**HAP Placement in Wireless Powered Communication Networks Using Deep Learning**

**Hong-Sik Kim, Hanyang University, Department of Computer Software**

**Abstract**

HAP (Hybrid Access Point) is a node in Wireless Powered Communication Networks (WPCN) that can distribute energy to each wireless device and also can receive information from these devices. For efficient use of network, we should maximize the throughput of the network. There are two kinds of metrics for throughput, that is, sum throughput and common throughput, each is the sum and minimum value of throughput between a HAP and each wireless device, respectively. There are two types of throughput maximization problems, sum throughput maximization and common throughput maximization. In this paper, we discuss about the latter. We used deep learning methodology to maximize common throughput. Our study implies that deep learning can be applied to maximize a simple function of common throughput maximization, which is a convex function or a combination of a few convex functions, and shows better performance than mathematical methodologies.

**1. Introduction**

In WPCN, there is Access Point (AP) mechanism **[1]** which contains energy nodes (EN), wireless devices (WD) and access points (AP). First, energy nodes send energy to each wireless devices. When ENs receive the energy, it sends information to APs using the energy. That is, ENs send energy to WDs and WDs send information to APs. We can encapsulate AP and EN into Hybrid Access Point (HAP) and so can describe HAP mechanism. In this mechanism, the HAP sends energy to each WD, and each WD sends information to the HAP. The HAP allocate time for sending energy to each WD to itself, and for sending information to each WDs, so time allocation to itself and each WDs is also an important issue.

Because the distance between the HAP and each WD is different among each WD, there is energy efficiency gap between the WDs caused by the difference of throughput for each WD. That is, a WD near to the HAP receives more energy and uses less energy to transmit information, and another WD far from the HAP receives less energy but uses more energy to transmit information. To solve this unfairness problem, the worst case, a WD which receives least energy and uses most energy, is very important. For this case, we use the concept of common throughput which is the minimum value of throughput among the throughput values of each WD, and we concentrate on maximizing the common throughput value in the WPCN environment.

This paper introduces a methodology to place HAPs in WPCN environment to maximize common throughput when time allocation is optimized, by using deep learning, and shows that this methodology is meaningful to solve this problem and shows better performance than the mathematical methodology already studied.

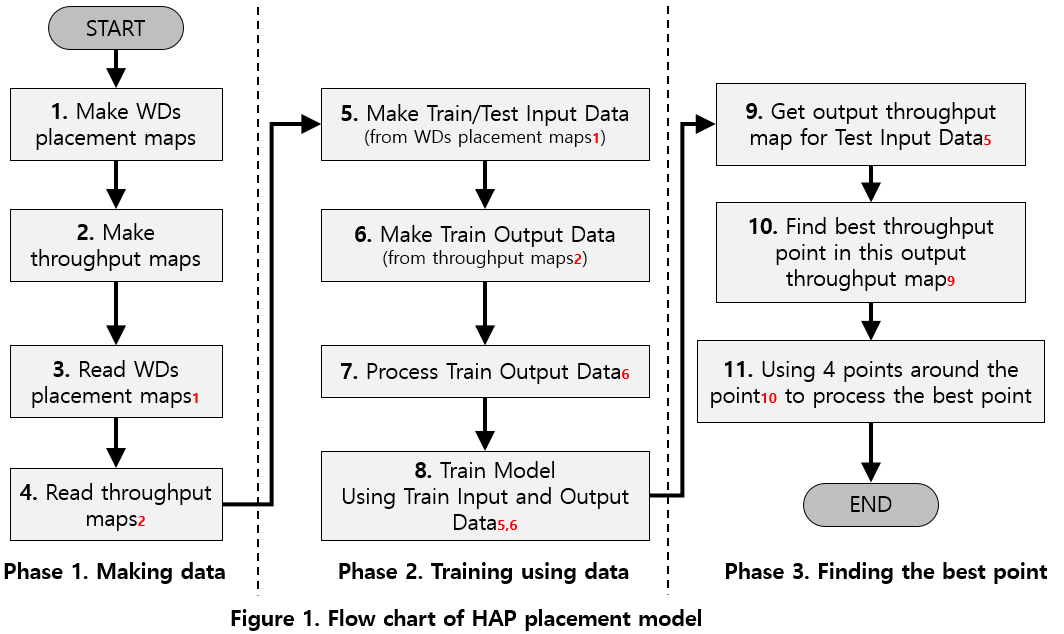
**2. Related Works**

In **[2]**, Suzhi Bi and Rui Zhang researched about the placement optimization of Energy and Information Access Point in WPCN using Bi-section search method, Greedy algorithm, Trial-and-error method, and alternating method for joint AP-EN placement. There can be more than 1 HAPs WDs in the supposed environment of this paper. Its methodology is repeatedly adding HAPs and check if each WD satisfies conditions in the environment.

