Fast Node Selection of Networked Radar Based on Transfer Reinforcement Learning

Yanjun Cao, Yuan Wang, Jingjing Guo, Li Han, Chao Zhang, Jin Zhu, Tianyang Zhang, Xiangrong Zhang

Fast node selection of networked radar based on transfer reinforcement learning

Yanjun Cao[1], Yuan Wang[2], Jingjing Guo[3], Li Han[1], Chao Zhang[1], Jin Zhu[1], Tianyang Zhang[2], and Xiangrong Zhang[2]

1 The 54th Research Institute of China Electronics Technology Group Corporation
2 Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xian, Shaanxi Province 710071, China
3 Xidian University, Natl Lab Radar Siganl Proc, Xian 710071, People's Republic of China

Abstract. The networked radar system can synthesize different echo signals received by various radars and realize the cooperative detection of multiple radars, becoming more and more critical for data fusion sharing and network collaboration. However, due to the large number and wide range of nodes in the networked radar system, there exists a redundancy problem in radar node assignment, which causes additional resource consumption and slows down the task execution speed of radar node selection. To solve the above problem, this paper proposes a fast radar node selection method based on transfer reinforcement learning to quickly select the optimal and minimum node resources. The proposed method devises a novel reward function for the Monte Carlo Tree and a different termination state of iteration to select the minimize the number of radar nodes. In order to further accelerate the selection of radar nodes, transfer reinforcement learning is presented to fully leverage the previous knowledge. Experimental results show that our proposed method can quickly select the optimal and minimum radar nodes in a brief period, significantly improving the speed of radar node selection in the networked radar.

Keywords: Reinforcement Learning · Transfer Learning · Monte Carlo Tree.

1 INTRODUCTION

The nodes of the networked radar system can coordinate in different dimensions such as time, space, frequency, waveform, polarization and angle of view through the central node, which can better realize data fusion sharing and network coordination. However, due to the wide range of networked radar deployment and numerous nodes, there is a lot of redundancy for nodes assigned by a single task, which seriously slows down the execution speed of the task and the timely response of the networked radar system.

Reinforcement learning has the ability to actively explore the environment, which can obtain feedback from the environment and make action decisions based

on the feedback. This ability to continuously learn in a dynamic environment has given reinforcement learning a lot of attention. Silver[13, 14] et al. learned the strategy of go game through reinforcement learning and achieved excellent results. Bello[1] et al. used reinforcement learning to train pointer networks to generate solutions for up to 100 nodes of the synthetic Travelling Salesman Problem (TSP) instances. In the networked radar system, different radar nodes may be unselectable when occupied by other tasks and the radar observation target usually moves fast, which leads to the rapid change in the search environment of the networked radar system. Therefore, reinforcement learning, as a learning method that interacts with the environment, is well suited for the radar node selection task in the networked radar system.

This paper presents a fast networked radar node selection method based on transfer[15] reinforcement learning. Specifically, we devise a novel reward function of the Monte Carlo Tree and redefine the termination state to select the least and optimal combination of nodes, which tackles the redundancy problem of single task radar node assignment. Besides, a large number of searches are required to reconstruct the Monte Carlo Tree for different tasks each time, seriously reducing the search efficiency and constraining the quick response of the networked radar systems. To further accelerate the search progress, this paper proposes transfer reinforcement learning to avoid random blind global searches, which highly reduces the node search time of networked radar system, and improves the quick response performance of networked radar system.

2 RELATED WORK

2.1 Radar node selection

In the past few decades, radar node selection of MIMO array has attracted great attention in radar applications. Berenguer *et al.*[2] proposed a fast adaptive node selection method based on discrete random optimization, which uses active random approximation iteration to generate an estimated sequence of solutions. Mendezrial selected radar nodes through a compressed spatial sampling of received signals to reduce the complexity and power consumption of millimetre-wave MIMO systems. X. Wang *et al.*. developed a series of deterministic based theory and optimization based methods for selecting radar node subsets and reconfiguring array structures to maximize the output SNR and improve the estimation of azimuth (DoA). In MIMO radar, most scholars mainly study node selection from target parameter estimation. Godrich[5] proposed an optimal radar node arrangement scheme to reduce the CRLB of aircraft speed estimates. Then he used combinatorial optimization to minimize the positioning error of multiple radar systems. Gorji[6] minimized its estimation error by calculating the target position estimation performance (CRLB) of MIMO radar node combinations. In MIMO radar sensor networks, joint antenna subset selection and optimal power distribution are realized by convex optimization.

