CEPAT Journal of Computer Engineering: Progress, Application and Technology
Vol. 5, No. 2, May 2026, pp. 20-32
ISSN: 2963-6728, DOI: https://doi.org/10.25124/cepat.v3i02.8025 ❒ 20

Resource block allocation: performance comparison of auction, greedy, and round robin algorithms

Fortuna Kahyangan[1], Muhammad Harits Fakhrudin[1], Revin Abyan Furqan[1], Linda Meylani [1], Vinsensius Sigit Widhi Prabowo[1]
[1]Department of Telecommunication Engineering, School of Electrical Engineering, Telkom University, Indonesia

Article Info

Article history:

Received February 8, 2026
Revised March 27, 2026
Accepted April 25, 2026

Keywords:

HetNets Greedy Auction Round Robin resource block K-Medoids

ABSTRACT

Modelling and design of radio resource allocation in Heterogeneous Networks (HetNets) is a very important topic in the development of modern wireless communication technologies. This study evaluates the performance of resource block (RB) allocation in a two-level HetNet model consisting of one macro cell base station (MBS) and four small cell base stations (SBS). Utilizing K-Medoids clustering, allocations are analyzed under various conditions using Greedy, Auction, and Round Robin algorithms. Simulations reveal that the Greedy algorithm outperforms the Auction and round robin algorithms in optimizing data rate, sum rate, spectral efficiency, power efficiency, and fairness. Specifically, the Greedy algorithm achieves an average data rate of 1.642×10^7 bps, an average sum rate rate of 1.218×10^9 bps, an average spectral efficiency of 3.046 bps/Hz, an average power efficiency of 1.650×10^7 bps/W, and an average fairness of 0.329, indicating its effectiveness in improving HetNet performance.

Corresponding Author:

Fortuna Kahyangan
Department of Telecommunication Engineering, School of Electrical Engineering, Telkom University
Bandung, Indonesia
Email: fortunakahyangan@student.telkomuniversity.ac.id

1. INTRODUCTION

Radio resource modelling and allocation design in Heterogeneous Networks (HetNets) is a very important topic in the development of modern wireless communication technologies. HetNets are a critical technology in beyond telecommunication systems [1]. HetNets leverage the densification of small cells to improve key network parameters such as coverage, capacity, latency, and load distribution. The architecture of HetNets consists of multiple types of cells, including macrocells and small cells, which operate together to enhance the overall network performance HetNets consist of various cell types, microcells, macrocells, picocells, and femtocells, each with their own coverage and power. With the integration of these different cell types, network capacity and spectrum efficiency can be increased while reducing power consumption. However, this diversity also poses challenges in terms of interference management and optimal resource allocation. According to research, effective radio resource management in HetNets requires an approach that can handle the diversity of user traffic [2].

In addressing the resource allocation problem in HetNets, several algorithms have been proposed, including the auction algorithm, greedy algorithm, and Round Robin algorithm. The auction algorithm, greedy algorithm, and round robin algorithm offer different approaches in addressing the resource allocation problem in HetNets. The auction algorithm enables resource allocation based on an auction mechanism that can optimise spectrum efficiency by considering bids from various users [3]. The greedy algorithm, on the other hand, focuses on resource allocation iteratively by selecting the best available option at each step [3]. Meanwhile, the round robin algorithm distributes resources evenly among users [4].

One technique that is also used in this problem is clustering. Clustering, one of the techniques in machine learning, has been proposed as a method to improve resource management in heterogeneous networks (HetNets). This technique enables automatic grouping of cells based on similar characteristics, such as capacity requirements, signal quality, and usage patterns. By grouping similar cells together, clustering techniques aim to allocate resources more efficiently, reduce interference, and maximise throughput. For example, in congested networks, clustering can be used to identify and manage network traffic congestion, allowing network operators to dynamically adjust resource allocation to improve overall performance.

Resource allocation in heterogeneous networks (HetNets) is critical to ensure optimal performance and capacity of the entire network. A major problem encountered is the inefficient use of resources, which is caused by the imbalanced placement of base stations such as microcells, macrocells, picocells, and femtocells. In some areas, there is a high density of users resulting in data traffic congestion, while in other areas, resources are underutilised, causing the potential of the network to not be fully tapped. This inequity in capacity distribution results in some areas experiencing degraded service quality, while others are unable to utilise the available resources effectively.

Resource Block (RB) is the fundamental unit of resource allocation in wireless communication systems such as LTE (Long-Term Evolution) [5]. Each Resource Block consists of several sub-carriers in the frequency domain allocated for a specific period in the time domain. Resource Blocks are used to manage and allocate frequency spectrum to various users within the network to ensure efficient and optimized data transmission. The use of RBs allows the network to dynamically handle interference and optimize throughput based on the users' needs and current network conditions. Some crucial aspects of this allocation issue include (RB) allocation, where imbalances in RB allocation across different cells result in network disruptions and affect service quality for end users. Further, non-optimal allocation of frequency spectrum contributes to inter-network interference and reduces capacity and signal quality. In addition, inefficient use of power not only results in excessive power consumption but also increases operational costs and negatively impacts the environment, demanding a more strategic approach to network resource management.

The objective of this research is to allocate resource blocks to each user using three different algorithms: greedy algorithm, auction algorithm, and round robin algorithm. These approaches will be applied in two different clustering system models, namely K-Medoids and fixed clustering, to evaluate and compare their effectiveness in improving the management and distribution of resources in heterogeneous networks (HetNets). In this way, this research aims to identify the most efficient and fair allocation method for resource utilisation in dynamic and diverse networks.

The scientific contribution of our work in this research is to propose a HetNets system model in the form of a two-tier network with macrocell as the first-tier cell, small cell as the second-tier cell, and there are several user equipment connected to the small cell base station. In this study, we apply two system models, namely K-Medoids clustering and Fixed clustering, to evaluate and compare their effectiveness in improving resource management and distribution in HetNets networks. We use greedy algorithm, auction algorithm, and round robin algorithm to allocate resource blocks to each user.

