model convergence.

1. Introduction

With the rapid development of artificial intelligence technology, intelligence has become an important direction of the development of various application fields, and machine learning has made significant progress in many fields, especially the success of AlphaGo and AlphaZero technology in Go human-computer game, making machine learning method to solve traditional problems become a new way [1].

Wireless ad hoc network is to build the communication network coverage within the application scenario area according to the communication support requirements of users and application scenarios and provide users with random access communication channels. The main work is to reasonably select the deployment location of mobile nodes and the connection relationship between mobile nodes [2]. Therefore, the deployment process of wireless ad hoc mobile nodes can be analogized to the game playing process of both sides of Go game, mobile game. The nodes and users can be regarded as the black and white pieces of Go, and the grid terrain area in the application scene can be regarded as the board of Go. On this basis, we explore the application of machine learning in wireless ad hoc network and build a deep reinforcement learning model of wireless ad hoc network based on the AlphaZero algorithm, so as to realize the intelligent deployment of mobile node location. The key and difficulty of the method are to generate a large number of high-quality sample data through the continuous self-play of the model. In the formal scenario application, the best deployment probability of mobile nodes can be predicted by sampling the sample data for supervised learning. The important foundation of machine learning comes from the sample data. Compared with the sample collection of chess game data, the data accumulation of wireless ad hoc network and other application fields is less, and the amount of data available is limited. After the model algorithm is built, the focus is how to generate a large number of high-quality and type rich data samples through the model of

self-play to guide the model optimization and realize the best prediction of the deployment location and networking of mobile nodes in practical application.

2. Analysis of Traditional Algorithm Model

At present, for small and typical wireless ad hoc networks, it can be realized by mature network topology automatic planning technology; for large and heterogeneous mesh networks and wireless ad hoc networks under special application scenarios, it mainly uses network planning tool software to carry out static planning in advance and dynamically fine tune and optimize the adaptation in practical application.

2.1. Review of Traditional Algorithms. At present, the algorithm model of multiobjective heterogeneous wireless network mainly abstracts the connection relationship between mobile nodes and nodes into points and lines. Based on graph theory, it simulates, analyzes, and optimizes by setting approximate assumptions and constraints and designing various channel models, traffic models, wireless models, and link models. It mainly includes 4 categories: (a) Optimize design algorithms around network hierarchical structure, topological structure, etc., and design and improve algorithms for different types of network structures such as mesh, tree, and star. For example, for the problem of dense and dense network structures, a K-means-based algorithm is the proposed random graph topology generation algorithm and hierarchical structure topology generation algorithm; for the construction of wireless sensor network plane topology structure, a network topology optimization algorithm based on a Voronoi diagram is proposed; for multihop packet wireless network structure, the shortest route tree table based on node switching is adopted. The method updates and optimizes the network topology. Aiming at the problem of network structure loss, taking the node degree and connectivity as constraints proposes a hierarchical network topology planning algorithm. In addition, it also focuses on network node mobility and network connectivity, links one or more indicators such as reliability and network capacity, sets approximate assumptions and constraints, and analyzes and studies the network topology by constructing various channel models, traffic models, mobility models, and link models. Typical algorithms include the Minimum Spanning Tree Algorithm (MST), Shortest Path Algorithm (SPT), Delaunay Triangulation Algorithm (DT), and Voronoi Diagram Algorithm [3, 4]. (b) Construct a multiobjective optimization function that focuses on multiobjective optimization combination algorithms that focus on user communication requirements, network coverage, deployment costs, network service quality, and other specific communication requirements. For example, around network connectivity, network fault tolerance, network throughput, and other index requirements, respectively, construct a wireless mesh network topology control model based on the conflict domain; around the wireless mesh backbone network topology throughput, propose a minimum spanning tree and conflict load joint topology control algorithm; propose a network topology planning method based on probability statistics based on indicators such as network coverage and network connectivity; based on network index systems and weights, propose a network topology planning method based on performance and effectiveness evaluation [5]. (c) Optimize artificial intelligence methods and strategies, focusing on solving multiobjective heterogeneous wireless network planning problems to build algorithm models, mainly including optimized search space algorithms, random search algorithms, intelligent algorithms, and improvements and combination algorithms based on the above algorithm models. For example, based on the heuristic search algorithm model, the network nodes and links are designed with the goals of optimizing the path loss in the wireless link and optimizing node deployment; abstract the node deployment problem in the network topology as the K center point problem in geometric mathematics. Optimize the links between nodes by constructing an improved particle swarm algorithm model; optimize network connectivity and network coverage by constructing simulated annealing algorithm models and genetic algorithm models; optimize wireless mesh network nodes by constructing an ant colony algorithm model. Optimize deployment; use the tabu search algorithm model to design a global optimization combination strategy and taboo table for the deployment of wireless mesh network nodes to achieve the global optimization of the network topology. Through a greedy algorithm, simulated annealing algorithm, tabu table algorithm, heuristic search algorithm, ant colony algorithm, particle swarm optimization algorithm, genetic algorithm, etc., there is intelligent algorithm optimization, improvement, and reorganization [6]. (d) Draw lessons from the concepts of complex networks, super networks, and fields in physics, and explore and study cross-domain cross-combination algorithms around the importance of network nodes and edges, information, and interaction between nodes. For example, the concepts of field and hypergraph in physics are introduced into complex networks and hypernetworks, and the network topology is optimized through the process of information interaction between nodes. In addition, in the application of artificial intelligence methods, it is proposed to use deep learning to intelligently plan wireless ad hoc network topology [7, 8].

2.2. Shortage of Traditional Algorithm. The static preplanning method is often used to optimize the topology design algorithm. The model is set for the fixed scene; the index system is greatly constrained by the conditions and cannot be adjusted flexibly and dynamically according to the scene changes effectively. (a) The accuracy of the network cannot be quantitatively evaluated, especially for large-scale network applications. The dynamic adjustment adaptability of the model is not enough, and the planning time is too long. Based on the multiobjective combination modeling method, restricted by the constraints, it can only be implemented according to different communication means or groups. The number of indicators that can be concerned is limited, and it cannot give full consideration to all indicator systems. (b) The accuracy of networking cannot be quantitatively evaluated. In the index optimization decision-making, it is easy to lose one or the other, and the universality is not high. Based on genetic algorithm, artificial bee colony algorithm,

particle swarm algorithm, and other artificial intelligence network topology optimization algorithms are mostly aimed at the convergence speed, accuracy, and robustness of the model, which are commonly used in simulation analysis and verification, and the real transformation application is still relatively small. (c) The research of network topology based on complex network theory cannot fully solve the problem of multinetwork interaction in wireless ad hoc networks and cannot fully represent the characteristics of network structure and effectively reflect the function of node type.

