Optimization of Resource Allocation in 5G MIMO Networks using Linear Assignment Algorithms

Nikolaos Prodromos[1], Damianos Diasakos[1], Vasileios Kokkinos[1], Apostolos Gkamas[2], Christos Bouras[1,*], Philippos Pouyioutas[3]

[1]*Computer Engineering and Informatics Department, University of Patras, Patras, Greece*
[2]*Department of Chemistry, University of Ioannina, Ioannina, Greece*
[3]*Computer Science Department, University of Nicosia, Nicosia, Cyprus*
up1072549@upnet.gr, up1084632@upnet.gr, kokkinos@cti.gr, gkamas@uoi.gr, bouras@upatras.gr, pouyioutas.p@unic.ac.cy

Abstract—Massive Multiple Input Multiple Output (MIMO) is an essential technology that can significantly improve the performance of 5G wireless networks by using multiple antennas in base stations, improving coverage, reducing interference, and increasing data throughput. In this comprehensive study, we propose and analyze advanced optimization techniques for resource allocation in 5G MIMO networks, focusing on three distinct approaches: simple sorting, Hungarian Algorithm, and Minimum Cost Flow Algorithm. Simulations are performed using the publicly available DeepMIMO dataset, where we evaluate each method under both static and dynamic scenarios, aiming to optimize bandwidth distribution and minimize power consumption. A key contribution of this work is the formulation and comparative evaluation of the resource allocation problem as an assignment-based model, allowing the examined methods to be compared under common DeepMIMO-based static and dynamic scenarios. The technical contribution of this work lies in the common assignment-based formulation and comparative evaluation of simple sorting, Hungarian, and Minimum Cost Flow allocation methods under the same DeepMIMO-based static and dynamic 5G MIMO scenarios. Our comparative analysis shows that, under the evaluated DeepMIMO-based scenarios, the examined assignment-based methods exhibit different trade-offs in throughput, energy-consumption-related performance, bandwidth utilization, and adaptability to varying user demands, offering useful insights for 5G MIMO resource allocation studies.

Keywords—5G Networks, Resource Allocation, Bandwidth Allocation, Energy Efficiency, MIMO Networks, Linear Assignment Algorithms

1. Introduction

The relentless growth in wireless data consumption, coupled with the proliferation of Internet of Things (IoT) devices, has spurred the development of 5G technology to cater to the escalating demands of diverse applications. Multiple-Input Multiple-Output (MIMO) systems, which employ multiple antennas at both the transmitter and receiver ends, have proven instrumental in realizing the ambitious goals of 5G networks, such as high data rates and massive device connectivity [1]. Building upon the successes of traditional MIMO, Massive MIMO (MM) systems integrate advanced techniques to unravel complex spatial correlations and channel behaviors, enabling the exploitation of multiuser diversity and achieving enhanced spectral efficiency. This advancement comes with new challenges, particularly in the effective allocation of limited network resources.

In the context of 5G MIMO environments, efficient resource allocation becomes paramount. Unlike traditional homogeneous data requirements, 5G scenarios encompass a wide range of applications, each with unique and often dynamic bandwidth demands, such as Ultra-Reliable Low Latency Communications (URLLC) and Enhanced Mobile Broadband (eMBB). Consequently, the allocation of users to base station antennas must be intelligently managed to optimize network performance while accommodating diverse user needs. Moreover, base stations are resource-constrained, possessing finite bandwidth and a limit to the number of users they can concurrently serve, further complicating the resource allocation challenge.

Additionally, base stations themselves are resource constrained, have limited bandwidth, and a limited number of users they can serve simultaneously. This article addresses the complex challenge of resource allocation in a 5G MIMO environment with heterogeneous bandwidth requirements.

In this manuscript, a comprehensive examination of network resource allocation strategies is presented through two distinct studies. Initially, the optimization of resource allocation was studied, employing fundamental algorithms including initial sorting and after that, two algorithms (Hungarian and Minimum Cost Flow) in order to deal with optimization of resource allocation problem as a linear assignment problem. After that, dynamic elements are introduced by incorporating bandwidth thresholds for base stations. This dynamic approach necessitates slight modifications to the algorithms that were used in our first research study [2], ensuring adaptability to changing network conditions while maintaining efficient resource allocation. By merging the above approaches, a holistic perspective is offered on network resource management, encompassing both static and dynamic scenarios to address the complexities of modern network environments effectively.

[*] **Corresponding author:**
Christos Bouras

In addition to Massive MIMO resource allocation, 5G networks must also support massive Machine-Type Communications (mMTC), where a very large number of low-power and sporadically active devices compete for limited access and radio resources. In this context, recent studies have investigated random-access optimization, PRACH traffic-load estimation, contention-based resource allocation, Non-Orthogonal Multiple Access (NOMA), power-domain multiplexing, and user clustering techniques to improve connectivity density and resource efficiency [3], [4]. These works highlight the broader importance of efficient resource allocation in 5G networks. However, the present manuscript focuses specifically on assignment-based user-to-base station allocation in DeepMIMO-based 5G MIMO scenarios, where simple allocation, Hungarian assignment, and Minimum Cost Flow are compared under common simulation assumptions.