Nevertheless, none of these methods can make feedback on changes in the environment. We use reinforcement learning to constantly interact with the en-

vironment to achieve efficient search in the rapidly changing environment of networked radar systems.

2.2 Reinforcement learning.

Reinforcement learning comprises Agent, Environment, State, Action, and Reward. After an agent acts, the environment will shift to a new state, for which the environment will give a reward signal (positive reward or negative reward). The agent then performs the new action according to specific strategies based on the reward of the new state and the environment feedback. The process described above is how an agent and the environment interact through states, actions, and rewards. Through reinforcement learning, an agent can know what state it is in and what actions it should take to get the maximum reward.

In the face of the complex networked radar system, it contains a lot of information and rules, which causes some difficulties in establishing the corresponding search environment. Monte Carlo search tree (MCTS) is an algorithm based on reinforcement learning, which does not require a given domain strategy or specific practical knowledge to make a reasonable decision. The ability to work effectively with no knowledge beyond the basic rules of selection scenarios means that MCTS can be widely used in many domains with only minor adjustments.

2.3 Transfer learning.

Transfer learning refers to acquiring knowledge from a source problem to solve a completely different but related new problem to achieve more efficient learning in a new task.

In transfer learning, there are two fundamental ideas: domain and task. The existing data is called the source domain, and the new knowledge to be learned is called the target domain. The data includes information data and model data. The specific task is to design the model to solve this problem. Generally, source domain data is migrated to solve the task of the target domain, that is, to build and design the ideal model of the target domain. This paper divides the transfer learning paradigm into three categories: inductive transfer, direct transfer learning, and unsupervised transfer learning. Among them, inductive transfer learning requires that the target task and source task are not the same but have a correlation, and so does unsupervised transfer learning. According to the content steps of transfer learning, transfer learning can be divided into four cases: transfer based on samples, transfer based on feature representation, transfer based on parameters, and transfer based on relative knowledge. Among them, the transfer from the sample to the sample refers to the transfer of source domain information to the target domain by assigning a certain weight to the target domain. The theory of feature transfer is to extract the main elements from the information contained in the source and target domains to make their distribution consistent. Relative data-based migration refers to mining the mapping of relative knowledge in the source domain and target domain.

The application of transfer learning can solve the drawbacks of information acquisition, that is, the absence of knowledge annotation or the utilization of historical source domain information, and generate valuable data for new tasks. In addition, individual users will be on the "shoulders of giants" when applying reinforcement learning and will be able to train their own tasks by using models placed on devices with superior computing power so as to improve the generalization ability of models. At the same time, a model with a high general degree is built on the basis of the individuation of other tasks. The model with better training results can flexibly cope with completely different environments and tasks, and the end-to-end requirements can meet the practical application.

Machine learning often relies on better training results: training and test data are crucial and must be distributed in the same domain. If the two data sets are very different, machine learning must spend more time accumulating new knowledge and models. Therefore, the study of transfer learning has begun to attract the attention of many researchers. Compared with machine learning in the past, transfer learning aims to extract high-quality training data by using the knowledge left by historical tasks to guide the rapid completion of recent tasks.

Reinforcement learning algorithm takes a long time to solve in the search process and is difficult to meet the requirements of real-time decision-making in a short time. The apparent advantage of transfer reinforcement learning lies in knowledge transfer ability. That is, every solving task is no longer independent and unrelated, and initial reconstruction is no longer required after inputting new tasks, so the acquired knowledge can be better utilized.Migration learning algorithm based on a large number of existing optimization of the effective extraction of knowledge, overcome classical mathematical calculation method relies on the building of mathematical model, to further speed up the optimization calculation of the traditional heuristic algorithm, effectively avoid the random global explore blindly, to guide the individual performs high accuracy of local search, optimization and improve efficiency and quality. Finally, it can give the minimum combination of nodes satisfying the requirements in more than a second among hundreds of millions of node combinations.

