User Association and Load Balancing based on Monte Carlo Tree Search

Hyeon-Min Yoo, Jung-Mo Moon*, Jee-Hyeon Na*, Een-Kee Hong
Kyung Hee University, *ETRI

yhm1620@khu.ac.kr, *jmmoon@etri.re.kr, *jhna@etri.re.kr, ekhong@khu.ac.kr

Abstract

The user association is becoming more and more complicate in a 5G ultra-dense heterogeneous networks (UD-HetNets) consisting of multi-tier base stations. In UD-HetNets, finding the optimal user association result achieved by considering all combinations of associations between user equipments (UEs) and small cell base stations (SBSs) causes extremely high computational complexity. In order to achieve near-optimal solution with reasonable complexity, we propose to apply Monte Carlo tree search (MCTS) algorithm to user association. MCTS finds a near-optimal solution by repeating random sampling several times without exploring all cases and achieves a proper load balancing. Simulation results show that the network capacity of proposed algorithm is significantly higher than conventional user association schemes, and the improvement is approximately 15% ~ 23% according to the number of tree search iterations.

Keywords: User Association; Ultra-dense Heterogeneous Network; Load Balancing; Monte Carlo Tree Search; Cell Range Expansion

I. Introduction

User association plays an important role in achieving load balancing and improving spectral efficiency of 5G network [1]. In UD-HetNets, due to the high density of base stations (BSs) and UEs, the number of combinations of UE-BS associations becomes too large and the computational complexity increases significantly. For example, if there are N BSs and M UEs, the number of possible combinations is N^M. Among these many combinations, the network requires to find optimal association result that maximizes the overall network capacity using a new user association algorithm with reasonable computational complexity.

In the conventional network, the cell range expansion (CRE) in which the UEs that requires high data rate (abbreviated as *HDR UE*s hereafter) offloads to the SBS with the highest biased received power has been widely used [2]. However, this algorithm can lead to load imbalance across the SBSs for non-uniform UE distribution [3]. Each UE cannot achieve high throughput and overall network capacity is degraded. Therefore, it is necessary to distribute the UEs to multiple SBSs effectively. To solve the load balancing problem and increase the network capacity, we need more sophisticated user association algorithm.

In this paper, we propose a centralized user association algorithm. The future network is envisioned to be configured based on the O-RAN architecture in which the RAN intelligent controller (RIC) manages multiple BSs and UEs [4]. In the O-RAN architecture, the centralized approach is suitable because the RIC knows all information of the network it manages. Centralized schemes show superior performance than the other schemes such as distributed and hybrid, but this scheme requires high computational complexity [5]. To overcome this drawback, our proposed algorithm is based on MCTS. The MCTS is a heuristic tree search method for finding near-optimal solution by repeating random sampling several times instead exploring all cases [6]. So, the user association algorithm based on MCTS can achieve near-optimal association result and ensure high performance with lower computational complexity. This result is the selected UEs-SBSs combination among a large number of possible combinations, which means that each UE associates to the best SBS, with respect to network capacity.

II. Point

One iteration of MCTS-based user association algorithm consists of four steps: *Selection, Expansion, Simulation, Backpropagation* to find the optimal solution and iterations are repeated to find the optimal solution. The more iterations progress, the deeper the tree will expand from the root node. The deeper the tree, the more UEs-SBSs combinations can be sampled, and the higher the probability of obtaining the near-optimal UEs-SBSs combination. Initially, the root node corresponds to the state where no *HDR UE* has yet been offloaded to a SBS. The root node does not have a child node yet, so there is no child node to select. Therefore, at this time, the *Expansion* step is executed immediately without going through the *Selection* step. In *Expansion* step, root node makes a child node. This child node represents a situation where an *HDR UE* associates to a SBS.