The Hungarian algorithm [5] and Minimum Cost Flow algorithm [6] are well-established optimization methods and have already been explored in the literature for several network optimization and resource allocation tasks [7,8]. Therefore, the novelty of this study does not lie in the development of new algorithms, but in the formulation, adaptation, and comparative evaluation of these methods for dynamic resource allocation within MIMO systems. Unlike prior research [7,8], which often treats these algorithms in static settings or with simplified assumptions, our work evaluates their adaptability under varying conditions using the DeepMIMO dataset, simulating complex 5G MIMO environments. Our solution addresses the dynamic nature of 5G MIMO environments by integrating the distinct bandwidth demands of diverse applications with the limitations of base stations, thereby optimizing bandwidth distribution and ensuring equitable and efficient resource allocation. The subsequent sections delve into the technical details of the proposed methodology, highlighting the integration of optimization techniques, and network performance evaluation. Our comparative analysis offers practical insights for network operators by demonstrating the efficiency and scalability of these algorithms under varying simulated 5G MIMO conditions. The results highlight how each algorithm can be best utilized, providing valuable guidance for practitioners aiming to enhance network performance and user experience. Dynamic bandwidth allocation in 5G MIMO networks has been extensively studied, with numerous works proposing strategies for user-to-base station assignment [9-14]. While these studies have contributed valuable insights, they often focus on specific approaches without a direct comparison across different algorithmic techniques. Our study addresses this by providing a comparative evaluation of three distinct algorithms: the Hungarian algorithm, the Minimum Cost Flow algorithm, and a simple sorting method. Using the DeepMIMO dataset to simulate realistic 5G MIMO conditions, we demonstrate how each algorithm performs under varying scenarios, offering new insights into their efficiency, scalability, and practical application in dynamic network environments.

The exploration of more advanced strategies is essential to address these challenges comprehensively. Researchers are now delving into cutting-edge technologies, including intelligent data-driven models, to devise dynamic bandwidth allocation schemes. Our research places a strong emphasis on adaptability, ensuring that the chosen allocation algorithms can respond in real-time to shifting demands, such as fluctuating user populations and evolving application requirements. It differs from the previous ones in the field due to our approach which focuses on adaptability and maintaining peak performance even in the face of fluctuating user populations and evolving application requirements. The Minimum Cost Flow algorithm excels in dynamically optimizing bandwidth allocation based on user needs and base station capacities. Furthermore, our approach enhances resource utilization within 5G MIMO networks, intelligently balancing user-to-base station assignments to achieve a more equitable distribution of resources. Through rigorous experimentation and analysis, the comparative performance of the examined approaches is evaluated in terms of allocation strategy, bandwidth allocation, and dynamic bandwidth management. This article examines assignment-based solutions for dynamic resource allocation, harnessing the power of both traditional and advanced algorithms to manage the allocation of users to base station antennas efficiently.

This article consists of 2 different studies which utilize the DeepMIMO [15] environment and scenario but differ in the parameters used for testing. The first approach examines algorithms for allocating users to base stations based on their Signal-to-Noise Ratio (SNR) values. On the other hand, the second approach places a strong emphasis on adaptability, ensuring that the chosen allocation algorithms can respond in real-time to shifting demands, such as fluctuating user populations and evolving application requirements. The research study mentioned above is considered to be more complex, since it uses parameters such as downstream and upstream requirements for each user and a bandwidth capacity for each base station. Initial results of this research have been presented in [2] and [16]. In this journal a more detailed evaluation of results is presented by examining the proposed algorithms in more topologies and investigating power consumption for each base station during the evaluation of the proposed algorithms.

Motivated by the increasing need for efficient user-to-base station association in dense 5G MIMO networks, the main research question addressed in this work is how different assignment-based optimization methods behave when signal quality, bandwidth demand, and base-station capacity constraints are jointly considered under the same simulation framework. More specifically, this study investigates whether established assignment-based algorithms can improve resource allocation compared with a simple SNR-based allocation strategy, and what trade-offs arise in terms of bandwidth utilization, overflow users, throughput, computational complexity, and energy-consumption-related performance.

The manuscript is structured as follows: Section II presents the related work and Section III presents the proposed algorithms for user allocation in 5G MIMO networks, including the simple sorting, Hungarian, and minimum cost flow algorithms. Section IV delves into the DeepMIMO environment, detailing the simulation setup and parameters used to generate the dataset. Following this in Section V, a comprehensive performance comparison of these algorithms is conducted. Section VI focuses on energy consumption analysis, evaluating the power efficiency of the algorithms. In Section VII, the conclusions based on the findings of our study are presented. Finally in Section VIII, future directions for research in optimizing 5G MIMO networks are proposed.

2. Related Work

In recent years, significant research efforts have been devoted to resource allocation techniques in the context of 5G wireless networks, with particular focus on enhancing efficiency and maximizing throughput. [17] presents an innovative approach to resource allocation that leverages multi-agent parameterized deep reinforcement learning to optimize user association and power allocation in ultra-dense heterogeneous networks. The proposed approach addresses the complexities of joint optimization in large-scale networks, achieving superior energy efficiency and Quality of Service (QoS) compared to traditional methods. This study highlights the potential of reinforcement learning to handle the dynamic resource allocation needs of 5G environments, balancing user demands and network constraints effectively.