3. Model of Algorithm

Based on the analogy between wireless ad hoc network and board games such as Go [9, 10], and referring to the AlphaZero algorithm framework, a deep reinforcement learning algorithm for wireless ad hoc network in typical application scenarios with full visibility is constructed [11].

3.1. Algorithm Principle. According to the principle of the AlphaZero algorithm [12], a deep neural network model $(p, v) = f_\theta(s)$ with parameters θ and the evaluation index system of network system effect are constructed $z \in \{-1, 1\}$. Taking the user location and the location state distribution of mobile nodes as the input s, the deployment location probability p of mobile nodes and the evaluation v of network system effect of wireless ad hoc network are output. As shown in formula (1), MCTS search is used for heuristic search and optimization through the interaction of neural network and MCTS: on the one hand, MCTS is guided to perform heuristic search according to the maximum deployment location probability predicted by a neural network; on the other hand, the maximum location probability predicted by MCTS search reacts on the weight update of neural network and forecasts the current again [13]. The next best deployment location of mobile nodes in grid map is to maximize the similarity between the prediction probability of optional location neural network and the search probability of MCTS and to minimize the difference between the network deployment effect v_t and the deployment success z_t [14].

$$l = (z - v)^2 - \pi^2 \log p + c\|\theta\|^2. \tag{1}$$

Specifically, according to formula (2), the gradient descent method is used to update the parameters θ of the neural network in every t iterations. According to formula (3), the weights of the neural network are updated and the sample data set is optimized (s, p, v). In the formal deployment, the weights of the neural network are optimized by the sample set to guide the prediction of the maximum location probability of mobile nodes, and the wireless ad hoc network with high quality and satisfying requirements is gradually generated. Figure 1 is shown the mathematical model of deploying mobile nodes.

$$\Delta\rho \propto \frac{\partial v_\theta(s)}{\partial\theta}(z - v_\theta(s)), \tag{2}$$

$$\Delta\theta \propto \frac{\partial \log p_\rho(a_t \mid s_t)}{\partial\theta} z_t. \tag{3}$$

3.2. Structure of the Algorithm's Model. The deep reinforcement learning model focuses on designing from key submodels such as input object feature extraction, heuristic search "exploration-balance" mechanism, and reinforcement learning iterative feedback mechanism. On this basis, clarify the model training process to ensure the effective integration of various functional modules and the smooth flow of the algorithm process.

3.2.1. Input. Compared with the neural network structure in the AlphaZero algorithm, the neural network structure of the wireless ad hoc network model under the condition of full visibility takes the binary feature plane as the input, where the optional position is represented by 0, and the nonoptional position or occupied position is represented by 1. The number of planes is reduced to 3: the first plane represents the current position of the user, the second plane is the current deployed position of the mobile node, and the third plane is the constraint condition.

3.2.2. Neural Network Structure. Due to the board space of Go 19×19, the number of layers of AlphaZero's neural network reaches 40~80. Considering the computing power of a personal computer, the structure of neural network in this study is simplified compared with that in the Go algorithm. As shown in Figure 2, the neural network is designed as 5~6 layers, and the structure is basically the same as that of the AlphaZero algorithm.

3.2.3. Output. There are two output terminals: the value output terminal generates the maximum probability of mobile node deployment location through Softmax function, and the policy output terminal generates the evaluation value p of wireless ad hoc network effect through Tanh function v.

3.2.4. Model Convergence Mechanism. The function loss entropy maximizes the similarity between the deployment location probability p of the communication guarantee unit and the output probability π in the MCTS and minimizes the difference between the expected value v of the neural network in the successful network topology planning and the MCTS evaluation value. The specific method is as follows: (a) In the process of continuous trial and error and interaction in reinforcement learning, the optimal choice of the deployment position of the communication guarantee unit is optimized to obtain the largest cumulative reward. (b) Generate the maximum probability π of the deployment location of the communication guarantee unit and the evaluation value z of the network topology planning through the MCTS heuristic search algorithm, and act on the neural network to update the parameters. (c) Use the predicted maximum probability p generated by the neural network to guide the MCTS to select the location of the communication guarantee unit with the highest probability for deployment and to collect sample data for the network topology planning scheme that meets the conditions. (d) Construct a sample database based on MCTS

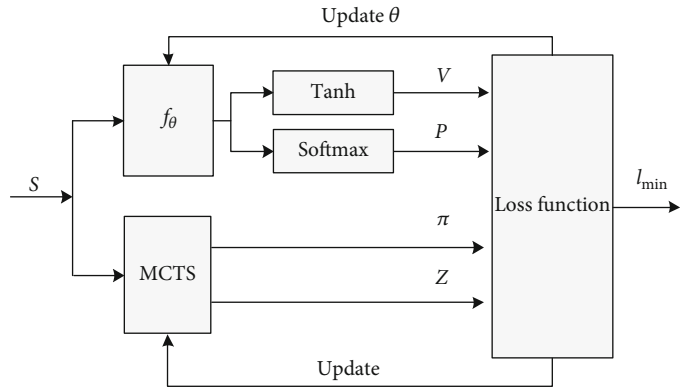

Figure 1: The mathematical model of deploying mobile nodes.

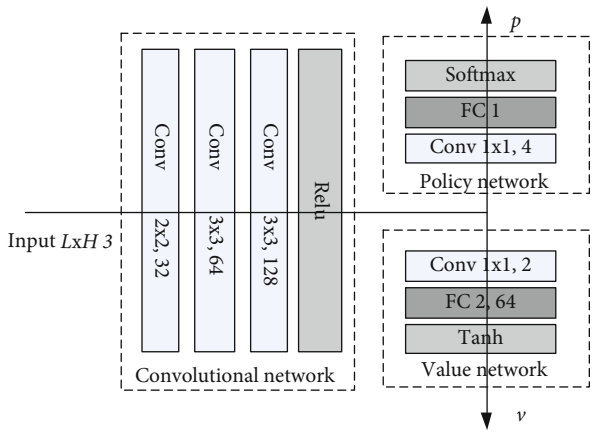

Figure 2: Structure of deep reinforcement learning model for wireless ad hoc networks.

sample data, and update the neural network parameters θ through iterative training of sample data.

3.3. Network Effect Evaluation System. The evaluation of Go and other chess is a zero-sum game of win or lose. The evaluation standard of wireless ad hoc network construction not only includes whether the network construction is successful or not but also involves the construction effect. It is necessary to turn the non-zero-sum game evaluation system constraint into a zero-sum game. As a preliminary exploration of intelligent wireless ad hoc network, this study only takes whether the network system can meet the minimum communication of users in the application scenario as the evaluation standard, that is, the communication distance of mobile node workshop must meet $D_{\min} \le d_{ij} \le D_{\max}$, among which are the minimum communication distance $D_{\min}$ and the maximum communication $D_{\max}$ distance that can ensure the communication quality and efficiency of wireless node workshop. For redundant channels, deployment costs, and so on, conditions can be set [15].