3 METHODOLOGY

3.1 Revisiting of Monte Carlo Tree

MCTS uses trees to store state, action and return data of Markov Decision Process(MDP), which simulates intelligent agent to generate data and calculate expectation to obtain the optimal strategy. The node attributes of the Monte Carlo Tree Search[4] include N, which represents the number of visits of the node during the training, with an initial value of 0. R, which is the reward of the current node and the initial value is 0. The larger the reward value is, the better the performance of the radar node will be for the target detection task executed. $Parent$, which is the parent node used for back propagation to traverse the entire tree up to the root node; $Children$, which is the set of the children in

the current node; *State*, which is the node's state, each state corresponding to a node combination.

We employ Monte Carlo Tree to simulate a large number of radar node combinations and calculate their reward. The prefix tree records these node combinations and the calculation results. The final optimal node combinations are predicted by simulation results. In Section 3.2, we introduce the selection operation of our Monte Carlo Tree when simulating the generation of various combinations of radar nodes. For the original Monte Carlo tree, it can only carry out fixed radar node selection and cannot settle node redundancy. Therefore, we introduce a new reward function in Section 3.3 to realize the minimization of node numbers. In order to further accelerate the search, we introduce transfer reinforcement learning in Section 3.4.

3.2 The lower bound of Cramero ($CLRB$)

We use the lower bound of Cramero ($CLRB$) to evaluate the performance of the radar node selection scheme. The smaller CLRB is and the fewer radar nodes it contains, the better the selection of radar node combinations, resulting in the larger reward value.

$CLRB$ is the linear unbiased lower bound of the target location parameter estimation problem. We first simulate the multi-node echo signal according to the target location provided by the on-duty radar, then obtain the fisher information matrix by taking the second derivative of the target location parameters in the echo signal, and finally obtain the lower bound of the unbiased estimation corresponding to the target location. In this paper, we use the on-duty radar to provide the approximate location range of the target and select the node combination through the node selection algorithm to minimize the target's positioning error at this position and achieve accurate positioning of the target. The approximate calculation process is as follows:

1)Calculate the time delay based on the approximate target position provided by the on-duty radar.

$$\begin{aligned}\tau_{mn}\left(X_q\right) &= \frac{R\left(T_m, X_q\right) + R\left(R_n, X_q\right)}{c} \\ &= \frac{\sqrt{\left(x_m - x_q\right)^2 + \left(y_m - y_q\right)^2 + \left(z_m - z_q\right)^2} + \sqrt{\left(x_n - x_q\right)^2 + \left(y_n - y_q\right)^2 + \left(z_n - z_q\right)^2}}{c}\end{aligned} \tag{1}$$

$R\left(T_m, X_q\right)$and $R\left(R_n, X_q\right)$ are the approximate distances of the transmitting node T_m and the receiving nodeR_n to the target respectively. x_q,y_q,z_q is the approximate position of the aircraft detected by the watch radar. x_m,y_m,z_m is the coordinates of the transmitting nodeT_m.x_n,y_n,z_n is the coordinates of the receiving nodeR_n.

2)Time delay is used to calculate the echo signal of multiple nodes

$$\begin{aligned} rcp_n(t) &= G_{R_n} \sum_{m=1}^{M} \sum_{q=1}^{Q} \delta_q a_{mn} \sqrt{P_m} G_{T_m} \exp\left(-j2\pi f_c \tau_{mn}\left(X_q\right)\right) s_m\left(t-\tau_{mn}\left(X'\right)\right)+N_n(t) \\ &\approx G_{R_n} \sum_{m=1}^{M} \delta_q a_{mn} \sqrt{P_m} G_{T_m} \exp\left(-j2\pi f_c \tau_{mn}\left(X_q\right)\right) s_m\left(t-\tau_{mn}\left(X'\right)\right)+N_n(t) \end{aligned} \tag{2}$$

G_{R_n},G_{T_m} is array gain.δ_q is the reflectivity of each scattered point.P_m(m=1,2,...,M) is the transmitting power of the transmitting nodeT_m.S_m is the waveform.$N_n(t)$ is additive white Gaussian noise. a_{mn}(m=1,2,...,M,n=1,2,...,N) is the direction vector of the transmitting or receiving node. f_c is the carrier frequency.