Another recent study [18] explores energy-efficient resource allocation through slicing and beam selection in millimeter-wave (mmWave) downlink networks. The authors propose an energy efficiency maximization framework to manage resource block allocation, beam selection, and transmit power optimization in a Radio Access Network (RAN) slicing context. By tailoring resource allocation to serve heterogeneous services efficiently, this approach enhances system-level performance by balancing the competing demands of diverse user requirements. Simulation results demonstrate significant gains in energy efficiency and resource utilization over baseline methods.

Moreover, [19] explored joint resource allocation strategies in a Time-Division Duplex (TDD)-based Cell-Free Massive MIMO (CFMM) system, where they address pilot contamination, a significant challenge in TDD systems through innovative pilot assignment and Access Point (AP) selection algorithms. Their methodology incorporates user-distance-ordering-based pilot assignment and a modified max–min power control scheme, which collectively enhance the performance by reducing interference and ensuring fair resource distribution among users.

The abovementioned works have collectively advanced resource allocation techniques in 5G networks, addressing key challenges such as optimal resource utilization, computational complexity, and energy efficiency. However, they mainly focus on specific optimization directions. For example, [17] focuses on learning-based user association and power allocation in ultra-dense heterogeneous networks, [18] studies energy-efficient resource allocation through slicing and beam selection in mmWave networks, while [19] investigates pilot assignment, AP selection, and power control in CFMM.

Recent studies have also investigated resource allocation in 5G and beyond wireless networks from the perspective of Age of Information (AoI), learning-based optimization, and energy-aware communication. In [20], AoI minimization in heterogeneous MEC networks is addressed through a federated learning-assisted hybrid DRL and convex optimization approach. In [21], throughput maximization under an AoI constraint is studied for energy-harvesting D2D-enabled cellular networks using an MSRA-TD3 approach. In [22], distributed DDPG-based resource allocation is investigated for AoI minimization in mobile wireless-powered Internet of Things systems. These studies demonstrate the importance of learning-based and AoI-aware resource allocation; however, they focus on different system models and optimization objectives, while the present manuscript focuses on assignment-based user-to-base station allocation under common DeepMIMO-based simulation assumptions.

Table 1 Comparison of representative resource allocation studies and the present work.

Work	*Main focus*	*Methodology*	*Difference from the present work*
[17]	User association and power allocation in ultra-dense heterogeneous networks	Multi-agent deep reinforcement learning	Focuses on learning-based joint optimization and does not compare assignment-based Hungarian, Minimum Cost Flow, and simple allocation under the same DeepMIMO setup
[18]	Energy-efficient resource allocation in 5G mmWave networks	RAN slicing, beam selection, and transmit power optimization	Focuses on slicing and beam selection rather than assignment-based user-to-base station allocation
[19]	Resource allocation in Cell-Free Massive MIMO systems	Pilot assignment, AP selection, and power control	Considers a different network architecture and optimization objective
[20]	AoI minimization in heterogeneous MEC networks	Federated learning-assisted hybrid DRL and convex optimization	Focuses on AoI-aware MEC resource allocation rather than DeepMIMO-based assignment comparison
[21]	Throughput maximization with AoI constraints in D2D-enabled cellular networks	MSRA-TD3 learning-based optimization	Studies energy-harvesting D2D communication with AoI constraints, which differs from the assignment-based 5G MIMO setting
[22]	AoI minimization in mobile wireless-powered IoT	Distributed DDPG-based resource allocation	Focuses on IoT and wireless-powered networks rather than user-to-base station assignment in DeepMIMO scenarios
This work	User-to-base station allocation in 5G MIMO networks	Simple sorting, Hungarian algorithm, and Minimum Cost Flow	Provides a comparative evaluation of assignment-based methods under common DeepMIMO-based static and dynamic scenarios

This comparison further clarifies that the present manuscript does not claim to introduce a new optimization algorithm. Instead, its contribution lies in the formulation, adaptation, and comparative evaluation of established assignment-based methods under common DeepMIMO-based assumptions and performance metrics.

In comparison, the present study focuses on the comparative evaluation of three assignment-based resource allocation approaches, namely simple sorting, the Hungarian algorithm, and the Minimum Cost Flow algorithm, under the same DeepMIMO-based 5G MIMO simulation framework. The comparison considers both static and dynamic scenarios, incorporating upstream and downstream user demands, base-station bandwidth limitations, user distribution, throughput, overflow behavior, and energy-consumption-related analysis. This comparative summary clarifies that the main distinction of this work lies in evaluating these methods under common simulation assumptions, allowing their trade-offs in allocation efficiency, computational complexity, resource utilization, and service quality to be more clearly identified.

Because the reviewed studies employ different system models, objectives, datasets, and baselines, a direct numerical comparison with all current MIMO resource-allocation approaches is not straightforward; therefore, the evaluation in this manuscript is limited to the considered assignment-based methods under common DeepMIMO-based assumptions and performance metrics.