The organization of this paper consists of four main sections. In Section 1, an introduction is presented about the problem statement, research objectives, scientific contributions, and paper organization. In Section 2, the proposed method is presented about the system model, algorithm process, and simulation parameters. In Section 3, results and discussions on the analysis of simulation results on system performance are presented. In Section 4, the conclusion of this paper is presented.

2. METHOD

The primary goal of this system modeling is to simulate system performance under various conditions and configurations. This involves allocating resource blocks to each user using three different algorithms: the Greedy algorithm, the Auction algorithm, and the Round Robin algorithm. The selection of three algorithms-Greedy, Auction, and Round Robin-in this study is based on their uniqueness and advantages in managing resources in a HetNets. The Greedy algorithm was chosen for its ability to quickly and simply distribute resources by prioritising the most profitable options at each step, although it does not always produce the global optimal solution. The Auction algorithm was used because of its advantage in creating competition among users for resource allocation, which often results in a near-optimal solution by considering users individual valuations of resources. Meanwhile, the Round Robin algorithm was chosen for its simplicity and its ability to provide a fair allocation by dividing the resources in rotation without discrimination, thus ensuring each user gets an equal share. By comparing these three algorithms, this research can evaluate the effectiveness of each in different scenarios, helping in determining the best approach for resource management in HetNets.

In addition to resource allocation, the model seeks to enhance overall network capacity and optimize performance parameters. These parameters include data rate, sum rate, spectral efficiency, power efficiency, and fairness within the system. By improving these metrics, the system intends to ensure a more efficient and equitable distribution of resources, thereby enhancing the user experience and the overall performance of the network.

2.1. System Model

The HetNets system model proposed in this study is a two-tier network consisting of macrocells as first-tier cells and small cells as second-tier cells. This model has three main agents: Macrocell Base Station (MBS), Small Cell Base Station (SBS), and Small Cell User Equipment (SUE). The communication scheme used is a downlink scheme, which means communication occurs from the base station to the user equipment (UE). In this model, the SUE is in idle mode, which means it is not involved in active communication but remains connected to the network and ready to switch to active mode if needed. The implementation of idle mode aims to reduce the complexity of cooperative communication and maximise resource usage.

SUEs are randomly deployed with locations determined using polar coordinates based on distance (d) and random angle (theta), so that the closest distance of SUEs to each SBS can be calculated. In this model, the distance from the MBS point to each SBS is considered the same to simplify the calculation. The figure above shows a network structure with an MBS at the centre and multiple SBSs scattered around the MBS, each serving multiple SUEs. The main signal flows from the MBS to the SBSs, and from the SBSs to the SUEs, showing the communication route in this downlink scheme.

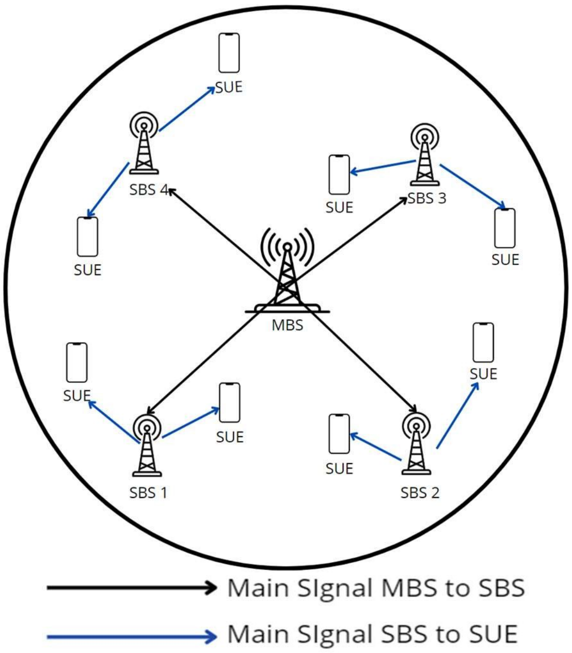

Figure 1. System Model

In this study, the clustering technique was also used. Clustering is a data analysis technique that aims to classify data into groups (clusters) based on similar characteristics [6]. The main objective of clustering is to maximize intra-cluster similarity and minimize inter-cluster similarity. The clustering algorithm used in this research is K-Medoids, which is a variant of K-Means but more robust to outliers [7].

K-medoids is one of the popular clustering algorithms and is a variant of the K-means algorithm. The way K-medoids works involves selecting several k objects from the data set as medoids or cluster centers[7]. The algorithm aims to partition the data into groups by minimizing inter-cluster similarity and maximizing intra-cluster similarity. K-medoids operates by calculating the distance between data objects, then iteratively replacing medoids with non-medoids if the replacement improves the overall clustering quality [8]. One of the advantages of K-medoids over K-means is its resilience to outliers, as it uses actual objects as cluster centers instead of average values [9]. The following is a visualisation of the system model in this study

2.2. Algorithm Process

This research compares performance parameters using greedy algorithm, auction algorithm, and round robin algorithm.

2.2.1. Greedy algorithm

A greedy algorithm is a method used to solve optimization problems by making decisions that appear to be the most optimal at each step, without considering decisions that have been made previously. It works on the principle of choosing the best local solution in the hope of achieving a globally optimal solution. Although simple and efficient, greedy algorithms do not always guarantee a globally optimal solution because the decisions taken at each step are irreversible and do not consider possible future solutions [8].