4. Analysis of Difficulties in Self-Play Training

The key point of self-play is to update the weights of neural network and generate the optimal data through continuous iteration. The key and difficult problems in the process of model training and optimization mainly include three aspects: first, how to optimize the model weight to ensure that the model can converge quickly and achieve effective solution; second, how to ensure that the model can be updated and optimized continuously and prevent overfitting in the process of model optimization; third, how to ensure that a large number of data samples with good quality and various types can be collected [16].

Considering the computing power of a personal computer, the analytic test method of difficult problems in this study is based on the size of 16×16 grid space, and the number of model self-play is set to 1500.

4.1. Parameter Optimization Strategy of Neural Network Based on Dynamic Learning Rate Updating

4.1.1. Setting Learning Rate. In the AlphaZero algorithm, the weight of neural network is updated by the method of random gradient descent in the process of self-play, so that the weight of the model is updated continuously and robust. The learning rate in the random gradient descent directly determines the performance of the model algorithm: the setting of the learning rate is too small, which is easy to cause the model to fall into the local minimum value and affect the convergence speed of the model, resulting in the overfitting phenomenon of the model; the setting of the learning rate is too large, which is easy to cause the model to oscillate near the extreme value, which makes the model unable to converge. Therefore, choosing the appropriate learning rate is the key to ensure the weight optimization and convergence of wireless ad hoc networks.

4.1.2. Method of Update Learning Rate. Learning rate updating methods mainly include the step-by-step reduction method, exponential decay method, and reciprocal decay method. The three methods have specific application scenarios in the field of machine learning [17]. The setting of learning rate of the AlphaZero algorithm adopts the step-by-step reduction method and sets the learning rate step by step according to the increase of training times [1]. Because the number of layers and structure of the neural network in this study is less than that of Go, the method of setting learning

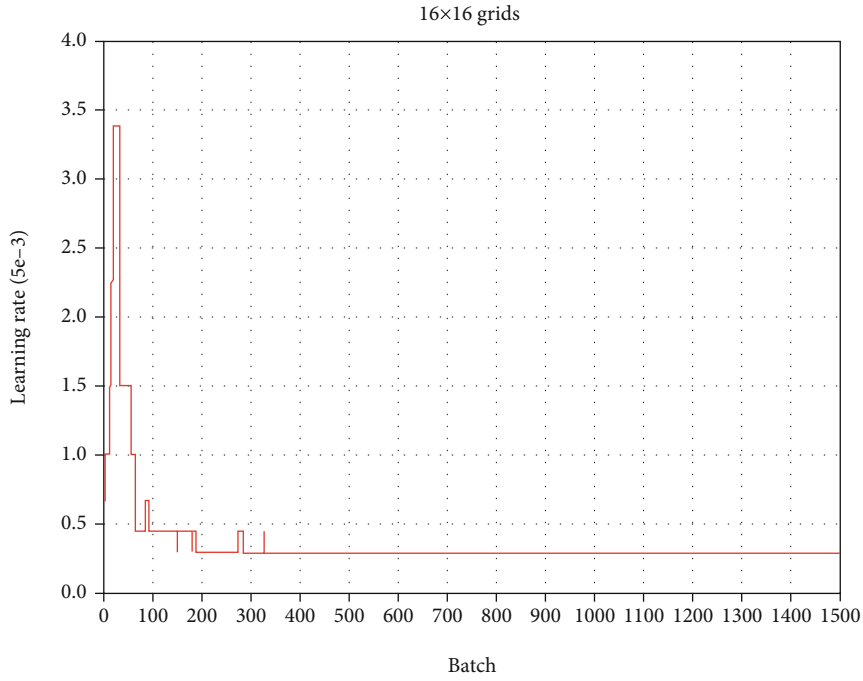

Figure 3: The change of model learning rate.

rate by the AlphaZero algorithm cannot be fully applied in the algorithm model of wireless ad hoc network. Through observation in the first 24 hours of the pretest phase, it is found that the model training has no effective results, and the model convergence is slow. On the basis of adopting the gradual reduction method and setting the initial learning rate to 0.005, increase the weighted value, and set the conditions for changing the weighted value to dynamically adjust the learning rate, as shown in Figure 3.

4.1.3. Method of Model Update. As the Go is a game between black and white chess players, the method to test the weight update of the model neural network is to generate a player according to the current latest model and the historical model, respectively, and judge the outcome of the game between the two parties. If the current latest model wins, the judgment model is updated; otherwise, the judgment model is not updated [17]. However, in wireless ad hoc networks, due to the lack of symmetry between mobile nodes and users, the above-mentioned Go evaluation method cannot be used to determine whether the model is updated. In this study, we consider designing an independent MCTS model as a third party. The players generated by the independent MCTS model do not make any changes except to increase the search depth when they fail in the game. After every 50 sampling training, the players of the model and independent MCTS model will deploy 10 mobile nodes, respectively. By comparing the success/failure ratio of the deployment, we can evaluate whether the model becomes better. If the success/failure ratio of the current sampling model is greater than that of the MCTS model, then the current model is considered to be better. At the same time, the number of searching steps of the MCTS model is increased by 1000. In the next 50 sampling training, the above steps will be continued for evaluation. As shown in Figure 4, 30 comparison results of 1500 local samples in this study can be judged that the model has been optimized 10 times.

4.2. Sample Data Generation Strategy Based on the Optimal Model

4.2.1. Sample Data Generation Analysis of the AlphaZero Algorithm. The model self-play process of wireless ad hoc network mobile nodes is to generate sample data through the model and optimize the model according to the data. The process of model optimization and sample data quality optimization is complementary and closely related. Therefore, on the basis of ensuring the convergence of the model algorithm, ensuring the model optimization can ensure the gradual improvement of the quality of data samples, focusing on the generation of data sample selection.

In the Go game, the AlphaGo algorithm and AlphaZero algorithm adopt two methods to generate sample data: the historical optimal model and the latest model. The method of generating sample data from the historical optimal model needs to add the test procedure of model updating and optimization in the process of model training [18]. For the reinforcement learning process of model optimization with the continuous iterative method, adding the test procedure at the same time will occupy part of computer resources and affect computer computing power and computing efficiency. Using the latest model to generate sample data cannot effectively ensure that the data sample data are generated by the optimal model [19].

4.2.2. Selection of Sample Data Generation. In this study, the sample data is generated from the latest model, and the model optimization inspection cycle is set at the same time.