3. Proposed Algorithms

In this proposed algorithmic framework, a comprehensive set of strategies is presented aimed at optimizing resource allocation within diverse network scenarios featuring multiple active base stations and varying user requirements. Initially, a sorting algorithm is introduced tailored for environments with a single base station, facilitating efficient resource allocation based on specific criteria. Following this, a reordering algorithm is presented designed to address the complexities of scenarios featuring multiple active base stations, ensuring optimal resource distribution. Subsequently, the Hungarian algorithm is integrated, a powerful combinatorial optimization technique adept at solving assignment problems, to further enhance resource allocation efficiency. Additionally, the Minimum Cost Flow algorithm is introduced, which minimizes strategically resource allocation costs while simultaneously meeting capacity constraints and user demands. These algorithms, previously validated through simulations, are adapted to dynamic scenarios, effectively incorporating bandwidth requirements for each base station. The pseudocode descriptions are included to clarify the implementation logic of each allocation method and to support reproducibility; however, the contribution of the manuscript lies in the common formulation, adaptation, and comparative evaluation of the examined assignment-based methods rather than in the pseudocode itself. The main idea behind the examined algorithms is to compare a simple SNR-driven allocation baseline with two established assignment-based optimization methods, highlighting their strengths in terms of implementation simplicity, balanced user distribution, bandwidth utilization, and capacity-aware allocation.

Before presenting the examined algorithms, the resource allocation problem can be described through a common assignment-based formulation. Let U denote the set of users and B denote the set of available base stations. The binary decision variable $x_{u,\beta}$ indicates whether user u is assigned to base station b, where $x_{u,\beta} = 1$ if the assignment is selected and $x_{u,\beta} = 0$ otherwise. Each user is associated with a bandwidth demand d_u, while each base station b has an available bandwidth capacity C_β. The term $q_{u,\beta}$ denotes the allocation score between user u and base station b, which may depend on SNR, pathloss, bandwidth demand, or a weighted combination of these quantities depending on the examined algorithm.

The general assignment objective can be expressed as:

Maximize:

$$\Sigma_u \Sigma_\beta q_{u,\beta} x_{u,\beta} \tag{1}$$

Subject to:

$\Sigma_\beta x_{u,\beta} \leq 1$, for every user u

$\Sigma_u d_u x_{u,\beta} \leq C_\beta$, for every base station b

$x_{u,\beta} \in \{0, 1\}$, for every user u and base station b

The first constraint ensures that each user is assigned to at most one base station, while the second constraint ensures that the total bandwidth demand assigned to each base station does not exceed its available capacity. Users that cannot be assigned without violating the capacity constraints are treated as overflow users. The examined simple allocation, Hungarian, and Minimum Cost Flow algorithms can therefore be interpreted as different strategies for solving or approximating this assignment-based resource allocation problem under the same simulation assumptions.

3.1. Algorithm 1 - One Base Station Active

The first algorithm focuses on user assignment to a single base station. The fundamental goal is to allocate users based on their pathloss values, which reflect the signal quality they experience. To achieve this, a sorting algorithm is utilized so that the users' pathloss values are organized in ascending order. By doing so, the users with the lowest pathloss can be identified, indicating stronger signal quality. These users are prioritized for assignment to the base station.

Algorithm 1 – Storing the pathloss for each user to an array of structs

```
counter=0;
for bs = 1:numBS
      pathloss_bs = sorted_pathloss_bs_ue([sorted_pathloss_bs_ue.basestation] == bs);
      usernumber = [pathloss_bs.usernumber];
      basestation = [pathloss_bs.basestation];
      pathloss = [pathloss_bs.pathloss];
      for i = 1:min(num_tx_antennas, length(usernumber))
            counter = counter + 1;
            success_pathloss(counter) = struct('usernumber', usernumber(i), 'basestation', basestation(i), 'pathloss', pathloss(i));
      end
end

counter = 0;
for bs = 1:numBS
      pathloss_bs = pathloss_bs_ue([pathloss_bs_ue.basestation] == bs);
      usernumber = [pathloss_bs.usernumber];
      basestation = [pathloss_bs.basestation];
      pathloss = [pathloss_bs.pathloss];
      [~, index] = sort(pathloss, 'ascend');
      sorted_usernumber = usernumber(index);
```

```
        sorted_basestation = basestation(index);
        sorted_pathloss = pathloss(index);
        for i = 1:length(sorted_usernumber)
            counter = counter + 1;
            sorted_pathloss_bs_ue(counter) = struct('usernumber', sorted_usernumber(i), 'basestation', sorted_basestation(i), 'pathloss', sorted_pathloss(i));
        end
    end
```

The algorithm for assigning users to single base stations involves several steps. First, it creates an array to store user information, including user number, base station, and pathloss value. Next, the users are grouped by base station and sorted by ascending pathloss value. The sorted pathloss index is then used to adjust user and base station information accordingly. Finally, the algorithm creates a new array and fills it with sorted values to ensure optimal user mapping based on path loss. The time complexity of this approach is primarily determined by the sorting operation, which has a complexity of $O(nlogn)$ for each base station, where n is the number of users. Since the process is repeated across m base stations, the overall time complexity is $O(m * n * logn)$. By implementing this algorithm, users can be effectively assigned to the single active base station based on their pathloss values. This approach ensures a fair distribution of users while considering signal quality, ultimately optimizing the performance of the 5G network.