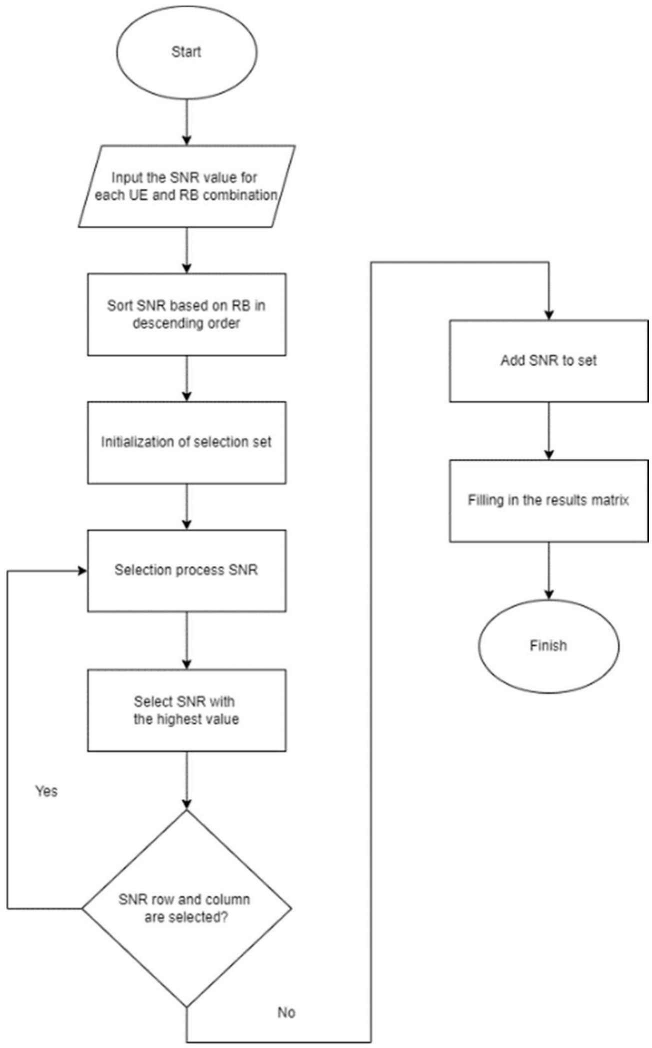

Figure 2. Flowchart of greedy algorithm

The diagram above illustrates the flowchart of greedy algorithm. The greedy algorithm method and implementation starts with initialisation and data collection. The first step is to collect all the elements in the input parameters, where each element consists of a resource block (RB) value and its row and column coordinates. This collection allows the valuation of each RB based on the value provided by the user. Once all the elements are collected, they are sorted in descending order by RB value so that allocation can start from the highest value. This process ensures that the elements with the highest values are prioritised in the allocation.

Next, the algorithm proceeds to the greedy selection process. Several data structures are initialised, including a "selected" list to store the selected row and column pairs, and two sets of "rows_selected" and "cols_selected" to keep track of the selected rows and columns. The algorithm then goes through the sorted elements and selects the element with the highest value whose rows and columns have not been selected. The selected elements are added to the "selected" list, and their rows and columns are marked as selected, ensuring that each row and column is only selected once. After the selection process, a result matrix is initialised with a value of zero, of the same size as the input parameters. This matrix is updated with the RB values for the row and column pairs that have been selected in the selection process, ensuring that the user gets the RB that is most valuable to them.

2.2.2. Auction Algorithm

The auction algorithm is a method used to solve various optimization problems, such as the assignment problem and resource allocation. It mimics an auction process where bidders submit higher or lower bids to outbid each other. The algorithm is designed to ensure that resource block allocation is optimized and fair, avoiding manipulation by bidders [9].

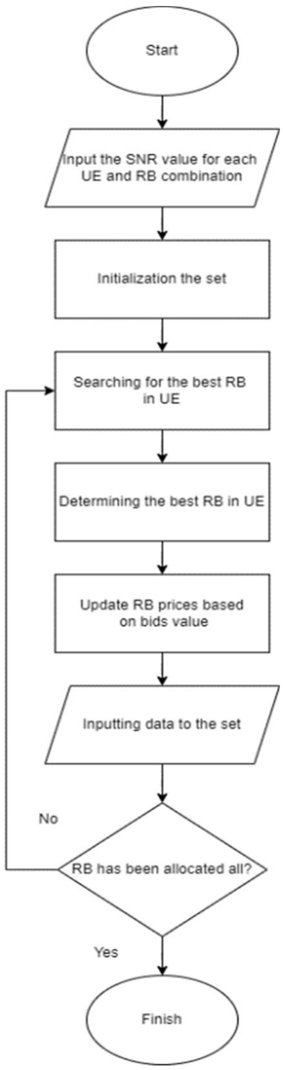

Figure 3. Flowchart of auction algorithm

The diagram above illustrates the flowchart of auction algorithm. The method and implementation of the auction algorithm starts with initialisation and data preparation. The first step is to determine the number of users and the number of available Resource Blocks (RBs) from the input parameters, which is the basis for knowing the scale of allocation that needs to be done. A result matrix is initialised with a value of zero to store the result of RB allocation for each user. In addition, a bid matrix is also initialised with a value of zero to keep track of the bids from each user for each RB, as well as the RB price which is initialised with zero. An "assigned_rb" list is also set up to keep track of the RBs that have been allocated to each user. In the user bidding process, each user values each RB based on the difference between the RB value and the current price, then selects the RB that provides the highest value. Users then make a bid slightly higher than the highest value they obtained and store it in the bid matrix. RBs are allocated to the user who offers the highest value, and if there are multiple users with the same highest bid, the user with the highest RB value for that RB will be selected as the winner.

Next, the price of the RB is adjusted to the highest bid received to ensure that each RB is awarded to the user who values it the highest. If there are multiple users offering the same highest value, the losing user will be marked as unallocated and return in the queue to make a new bid. Once all bids are complete, the RB will be allocated to the user who won the last bid for that RB. At this stage, a conflict check is performed; if there are multiple users allocated to the same RB, the user with the highest RB value will still get the RB, while the other users will return in the queue for new bids. Finally, the result matrix is updated to reflect the final allocation of RBs to each user, based on the highest bid and any price adjustments made.