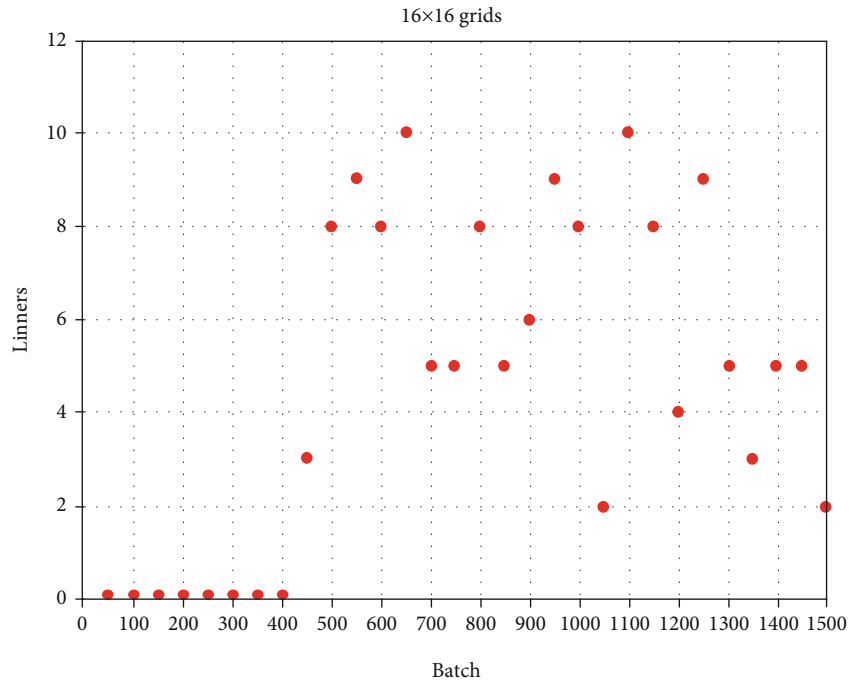

Figure 4: The game winning and losing of the current model and independent MCTS model.

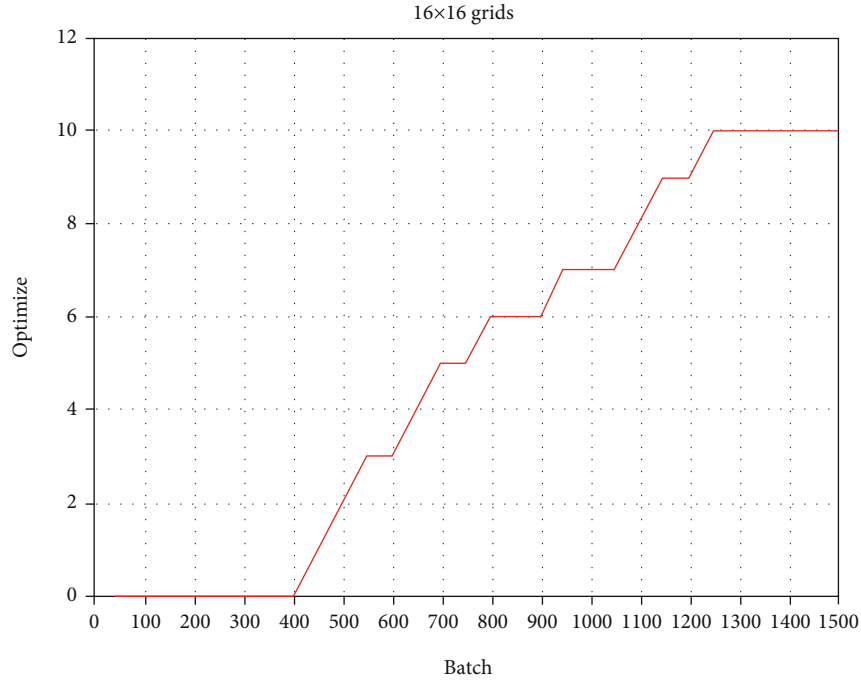

Figure 5: The result of model optimization.

After every 50 samplings, the model optimization is judged. The main considerations are as follows: first, according to the principle analysis of model update in the AlphaZero algorithm, the latest model is generally not worse than the historical optimal model, and the quality of sample data generation is guaranteed. Second, because of the high frequency of update and change of the latest model, the generated sample data is relatively independent, which can improve the coverage of sample data and improve the diversity of sample data and the speed of model convergence. The third is to check the optimization of the model on a regular basis. Instead of comparing the current latest model with the current optimal model every time, it can save training resources and time cost. Figure 5 shows the optimization of the model.

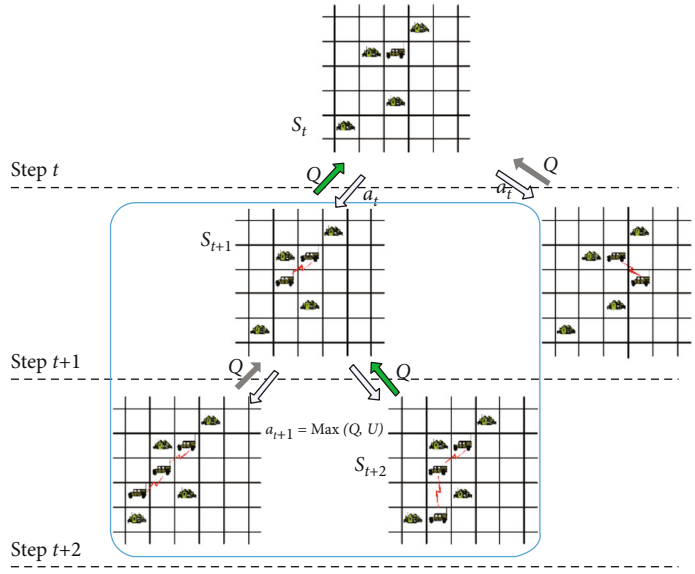

Figure 6: Schematic diagram of the realization process of "exploration-utilization" heuristic search.

4.3. Strategies to Improve the Quality of Sample Data Based on Heuristic Search

4.3.1. Sample Data Expansion Method. At present, the commonly used methods of expanding data set in machine learning mainly include the following: first, collecting more data from the data source; second, resampling the data to obtain more data; third, adopting technical processing to the original data, such as adding random noise to expand the data; fourth, generating new data artificially according to the distribution of data set, such as the AlphaZero series algorithm. According to the symmetry and invariability of Go board, the data sample set is expanded by rotating and changing the board orientation [1, 20].

4.3.2. Sample Data Expansion Analysis. Compared with the AlphaZero algorithm, there are two problems in sample data generation of mobile nodes in wireless ad hoc network: first, the terrain characteristics of the application scene of wireless ad hoc network do not have the image and turning characteristics of Go board, so the number of samples cannot be expanded through board turning. Second, compared with the attributes of Go black and white chess pieces, the attributes of mobile nodes and users in wireless ad hoc networks are different. Therefore, it is not like the game of Go, no matter whether it is black or white; as long as the current situation of chess players winning, the sample data can be collected. The deployment of mobile nodes in wireless ad hoc networks can only collect the sample data of successful deployment of current mobile nodes.