3.2. Algorithm 2 - Many Base Stations Active

To further our research, resource allocation strategies are investigated when multiple base stations are simultaneously active. Algorithm 2 assigns users to the base station that provides the highest SNR while respecting the capacity limit of each base station. The algorithm first stores user information and keeps the best SNR-based association for each user. Users are then sorted by SNR for each base station, and if a base station exceeds its predefined capacity, the excess users are moved to an overflow structure and reassigned where possible. In the dynamic case, SNR and bandwidth requirements are combined through weighted scores, allowing the allocation to consider both signal quality and user demand. For readability, the pseudocode summarizes the main allocation operations and omits low-level implementation details, such as repeated structure-field updates. The initial user-checking step has complexity $O(n)$ where n is the total number of users. Sorting users by SNR values for each base station adds an $O(nlogn)$ complexity, primarily due to the sorting operation. Since this process is repeated for m base stations, the overall time complexity can be approximated as $O(m * n * logn)$. Algorithm 2 is retained as a simple and interpretable baseline, since it directly assigns users according to their strongest SNR values; however, its limitations are also discussed, particularly its tendency to create unbalanced base-station loading and overflow users under high-density scenarios.

Algorithm 2 – Simple Dynamic Allocation

```
for ii = 1:length(sorted_pathloss_bs_ue)
  user = sorted_pathloss_bs_ue[ii];
  idx = find_matching_user(single_users, user.usernumber);

  if is_empty(idx)
    append_user(single_users, user);
  else
    if user.SNR > get_max_SNR(single_users, idx)
      max_idx = get_max_SNR_index(single_users, idx);
      update_user(single_users, idx[max_idx], user);
    end
  end
end

for bs = 1:numBS
  pathloss_bs = filter_single_sorted_pathloss_bs_ue(single_sorted_pathloss_bs_ue, bs);

  if length(pathloss_bs) > capacity
    for i = 1:capacity
      append_to_success_pathloss(success_pathloss, pathloss_bs[i]);
    end

    for i = (capacity+1):length(pathloss_bs)
      append_to_bs_overflow(bs_overflow, pathloss_bs[i]);
    end

  else
    for i = 1:length(pathloss_bs)
      append_to_success_pathloss(success_pathloss, pathloss_bs[i]);
    end
  end
end

for user = 1:length(bs1_overflow)
  user_num = bs1_overflow[user].usernumber;
  user_pathloss = filter_sorted_pathloss_bs_ue(sorted_pathloss_bs_ue, user_num);
  [~, idx] = sort_pathloss_by_SNR_descending(user_pathloss);

  for z = 1:length(idx)
    number_bs = 0;

    for z1 = 1:length(success_pathloss)
      if success_pathloss[z1].basestation == idx[z]
        number_bs = number_bs + 1;
```

```
            if number_bs < capacity
                append_to_success_pathloss(success_pathloss, user_pathloss([user_pathloss.basestation] == idx[z]));
                break;
            end
        end
    end
  end
end
    // Normalize SNR and bandwidth requirements, then calculate combined score for each user and base station
    for i = 1 to numUsers:
        for currentBaseStation = 1 to bs:
            userIndex = (currentBaseStation - 1) * numUsers + i
            normalizedSNR = Normalize(dynamic_pathloss_bs_ue(userIndex).SNR)
            normalizedBandwidth = Normalize(dynamic_pathloss_bs_ue(userIndex).Downstream)
            combinedScore = weightSNR * normalizedSNR + weightBandwidth * normalizedBandwidth
            combinedCostMatrix[i][currentBaseStation] = combinedScore

    // Allocate users to base stations based on combined scores and available bandwidth
    for i = 1 to numUsers:
        bestBaseStation = 0
        sortedBaseStations = SortIndices(combinedCostMatrix[i]) descending
        for j = 1 to numBS:
            currentBaseStation = sortedBaseStations[j]
            userIndex = (currentBaseStation - 1) * numUsers + i
            userRequirementInHZ = dynamic_pathloss_bs_ue(userIndex).Downstream / log2(1 + 10^(dynamic_pathloss_bs_ue(userIndex).SNR / 10))

            if allocatedCapacity_bps[currentBaseStation] + userRequirementInHZ <= bscapacity:
                allocatedCapacity_bps[currentBaseStation] += userRequirementInHZ

                // Update allocatedStruct with user information
                allocatedStruct.Add()
        break // Exit loop after allocation
```

3.3. Algorithm 3 - Hungarian Algorithm

The third algorithm serves as an effective solution for addressing linear assignment problems. These problems involve the allocation of rows to columns in a manner that each row is assigned to a column while minimizing the total cost of these assignments. To apply this algorithm to our resource allocation problem, it must be transformed into a linear assignment problem. Leveraging the Hungarian algorithm, which uniquely matches each row to a column based on their associated costs, the most suitable base station for each user to connect to is determined. By utilizing the $M = matchpairs(Cost, CostUnmatched)$ MATLAB function [19] as the Hungarian Algorithm implementation, the linear assignment problem is solved, considering both the rows and columns of the 'Cost' matrix. The 'CostUnmatched' parameter provides the cost associated with not assigning each row to a column and not having a row allocated to each column.