2.2.3. Round Robin Algorithm

Round Robin algorithm is one of the scheduling methods used for resource block allocation in computing and networking systems. This method works by distributing resources in rotation to each process or user in the queue. This process keeps repeating until all processes or users get their turn. The Round Robin algorithm ensures that each process or user gets a fair allocation of resources, without prioritizing one process over another [10].

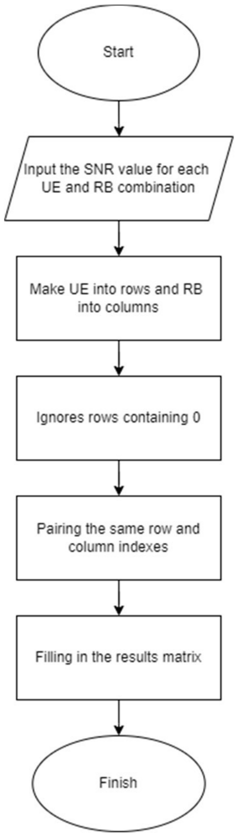

Figure 4. Flowchart of round robin algorithm

The diagram above illustrates the flowchart of round robin algorithm. The method and implementation of the auction algorithm starts with initialisation and data preparation, where the first step is to determine the number of users and the number of available Resource Blocks (RBs) from the input parameters. A zero-value result matrix is initialised to store the result of RB allocation for each user. Next, user validation is performed by identifying users who are connected to the small cell, i.e. those who have at least one RB request. Users whose request values are all zero are excluded from the allocation process. After identifying the users connected to the small cell, their total number is counted to know how many users the algorithm needs to serve. If no users connected to small cells are detected, the algorithm immediately returns a result matrix containing only zeros, indicating that no allocation needs to be done. However, if there are users connected to the small cell, the process continues to the next step to perform RB allocation.

The allocation process starts with the initialisation of the allocation status, where a status list is created to track whether each RB has been allocated or not, as well as the user status to ensure each user gets a fair chance. The RB index starts from the first index to be allocated and will be increased in turn to ensure fair allocation. The algorithm then enters a loop that iterates through each user connected to the identified small cell. In the loop, the algorithm checks whether the current RB has been allocated and whether the current user has received RB. If the RB has already been allocated or the user has already received an RB, the RB index will be increased to attempt the next RB allocation. If the checked RB has not been allocated and the current user has not obtained an RB, the algorithm will allocate that RB to the user, updating the result matrix with the user's RB request value at that index. Once the allocation is done, the RB and user statuses are updated to reflect that they have been allocated, and the RB index is increased to proceed to the next allocation. After all users connected to the small cell are allocated RBs in turn, the result matrix that has been populated with the RB allocations is returned as the result of the algorithm, showing how the RBs have been distributed among the users connected to the small cell.

2.3. Simulation Parameter

In this research, the RB allocation process is carried out centrally, the RB allocation process is thoroughly to each SUE with a variation in the number of SUEs and is carried out in the K-Medoids clustering system model. The variation starts from 50 SUEs to 100 SUEs with an increase of 5 SUEs. All SUEs were allocated 40 RBs from each SBS. Each RB that has been used by one SUE, cannot be used again by another SUE.

Table 1. Simulation Parameter

Parameter	Value
Number of SUE	50, 55, 60, 65, …, 100
Number of RB	40
Frequency carrier	1800 MHz
System Bandwidth	10 MHz
Transmit power SBS	30 dBm
Gain Tx SBS	10 dB
Shadowing	Log normal model
Fading	Rayleigh model
Path loss model	Cost 231

In the simulations conducted, the "Number of SUEs" refers to the number of user equipment (UE) users connected to the Small Base Station (SBS). The variation in the number of users tested was from 50, 55, 60, 65, to 100. This analysis is important to understand how the performance of the communication system changes as the number of users increases.

Furthermore, the "Frequency Carrier" used is the 1800 MHz frequency. This frequency is one of the most frequently used bands in mobile communications, describing the centre frequency used for signal transmission. Another parameter described is the "System Bandwidth," which refers to the total frequency bandwidth available for data transmission, which in this case is 10 MHz.

SBS Transmit Power is the signal strength emitted by the SBS, with a value of 30 dBm (decibel milliwatts). SBS Tx Gain, which is the antenna gain of the SBS, is measured at 10 dB. This value indicates the effectiveness of the antenna in amplifying the transmitted signal.

Shadowing refers to the effect of physical obstructions such as buildings or trees that cause slow variations in the received signal strength. In this simulation, the model used to describe shadowing is the log normal model, which is important for predicting and analysing the effect of obstructions on signal reception in a given environment.

Fading is a phenomenon affecting wireless communication signals as they propagate through space. Fading can cause the signal's strength to decrease (attenuate) or fluctuate due to various obstacles in the environment, such as buildings, trees, or even atmospheric conditions. It typically results from the multipath propagation of radio signals whereby the received signal is the sum of multiple copies of the transmitted signal that have taken different paths to reach the receiver [10] .