If the UE1 is connected to a distant SBS, the performance is surely very degraded. So, it is reasonable that UE1 associates only to the one of its neighbor SBSs (N-SBSs) that provide the UE1 with high received power. The number of N-SBSs is adjustable value, and it should be preliminarily determined before the algorithm starts to run. Now the expanded child node contains UE1 and one of its N-SBSs. This child node immediately runs *Simulation* step. In *Simulation* step, under the state that UE1 associates to the selected N-SBS, all other *HDR UE*s also associate to one of its N-SBSs by random sampling through Monte Carlo method. Then we can get a *network state* that all *HDR UE*s associate to the one of its N-SBSs. This *network state* is just one of the UEs-N-SBSs combinations, which is an entirely random combination without considering load balancing. The child node stores the network capacity value by aggregating all throughputs of each UE in this combination in *Backpropagation* step.

Next, root node makes other child nodes indicating that the UE1 associates to other N-SBSs through *Expansion*. *Simulation* and *Backpropagation* are also repeated in these child nodes. Then, if root node makes child nodes as many as the predetermined number of N-SBSs, the root node becomes a *fully expanded* node. It means that now root node couldn't make child node anymore. On the other hand, since the child nodes of the root node do not yet have their child node, they can expand the tree by creating their child nodes. So, in order for the tree to continue its *Expansion*, *fully expanded* root node must select one of its child nodes. So, root node executes *Selection* step. In *Selection*, it can select one of its child nodes by tree policy. It has been proven that high performance is guaranteed when upper confidence bounds for trees (UCT) applied as a tree policy [6].

$\bar{X}_j$ is a value between [0, 1], which means the average reward of child node j. In our proposed algorithm, $\bar{X}_j$ is the ratio of the throughput of the child node j to the sum of throughputs of all child nodes that have the same parent node including itself. If we don't consider the bias term

$\sqrt{(2\ln n)/n_j}$, the higher the value of $\bar{X}_j$ is, the higher the probability of this node being selected (i.e., *exploitation*). However, although the selected child node shows good performance right now, it can be a bad choice if we consider the associations of all other *HDR UE*s. Therefore, it is necessary to expand the tree by exploring multiple nodes through bias term (i.e., *exploration*). n is the number of visits of a *fully expanded* node (i.e., parent node), and n_j in the denominator is the number of visits to its child node j. So, the values of bias term of the less-visited child nodes become larger, and UCT can ensures *exploration* of all child nodes. Thus, using UCT as a tree policy makes the tree expand while balancing *exploitation* and *exploration*. The *exploration constant* C_p can be adjusted to encourage or discourage *exploration*.

The child node with the highest UCT value is selected in *Selection* step and this child node becomes the *current node*. The *current node* makes its child node through *Expansion*. This child node contains UE2 and one of its N-SBS. And similarly, this node runs *Simulation-Backpropagation*. The important thing is that, at this time, the UE1-N-SBS (represented by the parent node) is already determined and the UE2's association (represented by the *current node*) is tested. Under these conditions, all the remaining *HDR UE*s associate to its N-SBS and an UEs-N-SBSs combination is created. In this combination, the capacity value of the *current node* is calculated by aggregating the throughput of each UE. In *Backpropagation,* the *current node* stores this capacity value and updates the capacity value of upper nodes, including its parent node. However, the upper nodes already have its previously calculated capacity value. In this case, in our proposed algorithm, the upper nodes replace its capacity value with the average of the existing value and updated value. The average value is obtained by dividing *cumulative capacity*, which is the accumulated value of the capacity value whenever *Backpropagation* occurs, by the number of visits.

Starting at the root node, the tree is expanded by the predetermined number of iterations (*nIter*). However, as mentioned above, before the root node become a *fully expanded* node, iterations run without *Selection*. Fig. 2 shows an example of an expanding tree. After the *Expansion* is completed as much as *nIter*, the tree returns the best child node of root node with the highest capacity value. In other words, among the child nodes containing UE1 and one of its N-SBS, the tree will return the child node containing the optimal N-SBS in terms of the overall network capacity. When we denote the N-SBS of the returned child node as SBS1*, UE1-SBS1* becomes new root node and repeats the same process. It means that the algorithm is executed while UE1 is associated to its optimal N-SBS (i.e., SBS1*). This time, the tree returns UE2 and its optimal N-SBS. When this operation is repeated for all *HDR UE*s, we can obtain the optimal UEs-N-SBSs combination.