When it comes to the dynamic allocation case, the algorithm normalizes the SNR and bandwidth requirements for each user. It then finds the combined score for each user, using the normalized SNR and the normalized bandwidth, multiplying each by its corresponding weight. The weights are 0.6 for SNR and 0.4 for bandwidth, meaning that the SNR metric (which represents the signal strength of a user to a base station) is given more significance than the requested bandwidth metric when allocating a user to a base station. These weights were selected as practical design parameters for the evaluated scenarios, giving slightly higher priority to signal quality, since SNR directly affects link reliability and achievable throughput, while still considering the bandwidth requirements of each user. Other parameters, such as base-station capacity and bandwidth limits, follow the simulation setup described in Section IV and are used consistently across all examined algorithms to ensure a fair comparison. Next, each base station's user assignments have to be stored in three initialized data structures. An overflow matrix is built for users who cannot be accommodated due to bandwidth restrictions. According to each iteration of the changed matrix, downstream values are recalculated, and users are assigned to suitable base stations in accordance with their needs. The user's information is added to the appropriate structure if the base station has enough bandwidth; otherwise, the user is flagged for overflow.

The selected 0.6/0.4 weighting configuration is used consistently across the evaluated scenarios in order to provide a fixed and reproducible comparison among the examined algorithms. Increasing the SNR weight would be expected to prioritize link quality and achievable throughput, while increasing the bandwidth weight would give higher priority to user demand. A broader sensitivity analysis over multiple weight combinations and system parameters is not included in the present evaluation and is therefore identified as a limitation and future research direction.

The Hungarian assignment step is implemented through the MATLAB matchpairs function, which solves the linear assignment problem for the rows and columns of a cost matrix. Let U denote the number of users, B the number of base stations, and A the number of available antenna/resource positions per base station. The resulting assignment matrix has dimensions U x (B · A). According to the reported implementation characteristics of matchpairs, the worst-case complexity is $O(N^3 \log(N))$ for dense matrices and $O\big(nnz * Nlog(N)\big)$ for sparse matrices, where N denotes the assignment problem size and nnz is the number of non-zero entries in the cost matrix. In this manuscript, these expressions are used to indicate theoretical scalability trends of the implementation rather than hardware-dependent runtime guarantees.

Algorithm 3 – Hungarian Algorithm

```
    allocatedCapacity_bps = Array of size bs, initialized with zeros
    combinedCostMatrix = Array of size numUsers x (bs * bsAntennas), initialized with zeros
    antennaCounter = 1
    minSNR = Minimum SNR value from dynamic_pathloss_bs_ue
```

```
    maxSNR = Maximum SNR value from dynamic_pathloss_bs_ue

    for i = 1 to numUsers: // Normalize SNR & bandwidth, calculate combined score for each user & bs
      for currentBaseStation = 1 to bs:
        for antennaCounter = 1 to bsAntennas:
          userIndex = (currentBaseStation - 1) * numUsers + i
          normalizedSNR = (dynamic_pathloss_bs_ue(userIndex).SNR - minSNR) / (maxSNR - minSNR)
          normalizedBandwidth = (dynamic_pathloss_bs_ue(userIndex).Downstream - 1000000) / (25000000 - 1000000)
          combinedScore = weightSNR * normalizedSNR + weightBandwidth * normalizedBandwidth
          columnIdx = (currentBaseStation - 1) * bsAntennas + antennaCounter
          combinedCostMatrix[i][columnIdx] = combinedScore
M = HungarianAlgorithm(combinedCostMatrix, 1000) // Find optimal matching pairs using the Hungarian algorithm
```