The system model was built using the Cost-231 path loss model. Path loss is the weakening of the power of the information signal emitted by the transmitter antenna to the receiver antenna that occurs during the signal transmission process over a certain distance. Cost-231 path loss can be calculated using equation (1):

$$P_L = 46.3 + 33.9 \log_{10} f_c - 13.82 \log_{10} h_{Tx} - a(h_{Rx}) + (44.9 - 6.55 \log_{10}(h_{Tx})) \log_{10} D + C_{EN} \quad (1)$$

where f_c is the carrier frequency in MHz, h_{Tx} is the height of the SBS antenna in metres, $a(h_{Rx})$ is the correlation factor of the SUE antenna, D is the linear distance between the SBS antenna and the SUE antenna in km, and $C_{EN}$is an environment-dependent constant in dB. After getting the path loss value, you can calculate the received power with the following equation (2):

$$P_{Rx} = P_{Tx} - P_L - S - F + G_{SM} + G_{UE} \quad (2)$$

where P_{Tx} is the transmit power of the SBS antenna in dBm, P_L is the path loss in dB, S is shadowing, F is fading, G_{SM} is the SBS antenna gain in dB, and G_{UE} is the SUE antenna gain in dB.

After getting the received power value, it can calculate the Signal to Noise Ratio (SNR). SNR is the ratio between the strength of the desired signal and the strength of unwanted noise in a system or measurement. A higher SNR indicates better signal quality, because the desired signal is more dominant compared to noise. The following is the equation (3) for calculating the SNR value:

$$SNR = \frac{P_{Rx}}{T_N} \tag{3}$$

with P_{Rx} is received power and T_N is thermal noise.

3. RESULTS AND DISCUSSION

The results from the simulation are analyzed to evaluate the performance of each system. The analyzed performance parameters include data rate, total data rate, spectral efficiency, power efficiency, and fairness.

3.1. Data Rate

Data rate is the amount of data that can be transmitted in a certain period of time. There is a calculated data rate, which is the data rate on Small cell User Equipment (SUE). Equation (4) is used to calculate the value of the data rate on the SUE [11]

$$RS_i = B \cdot log_2(1 + SS_i) \tag{4}$$

where RS_i is the data rate at the i-th SUE, B is the bandwidth in Hz, and SS_i is the Signal to Noise Ratio (SNR) at the i-th SUE.

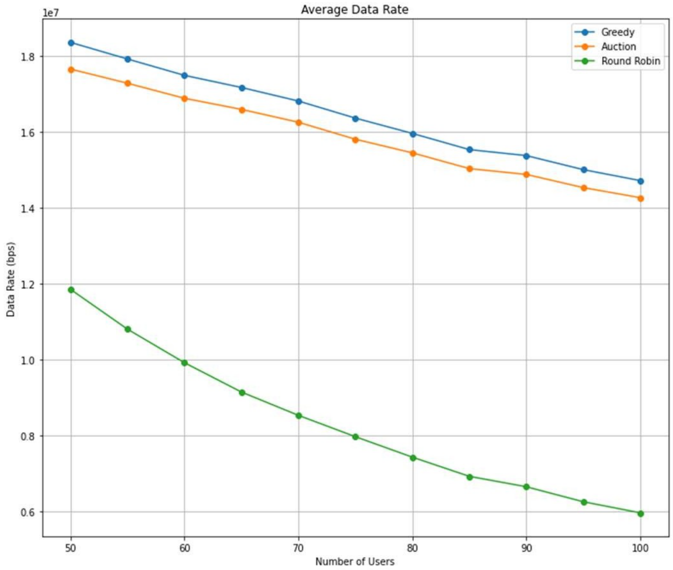

Figure 5. Result of data rate simulation

Table 2. Average data rate for each algorithm

Spesification	Greedy Algorithm	Auction Algorithm	Round Robin Algorithm
Data Rate (bps)	1.642×10^7	1.587×10^7	0.830×10^7

Figure 5 shows the results of data rate simulation using three different algorithms: Greedy, Auction, and Round Robin. The graph at the top illustrates the relationship between the number of users and the average data rate in bits per second (bps), with the horizontal axis representing the number of users varying from 50 to 100, and the vertical axis showing the average data rate on a scale of 10^7 bps. The blue curve representing the Greedy Algorithm shows a gradual decrease in the average data rate as the number of users increases from 50 to 100. The orange curve for the Auction Algorithm also shows a decrease in the average data rate, but it is slightly lower than that of Greedy. Meanwhile, the green curve for the Round Robin Algorithm shows a sharper decrease. Table 2 below the graph summarises the average data rate for each algorithm, with Greedy Algorithm having the highest average data rate of 1.642×10^7 bps, followed by Auction Algorithm at 1.587×10^7 bps, and Round Robin Algorithm with the lowest average data rate of 0.830×10^7 bps.

Greedy algorithm shows the best performance with a data rate difference of 0.055 × 10^7 bps higher than auction algorithm and 0.812 × 10^7 bps higher than round robin algorithm. Auction algorithm comes second with a difference of 0.757 × 10^7 bps higher than round robin algorithm. Greedy algorithm excels in allocating network resources for the highest data rate, while round robin algorithm shows the lowest efficiency in resource utilization.

3.2. Sum Rate

Sum rate is the sum of all data rates obtained by adding the data rate values in SUE. Equation (5) is used to calculate the sum rate value of the SUE [12].

$$RS_i = B \cdot log_2(1 + SS_i) \tag{5}$$

where R_{sum} is the sum rate, S is the number of SUEs, and RS_i is the data rate at the i-th SUE.