3)Using echo signal $rcp_n(t)$, spatial coherent processing, pulse compression and Angle measurement are used to predict target position parameters $\left(x, y, z, \delta^R, \delta^I\right)$.

4)Fisher information matrix is used to calculate the error of prediction target location parameters $\left(x, y, z, \delta^R, \delta^I\right)$, and then the lower bound of Cramero is:

$$CLRB_c = \sigma^2_{x_cCRB} + \sigma^2_{y_cCRB} + \sigma^2_{z_cCRB} \tag{3}$$

Where, $\sigma^2_{x_cCRB}$, $\sigma^2_{y_cCRB}$, $\sigma^2_{z_cCRB}$ and are the errors of coordinates X,Y and Z respectively, which are obtained from time delay τ_{mn} , echo signal $rcp_n(t)$ and Fisher information matrix. The detailed calculation process is referred to [7].

3.3 Selection Flow

We start with the root node and then add new radar nodes in turn. For the node that has not reached the termination state, we randomly generate a child node of the current node as the new node or select the child node with the largest UCB value among the children nodes of the current node that have been visited as the new node. The selection stops for nodes that reach the terminal state, and an iteration ends. where, the calculation formula of confidence value UCB is as follows:

$$UCB = \frac{r_i}{n_i} + C \times \sqrt{\frac{\ln F}{n_i}} \tag{4}$$

Where n_i is the access times of the node i, r_i is the reward value of the node i, and F is the total times that the parent node of the node i has been accessed.C is an adjustable hyperparameter (that is, an artificially set constant).In particular, we count the number of children to determine whether they are not accessed, partially accessed, or fully accessed.

3.4 Variable-number node search

The reward function for the number of nodes. In reinforcement learning, the intelligent agent's goal is formalized as a particular signal, called return, which is transmitted through the environment and is calculated by the reward function. Monte Carlo Tree searches in the direction of maximizing return. In

order to simultaneously search in the direction of minimizing the number of nodes, we introduce the number of nodes as a variable into the reward function r. Its specific calculation formula is as follows:

$$r = \begin{cases} (levels - len(move) + 1)/levels & 0 < CLRB \leq 10 \\ 0 & other \end{cases} \tag{5}$$

Where $levels$ is the number of radar nodes that can observe a target, $moves$ is the combination of radar nodes, and $len(moves)$ is the number of nodes. We give the reward value referring to the number of radar nodes. The fewer the number of radar nodes, the greater the reward value.

Two cases of termination. After obtaining the reward function, we redefined the termination state of Monte Carlo Tree. The termination state is the symbol of the end of tree search, indicating that the node combination has met the requirements and there is no need to continue the search to add new radar nodes layer-by-layer. Since our goal is to find the minimum node combinations that satisfy the requirements, we don't have to search all the way to the leaf node. We set up two cases to reach the termination state, ensuring that each iteration outputs the minimum combination of nodes. One is to reach the leaf node of the tree (including all radar nodes); In another case, these selected radar nodes have met the performance requirements.

3.5 Transfer reinforcement learning

Transfer reinforcement learning. Transfer learning refers to acquiring knowledge from the source problem to solve another task or a new problem with different but related environment in order to achieve more efficient learning in the new task. The application of transfer learning can solve the drawbacks of information acquisition, that is, the absence of knowledge annotation or the utilization of historical source domain information, and generate valuable data for new tasks. Transfer learning and reinforcement learning are combined to extract high quality training data by using the knowledge left by historical tasks to guide the rapid completion of tasks in different environments.

The transfer reinforcement learning used in this paper achieves the fixed domain migration of the target task across multiple source tasks[8]. In this case, different source tasks share the same domain, and the migration algorithm will take the knowledge gathered from a set of source tasks as input and use it to improve performance in the target task[9]. The RL algorithm has a large number of parameters that define initialization and algorithm behavior[11]. Some migration methods change and adjust algorithm parameters based on the source task[16]. For example, if the action values in some state-action pairs are very similar across the source task, then the node parameters of the Monte Carlo tree of the target task can be initialized accordingly to speed up the learning process[10]. By doing this, we can accelerate the convergence of the model.