Algorithm 3 – Matchpairs Allocation to data structs

```
    if basestationw == 1:
      // Allocate users to basestation 1
      bw1 = bw1 - downstream_valueHZ;
      if bw1 > 0:
        assignedtobs1.Add({'usernumber': usernumberw, 'Services': dynamic_pathloss_bs_ue(matching_index).Services})
      else:
        overflow.Add([usernumberw, basestationw])
    elseif basestationw == 2:
      // Allocate users to basestation 2
      bw2 = bw2 - downstream_valueHZ;
      if bw2 > 0:
        assignedtobs2.Add({'usernumber': usernumberw, 'Services': dynamic_pathloss_bs_ue(matching_index).Services})
      else:
        overflow.Add([usernumberw, basestationw])
    else:
      // Allocate users to basestation 3
      bw3 = bw3 - downstream_valueHZ;
      if bw3 > 0:
        assignedtobs3.Add({'usernumber': usernumberw, 'Services': dynamic_pathloss_bs_ue(matching_index).Services})
      else:
        overflow.Add([usernumberw, basestationw])

    second_best_snr_users = Empty Array of structs with fields 'usernumber', 'second_best_snr', 'service', 'bs', and 'reqinHZ'

    // Iterate through overflowed users and find second best SNR users
    for i = 1 to size(overflow, 1):
      usernumberov = overflow(i, 1);
      current_user_data = Filter dynamic_pathloss_bs_ue where 'usernumber' equals usernumberov
      sorted_data = Sort current_user_data by 'SNR' in descending order

      // Store information of second best SNR user and allocate if possible
      if size(sorted_data) >= 2:
        second_best_snr = sorted_data(2).SNR;
        reqinHZ = Calculate downstream value for second best SNR user

        if bw1 - reqinHZ >= 0 && sorted_data(2).basestation == 1:
          assignedtobs1.Add({'usernumber': usernumberov, 'Services': sorted_data(2).Services})
          Set matching row in overflow to zeros
        elseif bw2 - reqinHZ >= 0 && sorted_data(2).basestation == 2:
          assignedtobs2.Add({'usernumber': usernumberov, 'Services': sorted_data(2).Services})
          Set matching row in overflow to zeros
        elseif bw3 - reqinHZ >= 0 && sorted_data(2).basestation == 3:
          assignedtobs3.Add({'usernumber': usernumberov, 'Services': sorted_data(2).Services})
          Set matching row in overflow to zeros
      elseif size(sorted_data) >= 3:
        // Additional logic for third best SNR if required
end
```

3.4. Algorithm 4 - Minimum Cost Flow Algorithm

This study applies the Minimum Cost Flow algorithm to optimize resource allocation by modeling user-to-base station assignment as a flow problem with associated costs. The algorithm considers signal quality, user bandwidth demand, and network capacity constraints in order to allocate users to suitable base stations while minimizing the total allocation cost.

In the dynamic allocation case, the required downstream bandwidth and SNR of each user are used to guide the assignment process. The algorithm tracks the allocated bandwidth of each antenna/base station and ensures that capacity limits are not exceeded. If a feasible assignment cannot be made due to insufficient bandwidth, the user is marked as overflow and a second allocation attempt is performed by considering alternative base stations. The computational complexity of the Minimum Cost Flow algorithm depends on the network size and is typically $O(V^2E)$, where V is the number of nodes, representing users and base stations, and E is the number of edges, representing possible connections.

Algorithm 4 – Minimum Cost Flow

```
    # Function to find the next available task for a worker
    function find_next_available_task(worker_id, G, assigned_tasks):
      available_tasks = get_available_tasks(G, worker_id) - assigned_tasks
      if available_tasks is not empty:
```

```
        sorted_tasks = sort_tasks_by_weight(available_tasks, G, worker_id)
        return sorted_tasks[0]
    else:
        return None

# Main function to solve the assignment problem
function solve_assignment_problem(csv_file):
    initialize_bandwidth_variables()
    G = create_graph_from_csv(csv_file)
    max_worker_id = get_max_worker_id(G)
    assigned_tasks = empty_set()
    costs = empty_list()
    task_ids = empty_list()
    pathlosses = empty_list()
    flow_dict = find_min_cost_flow(G)
    overflow, overflow_matrix = process_workers(G, max_worker_id, assigned_tasks, flow_dict, costs, task_ids, pathlosses)
    print_results(costs, pathlosses, assigned_tasks, overflow, overflow_matrix)

# Function to process workers and handle overflow cases
function process_workers(G, max_worker_id, assigned_tasks, flow_dict, costs, task_ids, pathlosses):
    overflow = 0
    overflow_matrix = empty_list()
    for worker_id in range(1, max_worker_id + 1):
        task_dict = get_task_dict(flow_dict, worker_id)
        if task_dict is not None:
            task_id = find_next_available_task(worker_id, G, assigned_tasks)
            if task_id is not None:
                assign_task_to_worker(worker_id, task_id, G, assigned_tasks, costs, pathlosses, task_ids)
            else:
                print(f"No feasible assignment found for Worker {worker_id}")
        else:
            print(f"No feasible assignment found for Worker {worker_id}")
    handle_overflow_cases(G, assigned_tasks, overflow, overflow_matrix)
    return overflow, overflow_matrix

# Function to assign a task to a worker
function assign_task_to_worker(worker_id, task_id, G, assigned_tasks, costs, pathlosses, task_ids):
    edge_data = G[worker_id][task_id]
    task_downstream = edge_data['downstream']
    task_snr = calculate_snr(edge_data['pathloss'])
    downstream_value_HZ = task_downstream / log(1 + pow(10, task_snr / 10), 2)
    if is_bandwidth_available(worker_id, task_id, downstream_value_HZ):
        update_bandwidth(worker_id, task_id, downstream_value_HZ)
        store_assignment_info(worker_id, task_id, G, assigned_tasks, costs, pathlosses, task_ids)
        remove_assigned_task_edges(worker_id, task_id, G)
    else:
        handle_overflow_case(worker_id, task_id, G, assigned_tasks)

# Function to handle overflow cases
function handle_overflow_case(worker_id, task_id, G, assigned_tasks):
    overflow += 1
    print(f"No available bandwidth for Worker {worker_id} and Task {task_id}")
    overflow_matrix.append({'worker_id': worker_id, 'downstream_valueHZ': downstream_value_HZ})
    remove_assigned_task_edges(worker_id, task_id, G)

# Function to print the final results
function print_results(costs, pathlosses, assigned_tasks, overflow, overflow_matrix):
    num_users = len(costs)
    modified_pathlosses = [your_formula(pathloss, num_users) for pathloss in pathlosses]
    total_pathloss = sum(modified_pathlosses)
    user_data = total_pathloss / num_users
```