**3. HAP Placement Model**

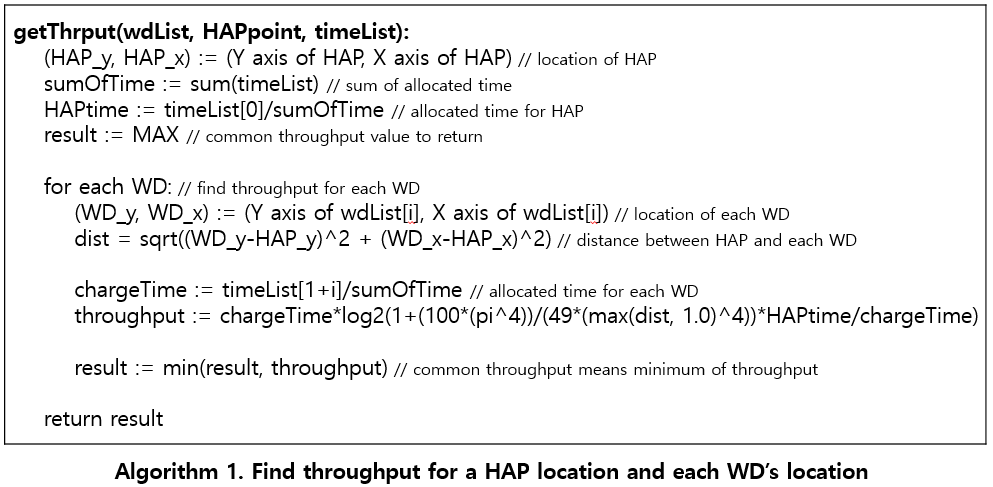
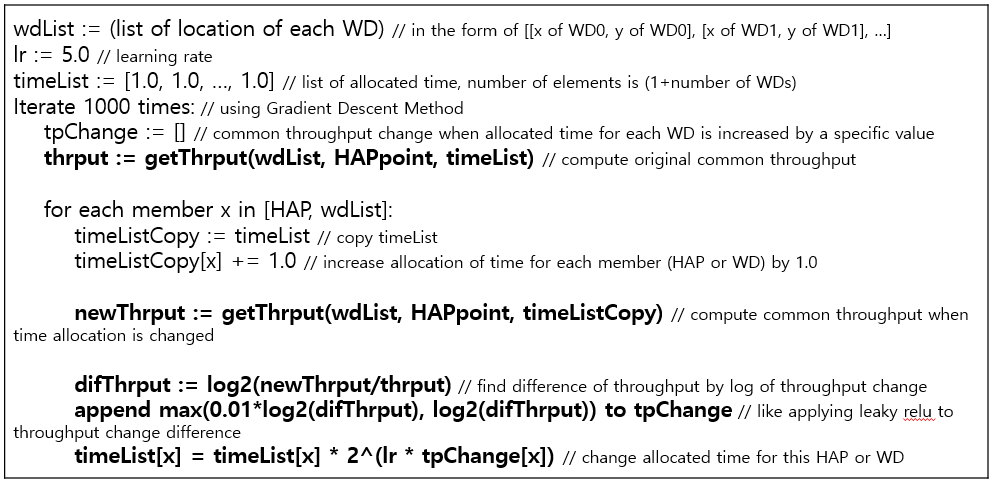
**3.1. Overview**

**Figure 1** is the flow chart for the HAP placement model. The model is composed of three phases. Making data is the first phase to create training and test data. Training using data is the next phase to process the data to training and test data for the deep learning model, and train using the model. Finding the best point is the last phase to find the best HAP placement point using the throughput map derived by this model.



**WDs placement map** means the square map with rows and columns, and random K blocks in the map contains a WD. There are no WDs with the same position in the map. From now on, we will call this . **Throughput map** means the square map with rows and columns, and each block contains the throughput value where the HAP is located in this block of . From now on, we will call this . **Best throughput point** means the point that maximize throughput value in . we will call this .

**3.2. Making Data**

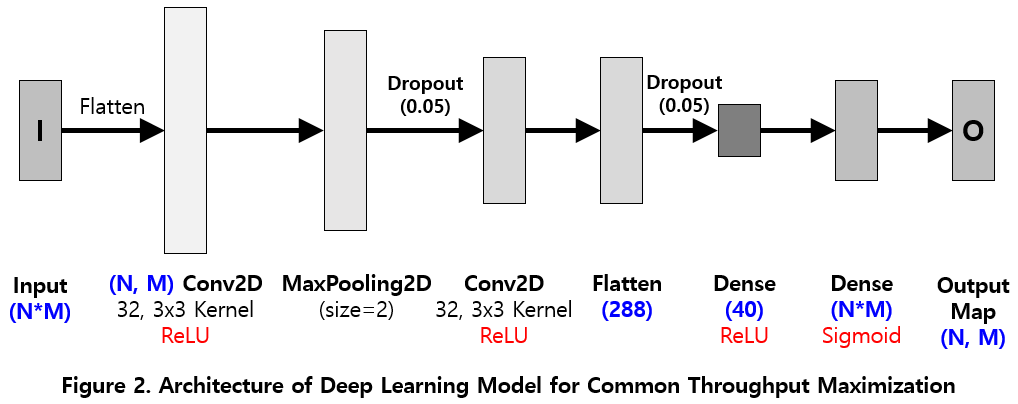
To make s, first define a square map with rows and columns. Then repeat placing a WD at the randomly selected point times. To make s using s, place HAP at each point in s and compute throughput for the location of HAP and each WDs using **Algorithm 1** because throughput value is computed using **Formula (6)** in **[3]**. getThrput function finds optimal time allocation given . Because I supposed that and where is the distance from the HAP and each WD, this formula can be converted into **(1)**. To prevent divide by 0 error and consider the limit of throughput, I supposed that distance is 1.0 when actual distance is less than 1.0.