The method of mobile node deployment in wireless ad hoc network is to optimize the MTCS search method by adjusting to maximize the balance of search width and depth. The data sample collection strategy mainly includes three aspects:

(a) Set the appropriate search depth. The search depth in the example is designed as 400 steps to ensure the diversity of deployment location selection of mobile nodes in the current state

(b) Remove some positions that cannot be deployed at all, such as water surface, large obstacles, low-lying, and other positions, and reduce MTCS search space by eliminating these positions that cannot be deployed [21]

(c) Adding noise. By adding noise (Dirichlet noise) [22], expand the search depth, narrow the search scope of MCTS, solve the problem of depth and breadth balance in the search process, and improve the balance of data sample distribution

The mathematical expression of the algorithm is shown in formula (4). Among them is the deployment location selection of the communication security unit at step$t$, and s is the current input state, is the expected value of successful network topology planning, is the deployment location selection probability of the communication security unit, and selects the current deployment of the communication security unit during multiple simulations, the number of times the location is counted.

$$\begin{cases} a = \arg\max\left(Q(s,a) + U(s,a)\right), \\ U(s,a) \propto \dfrac{p(s,a)}{(I + N(s,a))}, \\ W(s,a) = W(s,a) + v, \\ N(s,a) = N(s,a) + I. \end{cases} \tag{4}$$

Figure 6 shows the implementation process of the "exploration-utilization" heuristic search principle in the MCTS in the deep reinforcement learning model of network topology planning. The combination of MCTS and neural network is adopted to consider both the current best

Table 1: The structure of deep reinforcement learning neural network under different grid sizes.

Structure	6 × 6 grid	8 × 8 grid	10 × 10 grid	16 × 16 grid
Input layer	(6,6,3)	(8,8,3)	(10,10,3)	(16,16,3)
Convolutional layer 1	(6,6,32)	(8,8,32)	(10,10,32)	(16,16,32)
Convolutional layer 2	(6,6,64)	(8,8,64)	(10,10,64)	(16,16,64)
Convolutional layer 3	(6,6,128)	(8,8,128)	(10,10,128)	(16,16,128)
Convolutional layer 4	(6,6,4)	(8,8,4)	(10,10,4)	(16,16,4)
Convolutional layer 5	(6,6,2)	(8,8,2)	(10,10,2)	(16,16,2)
Output layer	(36,2)	(64,2)	(100,2)	(256,2)

deployment location selection for network topology planning and consideration, the overall planning result of the network topology. On the one hand, MCTS is used to simulate and evaluate the deployment position of the communication guarantee unit in the next state through multiple iterations. In the case of successful network topology planning and prediction, the location with the most selected times in the simulation evaluation is selected for deployment. On the other hand, MCTS selects the possible deployment locations of other communication security units in a random probability manner, so as to explore and expand the possible deployment locations of communication security units.

5. The Result of Self-Play Training

According to the analysis of the above-mentioned model training optimization method, the model training optimization process is designed to aim at the network adjustment and reconstruction needs after the network is destroyed and interfered by the enemy under the condition of complete visibility. According to the model offline self-training strategy, whether the model is optimized, converged, and resulted in model generation, carry out analysis to test whether the overall architecture of deep reinforcement learning model construction and training process design is applicable.

5.1. Model Initialization. Deep reinforcement learning model training and optimization is a complex and time-consuming process. It is unrealistic for machine learning to directly construct a model training optimization process in a complex environment, or even completely impossible to achieve. Due to limited hardware conditions, the model training optimization constructed in this paper adopts typical application scenarios and appropriately simplifies the depth of the model to verify the model construction method, training process design, and model optimization effect.

(a) Apply background settings. Different from the traditional simulation method, the goal of deep reinforcement learning training is to achieve neural network parameter tuning. The key is to ensure that the model can quickly converge through training and to collect high-quality sample data to improve the fit of the model function. The test standard is the quality feedback of the network topology structure generated by the model, and the application scenarios should be selected with the goal of being able to test the functional effects of the model

Since deep reinforcement learning model training and optimization is a step-by-step iterative optimization process, considering the model training optimization effect and the actual network topology planning, deep reinforcement learning model training optimization is based on the network topology planning under full visibility conditions as the background, and the network is defeated by the enemy. The ability of network topology reconstruction and adjustment in the case of damage and interference is tested for application scenarios. Full visibility conditions refer to the assumptions for the use of tactical Internet organizations in this article. Specifically, the terrain environment requires that the terrain undulates slowly and the terrain fluctuations are within 20 M. Microwaves and ultrashort waves can be regarded as unobstructed, complete visibility propagation and electromagnetic. The environment requires transmission loss and external interference to be within the ideal range, and the communication distance under different communication methods is within the maximum communication distance of the equipment.

(b) Neural network structure setting. According to the overall consideration of model construction in Section 2, the optimization of deep reinforcement learning model training focuses on the effective implementation of the network topology planning model function structure and training process and focuses on the detailed design of input objects, neural network structure, and output functions

According to the neural network structure framework, taking into account the model training speed and optimization effect, the site selection space of the communication guarantee unit is not easy to be too large. Table 1 shows the four grid spaces of 6×6, 8×8, 10×10, and 16×16. In the design of the neural network structure in the deep reinforcement learning model, in order to improve the efficiency of the algorithm, the input object image is converted into a binary feature plane instead.

(c) Evaluation method setting. According to the network topology evaluation ideas in this paper, the evaluation

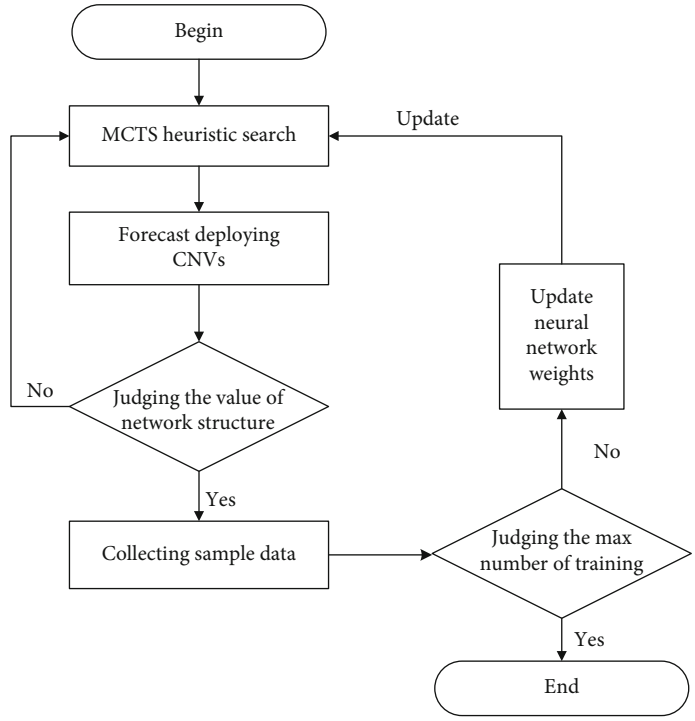

Figure 7: The deep reinforcement learning model training process in network topology planning.

method in the model training optimization is constructed according to three aspects: the connection relationship between network nodes, redundant links, and the connectivity of the whole network

Construction of the connection relationship between node pairs: in each step of the position layout of the communication guarantee unit, the shortest path is calculated according to the Dijkstra algorithm to determine whether the communication distance constraint is satisfied to determine the connectivity between the node pairs.