Training and preservation. We repeatedly trained the same tree by inputting aircraft positions in different directions. And then node parameters(moves, reward, visits) are saved through breadth traversing the Monte Carlo Tree.

Rebuilding tree. As the number of nodes increases layer by layer, we start to rebuild from the root node to find the combination of nodes whose prefix is the same as the parent node of this layer, and build new tree nodes. Ultimately, a trained Monte Carlo Tree is obtained by expanding layer by layer.

Fine tuning. The search for node combination is carried out on the basis of the real-time aircraft coordinates. A bias item is introduced into the reward function to give more rewards to the combination of nodes that meet the requirements during fine-tuning to search for the direction of the optimal combination of nodes. Finally, according to the fine-tuning Monte Carlo Tree, the optimal node combination is picked out.

Table 1. Radar node selection input parameters.

parameters	value
Number of Nodes	145
K(Boltzmann's Constant)	1.38e-23
T0	290
Noise Factor	1e0.3
Radar Loss	1e0.4
RCS	12.5
SNR	100
Number of large node lattice elements	256
Number of section lattice elements	16
B(Radar bandwidth)	1e6
Angle between radar plane and Z axis	$-\pi/4$

4 EXPERIMENTS AND ANALYSIS

We employ 145 radar nodes, including five large nodes for transmitting and 140 small nodes for receiving, facing in all directions. Specific input parameters is as shown in Table.2. In the fine-tuning stage, the target's position is (667.65, 7631.29, 6427.87), which is $1e4$ m away from the origin of the coordinate system, and the number of iterations is 1000.

(1)Validation of minimum number of nodes Based on the coordinate information of our radar nodes and the target, the minimum number of nodes our algorithm searched is four. So we search 50000 times randomly in the case of fixed three nodes and calculate CLRB. Since the baseline of CLRB is 10, as shown in Fig.1, the subgraph is a local amplification of the ranges of 10, it can be seen that the CLRB of the 3-node combination cannot meet the requirement. Therefore, we can analyze that our Monte Carlo Tree can find the minimum

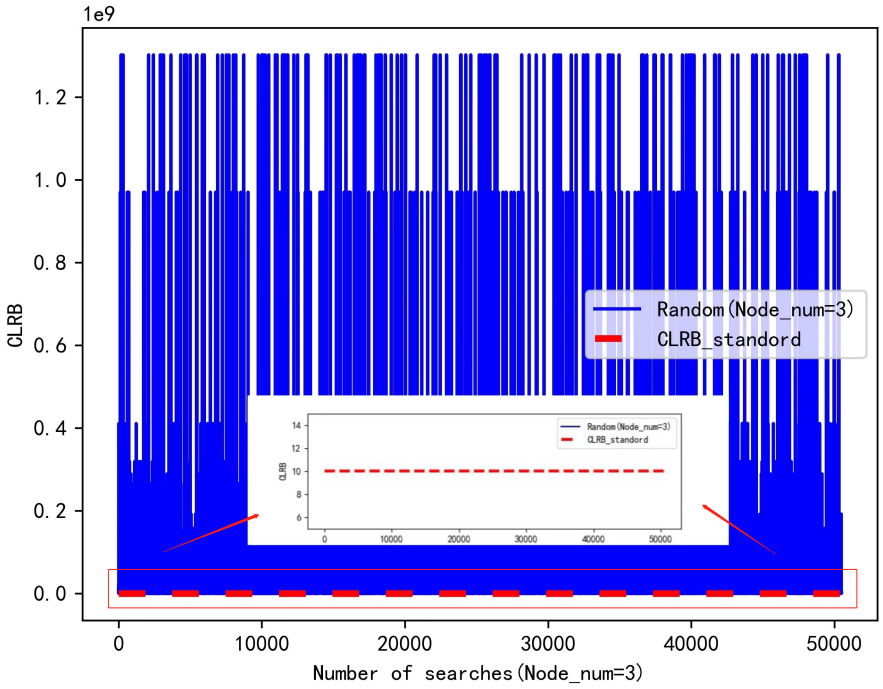

Fig. 1. 50,000 searches (Number of nodes = 3).

number of nodes that meet the requirements, which can solve the problem of node redundancy distribution to improve the execution speed of tasks.