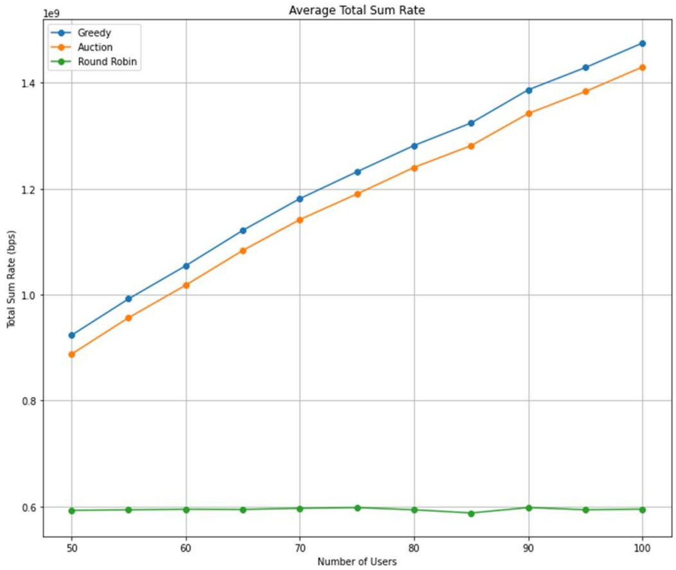

Figure 6. Result of sum rate simulation

Table 3. Average sum rate for each algorithm

Spesification	Greedy Algorithm	Auction Algorithm	Round Robin Algorithm
Sum Rate (bps)	1.218 × 10^9	1.177 × 10^9	0.594 × 10^9

Figure 6 shows the simulation results of the sum rate using three different algorithms: Greedy, Auction, and Round Robin. The graph at the top shows the relationship between the number of users and the average sum rate in bits per second (bps). The horizontal axis represents the number of users varying from 50 to 100, while the vertical axis shows the average sum rate in 10^9 bps. The blue curve representing the Greedy Algorithm shows consistent improvement when the number of users is 50 to 100. The orange curve for the Auction Algorithm also shows a similar but slightly lower improvement. In contrast, the green curve representing the Round Robin Algorithm shows an almost constant average sum rate, with no significant increase even as the number of users increases. Table 3 summarises the average sum rate for each algorithm: Greedy Algorithm has the highest average sum rate of 1.218 × 10^9 bps, followed by Auction Algorithm at 1.177 × 10^9 bps, and Round Robin Algorithm with the lowest average sum rate of 0.594 × 10^9 bps.

Greedy algorithm shows the best performance with sum rate difference of 0.041 × 10^9 bps higher than auction algorithm and 0.624 × 10^9 bps higher than round robin algorithm. Auction algorithm comes second with a difference of 0.583 × 10^9 bps higher than round robin algorithm. Greedy algorithm excels in allocating network resources to achieve the highest sum rate, while round robin algorithm shows the lowest efficiency in resource utilization.

3.3. Spectral Efficiency

Spectral efficiency is a measure of how efficiently a wireless communication system uses the available frequency spectrum. Spectral efficiency measures the total data rate that can be transmitted in a unit of time at a certain bandwidth unit. Equation (6) is used to calculate the spectral efficiency value [13].

$$S_{eff} = \frac{R_{sum}}{B \cdot RB} \tag{6}$$

S_{eff} with is the spectral efficiency, R_{sum} is the sum rate, B is the bandwidth in Hz, and RB is the resource block.

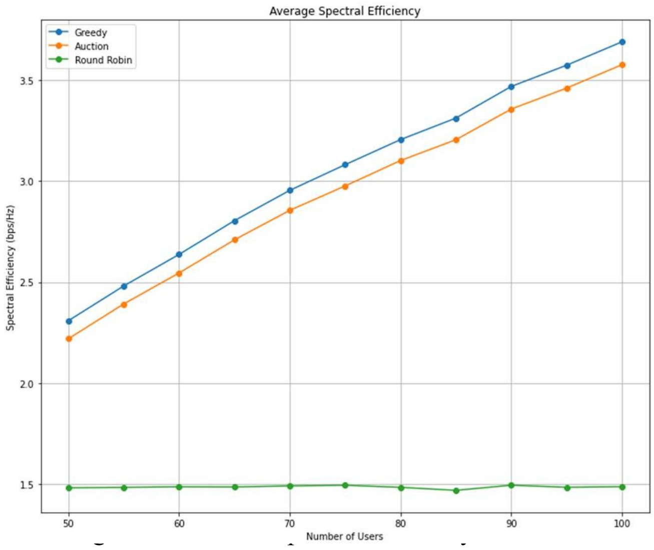

Table 4. Average spectral efficiency for each algorithm

Spesification	Greedy Algorithm	Auction Algorithm	Round Robin Algorithm
Spectral Efficiency (bps/Hz)	3.046	2.944	1.486

Figure 7 shows the simulation results of spectral efficiency using three algorithms: Greedy, Auction, and Round Robin. The graph at the top illustrates the relationship between the number of users (50 to 100) and the average spectral efficiency in bits per second per Hertz (bps/Hz). The blue curve for the Greedy Algorithm shows an increase in spectral efficiency as the number of users increases. The orange curve for the Auction Algorithm also shows a similar but slightly lower improvement. The green curve for the Round Robin Algorithm shows almost constant spectral efficiency, with no significant improvement. Table 4 below the graph summarises the average spectral efficiency: Greedy Algorithm has the highest efficiency at 3.046 bps/Hz, followed by Auction Algorithm at 2.944 bps/Hz, and Round Robin Algorithm the lowest with 1.486 bps/Hz.

In this research, greedy algorithm shows the best performance with spectral efficiency difference of 0.102 bps/Hz higher than auction algorithm and 1.560 bps/Hz higher than round robin algorithm. Auction algorithm comes second with a difference of 1.458 bps/Hz higher than round robin algorithm. Greedy algorithm excels in spectrum efficiency, which means it is more effective in using the frequency spectrum to transmit data. The auction algorithm also performed well although slightly lower than the greedy algorithm, while the round robin algorithm had the lowest spectral efficiency, indicating that it was less effective in using the frequency spectrum.

3.4. Power Efficiency

Power efficiency is the data rate obtained per unit of power of one watt. Equation (7) is used to calculate the value of power efficiency [14]

$$P_{eff} = \frac{R_{sum}}{(S \cdot P_{SU})} \tag{7}$$

where P_{eff} is the power efficiency, R_{sum} is the sum rate, S is the number of SUEs, and P_{SU} is the transmit power of the SUE.