Whole network connectivity evaluation: according to the communication distance constraint between the communication guarantee units, under the action of the heuristic search algorithm, the connection relationship that does not meet the distance constraint is first removed and then judged according to whether the whole network constitutes a connected graph method.

Redundant link construction: on the basis of judging the connectivity of the entire network, the Prim algorithm is used to generate the minimum spanning tree of the network, and the number of edges between the minimum spanning tree and the connected graph of the entire network is judged.

5.2. Condition Setting

(a) Software and hardware support: CPU 2.3GH, Ubuntu16.04 operating system, Python 2.7 version, development and design program based on the deep reinforcement learning technology framework in AlphaZero

(b) Grid size setting: the network deep reinforcement learning model is trained under four grid spaces of 6×6, 8×8, 10×10, and 16×16. The number of training samples is 1000, and the number of simulations per step of MCTS is set to 400 times

(c) Communication distance constraint setting: in the case of 6×6 grids, the effective communication distance between type 1 users and type 2 users is 1 grid (the distance between type 1 users in this area is already within 1 to 4 grids. In the grid, communication can be achieved). In the case of 8×8, 10×10, and 16×16 grids, the effective communication distance between type 1 users and type 2 users is 1 to 2 grids, and it is effective between type 1 users. The communication distance is 1 to 4 grids

(d) Learning rate setting: through the adjustment results of the learning rate update method such as the gradual decrease method, exponential decay method, and reciprocal decay method adopted in the machine learning in the previous work, the gradual decrease method is selected as the learning rate for model training optimization. The initial learning rate is set to 0.005, and the weighting value change condition is set to realize the dynamic change of the learning rate

5.3. Model Training Optimization Analysis. Model training optimization analysis is carried out in two dimensions, horizontal and vertical, horizontal analysis of model parameters and performance changes under the same grid size, and longitudinal analysis of model performance parameter changes under different grid sizes.

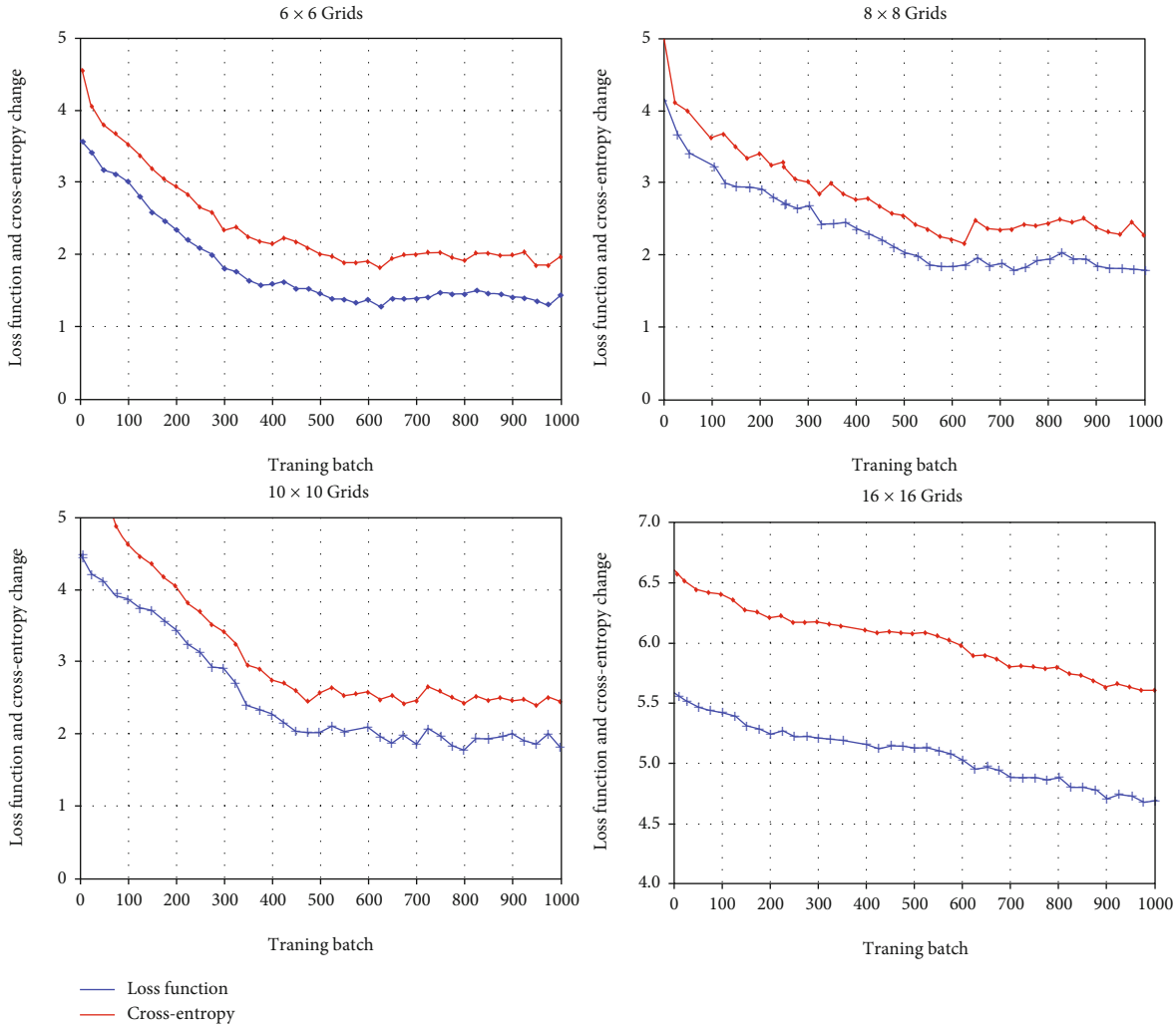

Figure 8: The changes of objective function in deep reinforcement learning model training.