We evaluated the performance of our algorithm with Monte Carlo simulation to validate that our proposed algorithm is applicable to diverse networks. So, in each simulation run, the UEs and BSs in a network are distributed randomly according to a homogeneous Poisson point process (PPP). We compare the performance of proposed algorithm with CRE scheme in terms of the overall network capacity. The percentage of capacity increment for 1,000 different networks is expressed as a complementary cumulative distribution function (CCDF). We set simulation area to $1km^2$, and PPP intensity of macrocell, SBS, UE to 3, 30, 100, respectively. We suppose that *HDR UE*s are offloaded to one of its three N-SBSs and exploration constant C_p is set to 0.1.

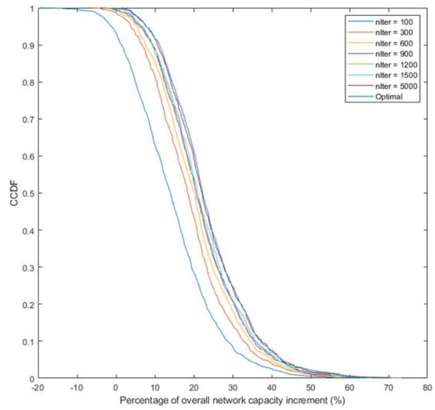

Fig. 1 Performance for the number of iterations (*nIter*)

Fig. 1 illustrates the percentage of capacity increment per 1,000 different networks, and the effect of the depth of the tree by varying *nIter*. Table 1 summarizes the specific numerical values in Fig. 1. Even if *nIter* is only 100, it shows a 14.6% average capacity improvement than CRE. The performance improvement jumps at *nIter* = 300, which means that this *nIter* is near-optimal. From *nIter* = 1200, although the depth of the tree deepens, the average improvement hardly increases and starts to converge to the optimal.

Table 1 Specific numerical values of graph in Fig. 1

nIter	Ave. capacity improvement [%]	Max. capacity improvement [%]
100	14.60	56.39
300	19.13	65.59
600	20.68	67.67
900	21.71	71.57
1200	22.02	72.35
1500	22.20	72.66
5000	23.32	73.72
Optimal	23.58	74.12

III. Conclusion

In this paper, we propose the user association algorithm based on MCTS. We proved the universality of our proposed algorithm by applying it to the diverse networks modeled by randomly distributed UEs and BSs. We hope that our algorithm would become a practical solution of the long-standing user association problem.

ACKNOWLEDGMENT

"This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2021-0-02046) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)"

Reference

[1] Y. Zhang, L. Xiong and J. Yu, "Deep Learning Based User Association in Heterogeneous Wireless Networks," in IEEE Access, vol. 8, pp. 197439-197447, 2020.
[2] K. Okino, T. Nakayama, C. Yamazaki, H. Sato and Y. Kusano, "Pico Cell Range Expansion with Interference Mitigation toward LTE-Advanced Heterogeneous Networks," 2011 IEEE International Conference on Communications Workshops (ICC), 2011, pp. 1-5
[3] J. G. Andrews, S. Singh, Q. Ye, X. Lin and H. S. Dhillon, "An overview of load balancing in hetnets: old myths and open problems," in IEEE Wireless Communications, vol. 21, no. 2, pp. 18-25, April 2014.
[4] L. Bonati, S. D'Oro, M. Polese, S. Basagni and T. Melodia, "Intelligence and Learning in O-RAN for Data-Driven NextG Cellular Networks," in IEEE Communications Magazine, vol. 59, no. 10, pp. 21-27, October 2021.
[5] D. Liu et al., User Association in 5G Networks: A Survey and an Outlook, IEEE Commun. Surv. Tutor. 18 (2) (2016) 1018-1044.
[6] C. B. Browne et al., A Survey of Monte Carlo Tree Search Methods, IEEE Trans Comput Intell AI Games 4 (1) (2012) 1-43.