Table 2. The average time of 50 times experiment.

Method	Time(s)
The original Monte Carlo Tree[4]	2.105
Ours	0.325
Genetic Algorithm[12]	13.209
Simulated Annealing Algorithm[3]	6.300

(2)Comparison results

After obtaining the minimum number of nodes, we employ classical search methods to conduct comparative experiments. In order to ensure that each method can search for solutions that meet the conditions, we set the minimum number of iterations, respectively, among which the original Monte Carlo Tree and the simulated annealing algorithm is 1000. Ours is 10 and the genetic algorithm is 50. In the comparison of time performance in Fig.2 and Table.2, when the number of fixed radar nodes is 4, our Monte Carlo Tree based on transfer learning is much faster. As shown in Fig.3 and Table.3, the node combination searched by our algorithm is better than that of other search algorithms in most

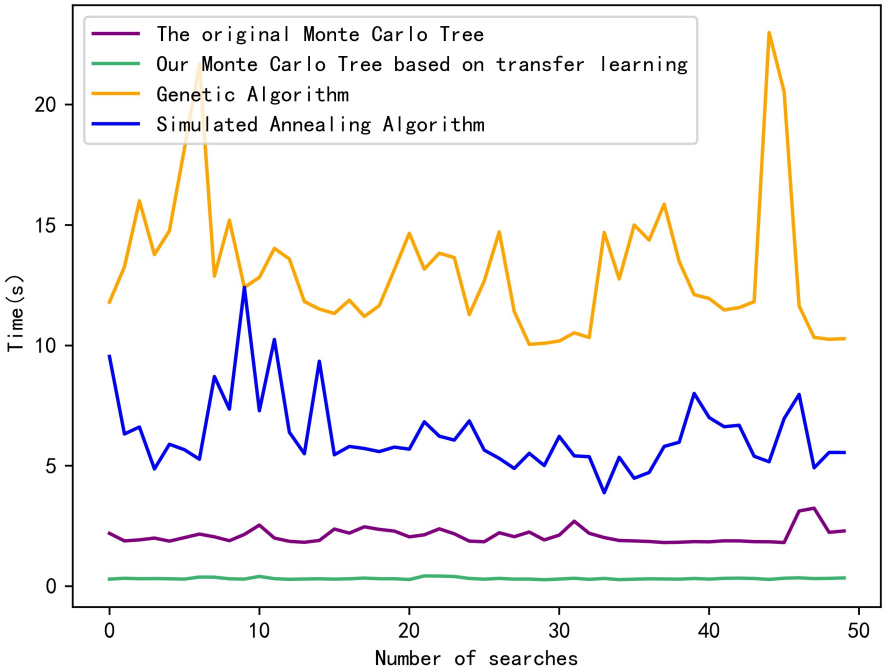

Fig. 2. Search time performance comparison(s).

Table 3. The average CLRB of 50 times experiment.

Method	CLRB(m)
The original Monte Carlo Tree[4]	2.098
Ours	1.239
Genetic Algorithm[12]	2.406
Simulated Annealing Algorithm[3]	3.025

cases. It can be seen that our algorithm is better able to find the optimal solution, which is conducive to improving the detection performance of the networked radar system.

5 CONCLUSION

In this paper, we propose a fast networked radar node selection method based on transfer reinforcement learning to tackle the redundancy problem in radar node assignment and accelerate the execution speed of radar node assignment in the networked radar system. Concretely, we devise a novel reward function and iterative termination state for the Monte Carlo tree to achieve the minimize and optimize radar nodes selection. To further speed up the radar node selection, we introduce transfer reinforcement learning to reuse the previous knowledge.