According to the analysis of the model training process, it takes about 24 hours, 60 hours, 72 hours, and 310 hours to complete the training in the four grid sizes of 6×6, 8×8, 10×10, and 16×16. It is not difficult to see that as the grid size increases, the time cost of model training optimization increases significantly.

5.3.1. Construction of Training Process. The network topology planning model training optimization based on deep reinforcement learning requires predesigning the connection relationship between the output and the input of each submodule and the flow relationship between the submodules. Figure 7 is a model training process design based on deep reinforcement learning network topology planning.

(a) Neural network model process: perceive input objects through a shared neural network and perform feature extraction. According to the sample data collected by the heuristic search algorithm, the value network and policy network weights are tuned, and the estimated value of the best deployment location of the communication guarantee unit and the network topology structure in the current state is predicted

(b) Heuristic search process: starting from scratch, randomly deploy communication guarantee units in the grid of the optional rasterized topographic map through heuristic search, and collect every step of the plan when a network topology planning scheme that meets the conditions appears, the deployment location of the communication security unit. Sample the collected plan sample data and input it into the neural network, and update the neural network weights and update the optimization model after reinforcement learning iterative training

(c) Reinforcement learning iterative feedback mechanism: use the function loss entropy method to update the selection probability of the deployment location of the communication guarantee unit, and maximize the similarity between the network topology evaluation of the heuristic search algorithm and the value network output value. Within the constraints of the

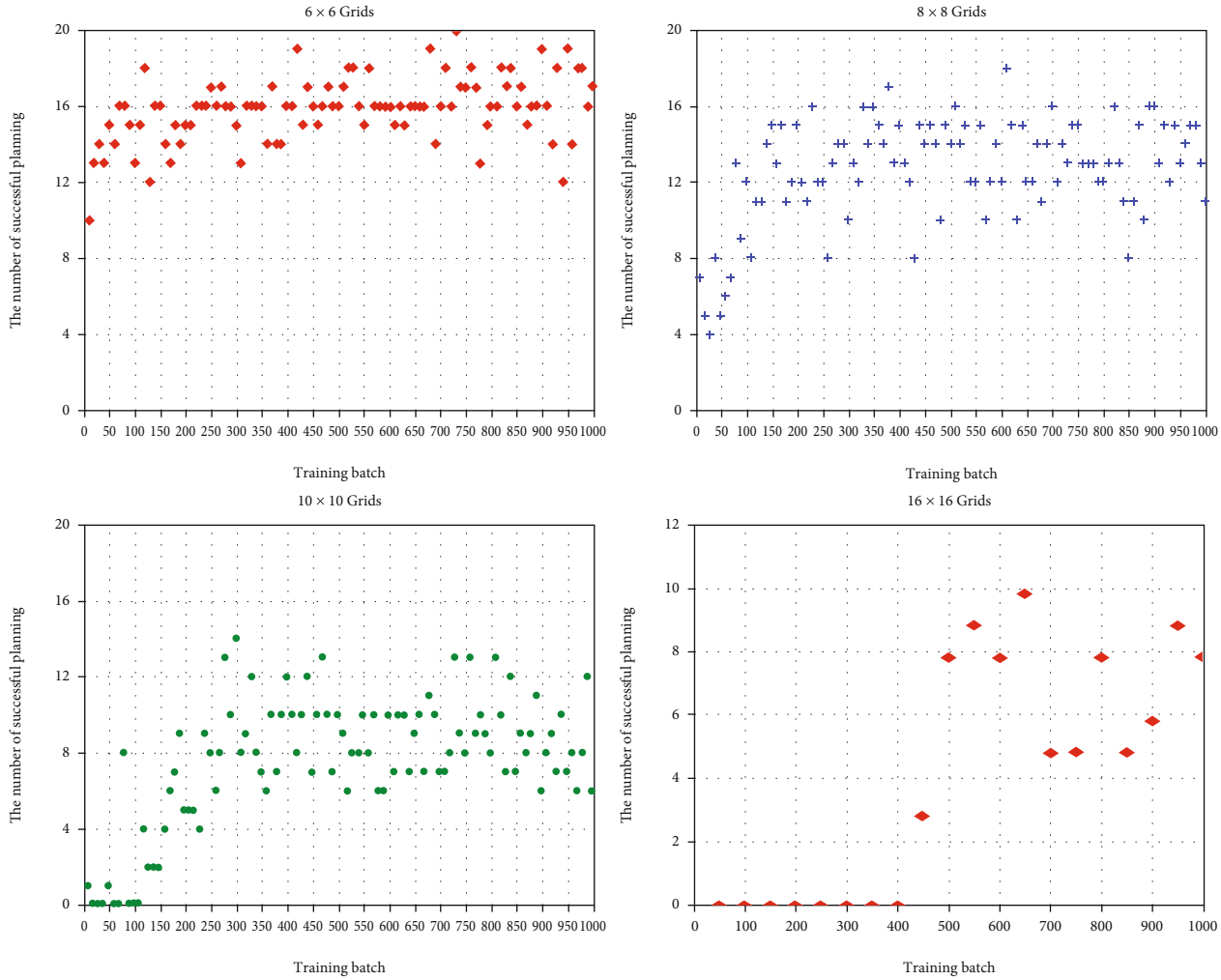

Figure 9: The distribution of successful times of network topology planning under four grid sizes.

maximum number of training times, the heuristic search algorithm process and the neural network model process are continuously repeated to ensure that the results of the strategy network and value network are gradually close to the expected value

Through the nesting and looping of the above submodules, the weights of the neural network are updated and iterated step by step, so as to continuously improve the accuracy and scientificity of the neural network function fit and prediction results.

5.3.2. Model Training. The change of model loss function reflects the construction effect of reinforcement learning iterative feedback mechanism. The change of loss function and cross-entropy reflects the interaction of neural network and heuristic search algorithm. The change of model optimization convergence can be seen through the change of loss function and cross-entropy. Figure 8 shows the changes in the model loss function and the cross-entropy in the strategy network in the four grid sizes in the case of 1000 training samples. The specific analysis is as follows.

Overall analysis: under the four grid sizes, although the loss function and strategy network cross-entropy fluctuate in sample training batches, the overall trend of change is gradually decreasing. It shows that the model training process is designed reasonably and the method is feasible.

Loss function analysis: in 6×6, 8×8, and 10×10 grid sizes, after model sampling training is 500 times, 700 times, and 800 times, the loss function basically remains at about 2.0, 2.4, and 2.5., which shows that the basic training of the model is completed under the above three grid sizes. However, after the model has been sampled and trained 1000 times under the 16×16 grid size, the loss function still continues to change and has not stabilized. It can be judged that the model can converge quickly and show a trend of gradual improvement under small grid sizes; as the number of grids increases, model convergence and parameter tuning take longer, and the training effect is not obvious.