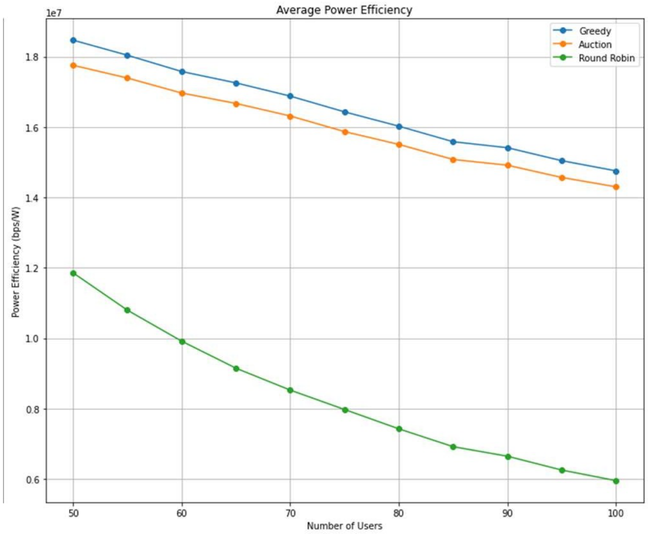

Figure 8. Result of power efficiency simulation

Table 5. Average power efficiency for each algorithm

Spesification	Greedy Algorithm	Auction Algorithm	Round Robin Algorithm
Power Efficiency (bps/W)	1.650×10^7	1.594×10^7	0.831×10^7

Figure 8 shows the simulated power efficiency results of three algorithms: Greedy, Auction, and Round Robin. The graph at the top displays the relationship between the number of users (50 to 100) and the average power efficiency in bits per second per Watt (bps/W). The blue curve representing the Greedy Algorithm shows a decrease in power efficiency as the number of users increases. The orange curve for the Auction Algorithm also shows a similar decrease. Meanwhile, the green curve for the Round Robin Algorithm shows a sharper drop. Table 5 below the graph summarises the average power efficiency, with Greedy Algorithm having the highest efficiency of 1.650×10^7 bps/W, followed by Auction Algorithm at 1.594×10^7 bps/W, and Round Robin Algorithm with the lowest efficiency of 0.831×10^7 bps/W.

In this research, greedy algorithm shows the best performance with power efficiency difference of 0.056×10^7 bps/W higher than auction algorithm and 0.819×10^7 bps/W higher than round robin algorithm. Auction algorithm comes second with a difference of 0.76×10^7 bps/W higher than round robin algorithm. Greedy algorithm excels in power efficiency, showing more effective use of power to achieve high data rate. The auction algorithm also performed well although slightly lower than the greedy algorithm, while the round robin algorithm had the lowest power efficiency, indicating a significant need for improvement in power usage.

3.5. Fairness

Fairness is an index to assess the fairness received by each user in obtaining rate data. In this research, the calculation of fairness uses Jain's Fairness Index. Equation (8) is used to calculate the fairness value [15].

$$F_{air} = \frac{(\sum_{i=1}^{S} RS_i)^2}{n \cdot \sum_{i=1}^{S} (RS_i)^2} \quad (8)$$

where F_{air} is fairness, S is the number of SUEs, and RS_i is the data rate at the i-th SUE, and n is the number of users in the system.

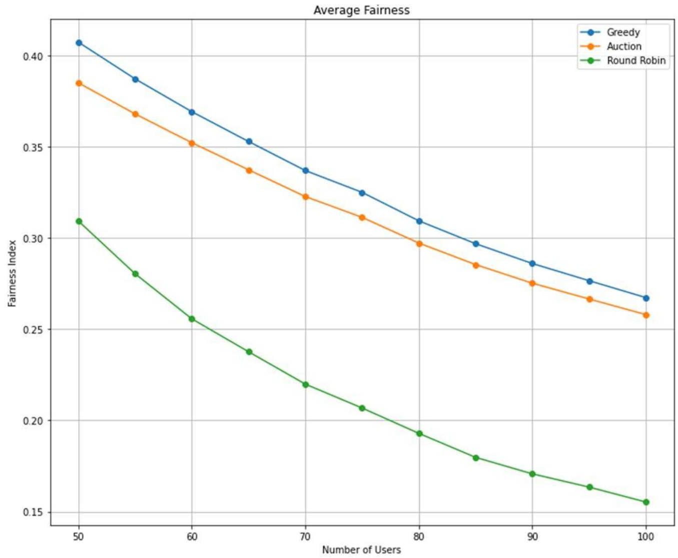

Figure 9. Result of fairness simulation

Table 6. Average fairness for each algorithm

Spesification	Greedy Algorithm	Auction Algorithm	Round Robin Algorithm
Fairness	0.329	0.314	0.216

Figure 9 shows the fairness simulation results using three algorithms: Greedy, Auction, and Round Robin. The graph at the top illustrates the relationship between the number of users (50 to 100) and the average fairness index. The blue curve for the Greedy Algorithm shows a decrease as the number of users increases. The orange curve for the Auction Algorithm also shows a similar decrease. The green curve for the Round Robin Algorithm shows a sharper decline. Table 6 below the graph summarises the average fairness index: Greedy Algorithm has the highest average fairness index of 0.329, followed by Auction Algorithm at 0.314, and Round Robin Algorithm with the lowest fairness index of 0.216.

Greedy algorithm shows the best performance with fairness difference 0.015 higher than Auction Algorithm and 0.113 higher than round robin algorithm. The Auction algorithm came in second with a difference of 0.098 higher than the round robin algorithm. Greedy algorithm excels in resource allocation fairness, showing a more even distribution among users. Auction algorithm also performs well although slightly lower than greedy algorithm, while round robin algorithm has the lowest fairness, indicating that it is less effective in providing fair resource allocation.