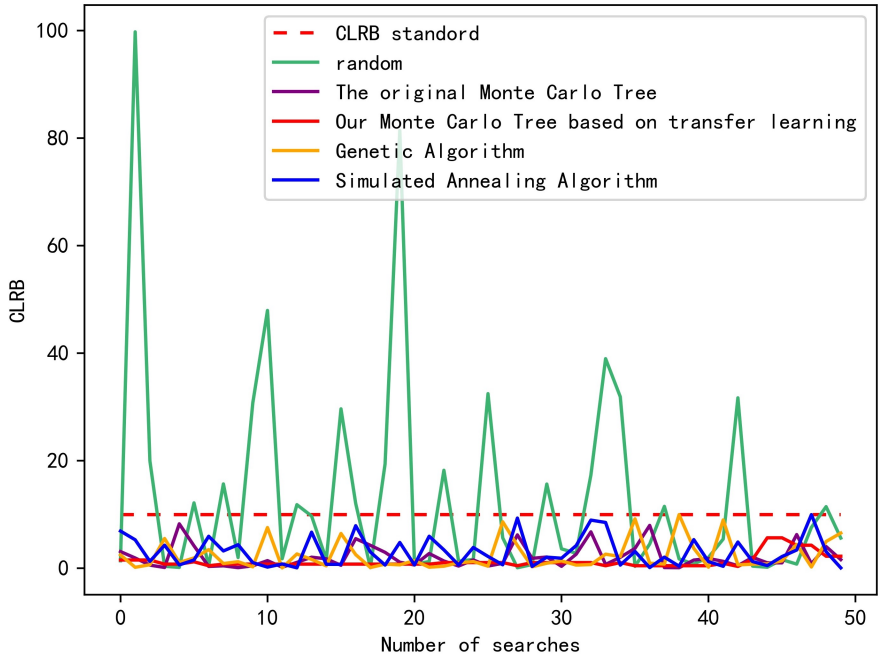

Fig. 3. Search results performance comparison(s).

Experimental results show that our proposed method can not only select the minimize and optimize radar node combination but also has a faster speed.

References

1. Bello, I., Pham, H., Le, Q.V., Norouzi, M., Bengio, S.: Neural combinatorial optimization with reinforcement learning. arXiv preprint arXiv:1611.09940 (2016)
2. Berenguer, I., Wang, X., Krishnamurthy, V.: Adaptive mimo antenna selection. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. vol. 1, pp. 21–26. IEEE (2003)
3. Bertsimas, D., Tsitsiklis, J.: Simulated annealing. Statistical science **8**(1), 10–15 (1993)
4. Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S.: A survey of monte carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in games **4**(1), 1–43 (2012)
5. Godrich, H., Petropulu, A.P., Poor, H.V.: Sensor selection in distributed multiple-radar architectures for localization: A knapsack problem formulation. IEEE Transactions on Signal Processing **60**(1), 247–260 (2011)
6. Gorji, A.A., Tharmarasa, R., Blair, W., Kirubarajan, T.: Multiple unresolved target localization and tracking using colocated mimo radars. IEEE Transactions on Aerospace and Electronic Systems **48**(3), 2498–2517 (2012)
7. Guo, J., Tao, H.: Cramer-rao lower bounds of target positioning estimate in netted radar system. Digital Signal Processing **118**, 103222 (2021)

8. Hou, Y., Ong, Y.S., Feng, L., Zurada, J.M.: An evolutionary transfer reinforcement learning framework for multiagent systems. IEEE Transactions on Evolutionary Computation **21**(4), 601–615 (2017)
9. Huang, B., Feng, F., Lu, C., Magliacane, S., Zhang, K.: Adarl: What, where, and how to adapt in transfer reinforcement learning. arXiv preprint arXiv:2107.02729 (2021)
10. Konidaris, G., Barto, A.: Autonomous shaping: Knowledge transfer in reinforcement learning. In: Proceedings of the 23rd international conference on Machine learning. pp. 489–496 (2006)
11. Lazaric, A.: Transfer in reinforcement learning: a framework and a survey. In: Reinforcement Learning, pp. 143–173. Springer (2012)
12. Mirjalili, S.: Genetic algorithm. In: Evolutionary algorithms and neural networks, pp. 43–55. Springer (2019)
13. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature **529**(7587), 484–489 (2016)
14. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al.: Mastering the game of go without human knowledge. nature **550**(7676), 354–359 (2017)
15. Torrey, L., Shavlik, J.: Transfer learning. In: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, pp. 242–264. IGI global (2010)
16. Zhu, Z., Lin, K., Zhou, J.: Transfer learning in deep reinforcement learning: A survey. arXiv preprint arXiv:2009.07888 (2020)