Strategy function cross-entropy analysis: the strategy function cross-entropy directly reflects the difference between the actual probability and the expected probability in the training of the communication guarantee unit. Similar to the change of the loss function, the cross-entropy of the function

gradually decreases and tends to stabilize under the first three grid sizes, indicating that as the sample sampling batch increases, the neural network predicts the probability of the deployment location of the communication guarantee unit from random. Approximately equal-probability gradually showed a trend of uneven distribution step by step, and the result prediction gradually showed a trend, indicating that the model gradually tends to converge, and the design of the interaction mechanism between neural network and heuristic search in network topology planning is feasible.

5.3.3. Design Aspects of Evaluation Indicators and Evaluation Methods. In this study, we adopted the method of regularly testing the effect of model network topology planning. The effect of model network topology planning was analyzed every 20 training sample batches. Analyze and evaluate the effect of network topology planning to test the effectiveness of the proposed evaluation method. Figure 9 shows that the model is tested for 20 network topology planning results after every 20 sampling training during the model training process. The specific analysis is as follows.

Evaluation index analysis: the network topology planning results under the four grid sizes meet the average network connectivity times of 17/20, 12/20, 10/20, and 6/20. It shows that the model can train the optimization model through the whole network connectivity index and realizes the improvement of the prediction accuracy probability of the communication guarantee unit site selection. It shows that the index decomposition method is adopted and the method of using the whole network connectivity index as the model evaluation index is feasible.

Evaluation method analysis: the effect of the evaluation method in machine learning cannot be directly observed, but the horizontal comparison and analysis of the network topology planning results under the four grid sizes show that the evaluation method is basically feasible and available. At the same time, as the number of grids increases, the number of times that the network topology planning result meets full connectivity gradually decreases, which indicates that the efficiency of constructing the connectivity relationship between node pairs in the evaluation method increases. When the model has a large solution space, the realization of the function requires further improvement methods.

6. Conclusion

Compared with traditional planning methods, the AI deep learning method does not solidify knowledge in an algorithm model but achieves the abstraction and understanding of knowledge through self-learning, reduces the interference of human factors, and improves scientificity. The application of artificial intelligence technology is based on large-scale sample data. In this study, by migrating the AlphaZero algorithm framework, a new mobile node deployment algorithm model under the condition of complete intervisibility is constructed as the specific practice exploration of artificial intelligence in the typical application of mobile wireless ad hoc network. The difficulty and key work are how to generate rich and comprehensive sample data through self-play, so the design of the self-play process is the core and foundation. Referring to the main methods of the AlphaZero algorithm and comparing the different points in the mapping between Go and wireless ad hoc network, this study solves the difficult problems that affect the training sample data generation, such as model optimization, data collection, and model convergence, realizes the migration application of AlphaZero technology in wireless ad hoc network under the condition of full visibility, and provides the next step for the model exploration and application in complex terrain. The basis of the research is given.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] D. Silver, T. Hubert, J. Schrittwieser et al., "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," 2017, https://arxiv.org/abs/1712.01815.

[2] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys*, vol. 37, no. 2, pp. 164–194, 2005.

[3] E. Amaldi, A. Capone, M. Cesana, I. Filippini, and F. Malucelli, "Optimization models and methods for planning wireless mesh networks," *Computer Networks the International Journal of Computer & Telecommunications Networking*, vol. 52, no. 11, pp. 2159–2171, 2008.

[4] H. Kim, E. C. Park, S. K. Noh, and S. B. Hong, "Angular MST-based topology control for multi-hop wireless ad hoc networks," *ETRI Journal*, vol. 30, no. 2, pp. 341–343, 2008.

[5] A. Noack, P. B. Bok, and S. Kruck, "Evaluating the impact of transmission power on QoS in wireless mesh networks," in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, Lahaina, HI, USA, 2011.

[6] M. E. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Physical review E*, vol. 64, no. 2, 2001.

[7] S. Sakamoto, E. Kulla, T. Oda, M. Ikeda, L. Barolli, and F. Xhafa, "A comparison study of simulated annealing and genetic algorithm for node placement problem in wireless mesh networks," *Journal of Mobile Multimedia*, vol. 9, no. 2, pp. 101–110, 2013.

[8] N. N. G. Le HD, N. H. Dinh, N. D. Le, and V. T. Le, "Optimizing gateway placement in wireless mesh networks based on ACO algorithm," *International Journal of Computer & Communication Engineering*, vol. 2, no. 2, pp. 45–53, 2013.

[9] O. E. David and N. S. Netanyah, "End-to-end deep neural network for automatic learning in chess," in *International Conference on Artificial Neural Networks*, pp. 88–96, Cham, 2016.

[10] C. Clark and A. J. Storkey, "Training deep convolutional neural networks to play Go," in *International conference on machine learning*, vol. 37, pp. 1766–1774, 2015.

[11] X. Zou, R. Yang, C. Yin, Z. Nie, and H. Wang, "Deploying tactical communication node vehicles with AlphaZero algorithm," *IET Communications*, vol. 14, 2019.

[12] D. Silver, J. Schrittwieser, K. Simonyan et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, pp. 354–359, 2017.

[13] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 2016.

[14] R. Coulom, "Efficient selectivity and backup operators in Monte-Carlo tree search," in *International conference on computers and games*, pp. 72–83, Berlin, Heidelberg, 2006.

[15] C. H. Liu, S. Y. Kuo, D. T. Lee, C. S. Lin, J. H. Weng, and S. Y. Yuan, "Obstacle-avoiding rectilinear Steiner tree construction: a Steiner-point-based algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 1050–1060, 2012.

[16] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[17] D. Silver, L. Newnham, D. Barker, S. Weller, and J. McFall, "Concurrent reinforcement learning from customer interactions," in *International conference on machine learning*, vol. 28, pp. 924–932, Atlanta, GA, USA, 2013.

[18] C. Finn, P. Christiano, P. Abbeel, and S. Levine, "A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models [EB/OL]," 2016, https://arxiv.org/abs/1611.03852.

[19] V. Mnih, A. P. Badia, M. Mirza et al., "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, vol. 48, pp. 1928–1937, New York, NY, USA, 2016.

[20] S. Loffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, vol. 37, pp. 448–456, Lille, France, 2015.

[21] D. Perez, P. Rohlfshagen, and S. M. Lucas, "Monte-Carlo tree search for the physical travelling salesman problem," in *European Conference on the Applications of Evolutionary Computation*, pp. 255–264, Berlin, Heidelberg, 2012.

[22] A. Doerr, N. D. Ratliff, J. Bohg, M. Toussaint, and S. Schaal, "Direct loss minimizaton inverse optimal control," *Molecular Ecology*, vol. 23, no. 10, pp. 2602–2618, 2015.