4. CONCLUSION

In this study, RB allocation scheme on a two-tier HetNet model, consisting of one MBS and four SBS, is evaluated using three algorithms: greedy, auction, and round robin. The simulation involves varying the number of SUE from 50 to 100, allocating 40 RB per SBS, and using K-Medoids clustering for data clustering. The performance comparison shows that the greedy algorithm consistently outperforms the auction and round robin algorithms across various parameters. Where the greedy algorithm has an average data rate of 1.642 $\times 10^7$ bps, the average sum rate of 1.218 $\times 10^9$ bps, the average spectral efficiency of 3.046 bps/Hz, the average power efficiency of 1.650 $\times 10^7$ bps/W, and the average fairness of 0.329. These results show that the Greedy Algorithm is superior in optimizing resource allocation to maximize data rate, spectral efficiency, power efficiency, and fairness, making it the most effective approach for this HetNet system model.

REFERENCES

[1] M. Ibrahim, U. S. Hashmi, M. Nabeel, A. Imran, and S. Ekin, “Embracing Complexity: Agent-Based Modeling for HetNets Design and Optimization via Concurrent Reinforcement Learning Algorithms,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4042–4062, Dec. 2021, doi: 10.1109/TNSM.2021.3121282.
[2] W. Bao and B. Liang, “Radio resource allocation in heterogeneous wireless networks: A spatial-temporal perspective,” *Proceedings - IEEE INFOCOM*, vol. 26, pp. 334–342, Aug. 2015, doi: 10.1109/INFOCOM.2015.7218398.
[3] S. Tripathi, C. Puligheddu, C. F. Chiasserini, and F. Mungari, “A Context-Aware Radio Resource Management in Heterogeneous Virtual RANs,” *IEEE Trans Cogn Commun Netw*, vol. 8, no. 1, pp. 321–334, Mar. 2022, doi: 10.1109/TCCN.2021.3115098.

[4] D. Yuan, X. Chen, D. Hu, and S. Zhang, "Self-Adaptive Game Theoretic Approaches for Resource Allocation in SC-FDMA Based Heterogeneous Networks," *IEEE Trans Veh Technol*, vol. 71, no. 11, pp. 12057–12072, Nov. 2022, doi: 10.1109/TVT.2022.3189659.

[5] L. Feng, Z. Yang, Y. Yang, X. Que, and K. Zhang, "Smart Mode Selection Using Online Reinforcement Learning for VR Broadband Broadcasting in D2D Assisted 5G HetNets," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 600–611, Jun. 2020, doi: 10.1109/TBC.2020.2977577.

[6] J. Oyelade *et al.*, "Data Clustering: Algorithms and Its Applications," *Proceedings - 2019 19th International Conference on Computational Science and Its Applications, ICCSA 2019*, pp. 71–81, Jul. 2019, doi: 10.1109/ICCSA.2019.000-1.

[7] M. K. Khan, S. Mahmud Ahmed, S. Sarker, and M. H. A. Khan, "K-Cosine-Medoids Clustering Algorithm," *2021 5th International Conference on Electrical Information and Communication Technology, EICT 2021*, 2021, doi: 10.1109/EICT54103.2021.9733540.

[8] A. Sabzi, Y. Farjami, and M. Zihayat, "An improved fuzzy K-medoids clustering algorithm with optimized number of clusters," *Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems, HIS 2011*, pp. 206–210, 2011, doi: 10.1109/HIS.2011.6122106.

[9] A. Radovanovic, X. Ye, J. V. Milanovic, N. Milosavljevic, and R. Storchi, "Application of the k-medoids partitioning algorithm for clustering of time series data," *IEEE PES Innovative Smart Grid Technologies Conference Europe*, vol. 2020-October, pp. 645–649, Oct. 2020, doi: 10.1109/ISGT-EUROPE47291.2020.9248796.

[10] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, "Self-organization in small cell networks: A reinforcement learning approach," *IEEE Trans Wirel Commun*, vol. 12, no. 7, pp. 3202–3212, 2013, doi: 10.1109/TWC.2013.060513.120959.

[11] "Data Communications and Networking - Behrouz A. Forouzan - Google Books." Accessed: Jun. 29, 2024. [Online]. Available: https://books.google.co.id/books?hl=en&lr=&id=bwUNZvJbEeQC&oi=fnd&pg=PR3&dq=B.+A.+Forouzan,+Data+communications+and+networking.+Huga+Media,+2007.&ots=qvmZJaPrqH&sig=ouQUY9ooxOuwHlZd28O-9DGsbrY&redir_esc=y#v=onepage&q=B.%20A.%20Forouzan%2C%20Data%20communications%20and%20networking.%20Huga%20Media%2C%202007.&f=false

[12] "IEEE Xplore Full-Text PDF:" Accessed: Jun. 29, 2024. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5997338

[13] J. Iqbal, M. A. Iqbal, A. Ahmad, M. Khan, A. Qamar, and K. Han, "Comparison of Spectral Efficiency Techniques in Device-to-Device Communication for 5G," *IEEE Access*, vol. 7, pp. 57440–57449, 2019, doi: 10.1109/ACCESS.2019.2914486.

[14] D. T. Huynh, X. Wang, T. Q. Duong, N. S. Vo, and M. Chen, "Social-aware energy efficiency optimization for device-to-device communications in 5G networks," *Comput Commun*, vol. 120, pp. 102–111, May 2018, doi: 10.1016/J.COMCOM.2018.02.008.

[15] F. Boabang, H. H. Nguyen, Q. V. Pham, and W. J. Hwang, "Network-Assisted Distributed Fairness-Aware Interference Coordination for Device-to-Device Communication Underlaid Cellular Networks," *Mobile Information Systems*, vol. 2017, no. 1, p. 1821084, Jan. 2017, doi: 10.1155/2017/1821084.