Then, because s and s are saved as text files, the model must read them before using them.

**3.3. Training Using Data**

First, make input data for training and testing based on , supposing that the number of training and testing data is and each. The model considers first maps as training data and next maps as test data. The input data is an map whose value at each block is -1 when a WD is at this block and 0 otherwise. Then make output data for training based on s corresponding to s. The output data is an map whose value at each block is processed version of the common throughput value derived by **Algorithm 1** where the HAP is at this block. The following is the processing procedure. First, find maximum throughput value for each training output map , and then divide each value , at the intersection of row and column in this map by the maximum value among these values. Last, process each value at each block using **(2)**.

Then training using input data s and corresponding output data s using the deep learning model described in **Figure 2** with Adam optimizer **[4]** with learning rate 0.001 and 1,000 epochs.

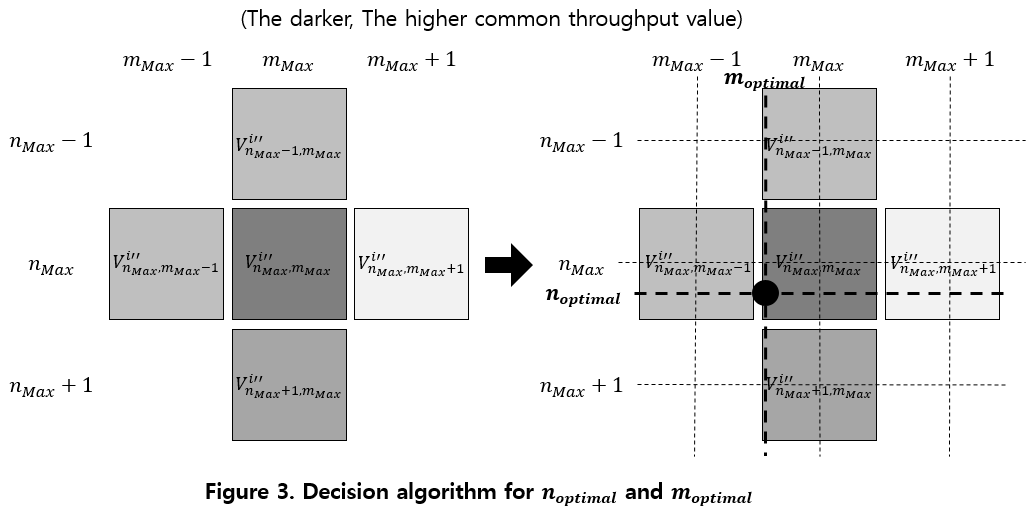


**3.4. Finding the Best Point**

Using test input data, the model finds best point for HAP placement. For each test input data created based on in **3.3.**, input these data into the model trained in **3.3.** and get output maps corresponding to . For each value at each block in each output map is converted by **(3)** using inverse function of sigmoid function, to convert them from from into form, where is the estimated common throughput value.

Then, for each output map, the model finds maximum value among values in blocks of this map. Let’s call row and column axis of this value in the map and each, and call the maximum value . Then the row axis and column axis of optimal HAP location is computed by **(4)** and **(5)** each, and is computed by **(6)**, described in **Figure 3**.

If is greater than , moves down from original position and otherwise it moves up. Similarly, if is greater than , moves right and otherwise it moves down. Because original common throughput, and can be converted into each other by just a linear transmission, there is no difference of and between when converted into and do not converted into any other form.



**4. Experiment**

**4.1. Experiment Design**

(This part will be written later)

**4.2. Metrics**

(This part will be written later)

**4.3. Experiment environment**

(This part will be written later)

**4.4. Experiment Result**

(This part will be written later)

**5. Conclusion**

(This part will be written later)

**6. References**

**[1]** Suzhi Bi, Yong Zeng, and Rui Zhang, “Wireless Powered Communication Networks: An Overview”, IEEE, available online at <https://arXiv:1508.06366>

**[2]** Suzhi Bi, Member, IEEE, and Rui Zhang, “Placement Optimization of Energy and Information Access Points in Wireless Powered Communication Networks”, IEEE Transactions on wireless communications, VOL. 15, NO. 3, MARCH 2016

**[3]** Hyungsik Ju and Rui Zhang, “Throughput Maximization in Wireless Powered Communication Networks”, available online at <https://arXiv:1304.7886v4>

**[4]** Diederik P. Kingma, Jimmy Lei Ba, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION”, ICLR 2015, available online at <https://arXiv:1412.6980>