To complement the asymptotic complexity discussion, Table 2 summarizes the computational complexity and practical trade-offs of the examined allocation algorithms. The table is intended to clarify the relationship between algorithmic cost, allocation behavior, and practical suitability under the evaluated DeepMIMO-based scenarios. Actual execution-time and memory profiling are implementation- and hardware-dependent and are therefore considered as future work.

Table 2 Computational Complexity and Practical Trade-Offs of the Examined Allocation Algorithms

Algorithm	***Complexity / implementation aspect***	***Performance behavior***	***Practical interpretation***
Simple allocation	Lowest computational burden; mainly based on SNR sorting and capacity checking	Simple and fast, but may produce unbalanced base-station loading and overflow users	Suitable as a baseline and for low-complexity allocation
Hungarian algorithm	Polynomial-time assignment method; higher computational cost due to construction and solution of the assignment matrix	Improves user-to-base station assignment quality and can provide higher per-user throughput in some scenarios	Suitable when improved assignment quality is preferred over minimum computational burden
Minimum Cost Flow	Graph-based optimization with computational cost depending on the number of nodes, edges, and assigned flow	Provides balanced resource allocation and effective bandwidth distribution in the evaluated scenarios	Suitable when capacity-aware allocation and balanced resource distribution are prioritized

This comparison shows that the examined algorithms involve different trade-offs. The simple allocation method has the lowest computational burden but provides limited control over load balancing. The Hungarian algorithm and Minimum Cost Flow require higher computational effort, but they can improve allocation quality under the evaluated DeepMIMO-based scenarios. Therefore, the choice of algorithm depends on the desired balance between computational simplicity, assignment quality, bandwidth distribution, and scalability. A complete runtime and memory profiling study over different hardware platforms, user densities, base-station deployments, and antenna configurations is outside the scope of the present work and is considered as future work.

The examined algorithms should be interpreted as assignment-level user-to-base station allocation methods rather than as per-slot MAC-layer schedulers. In dense deployments with thousands of users and large antenna configurations, the Hungarian and Minimum Cost Flow approaches may require centralized processing, and their runtime depends on the assignment-matrix size, graph size, solver implementation, and available computational resources. Therefore, the present framework is more suitable for comparative evaluation, network planning, or periodic resource reconfiguration than for direct ultra-low-latency scheduling without further implementation optimization. A complete evaluation of real-time scheduling feasibility, runtime latency, memory usage, distributed implementation, and hardware-specific acceleration is outside the scope of the present work and is considered as future work.

4. DeepMIMO Environment

4.1. DeepMIMO Features

In our simulations, the DeepMIMO framework was used for generating large-scale MIMO datasets. DeepMIMO is a powerful simulation tool that provides channel models based on accurate 3D ray-tracing data generated by Remcom Wireless InSite [15,24,25]. One of the key strengths of DeepMIMO is its ability to simulate real-world-like environments by capturing the effects of environment geometry and material properties on signal propagation. The framework's ability to produce datasets that depend on actual transmitter and receiver locations makes it ideal for simulating 5G MM networks. DeepMIMO's primary advantage is that it captures realistic propagation effects in highly complex environments, such as urban areas, where buildings, obstacles, and material types influences signal transmission. This feature makes it particularly suitable for evaluating algorithms in mmWave and MM scenarios, where precision in modeling channel characteristics is critical. Furthermore, DeepMIMO is designed to integrate seamlessly with machine learning applications, making it an ideal choice for developing and testing resource allocation algorithms that rely on data-driven models.

The DeepMIMO framework offers several advantages for simulating large-scale MIMO systems. One key benefit is its accurate ray-tracing capability, as it is built on the Remcom Wireless InSite ray-tracing engine. This allows for realistic modeling of signal paths, which is crucial for understanding multipath effects in dense urban environments. Additionally, flexible dataset generation is possible with DeepMIMO, enabling users to tailor datasets for specific scenarios, such as urban or rural environments, with various parameters like frequency bands, antenna configurations, and base station placements. The tool's scalability makes it suitable for simulating scenarios with massive numbers of users and antennas, essential for modeling the large-scale deployments typical in 5G networks. Furthermore, DeepMIMO is well-suited for integration with machine learning applications, as it generates extensive datasets that can be used for training and testing ML models. However, DeepMIMO has certain limitations. Its datasets are based on predefined environments, which limits their adaptability to real-time changes in network conditions, such as moving obstacles or changing weather. The computational complexity of running detailed ray-tracing simulations for large-scale environments can be high, especially when simulating dense urban areas with many users. Additionally, DeepMIMO's reliance on predefined scenarios, such as fixed urban street grids, may not fully capture the diversity of real-world environments, like indoor or suburban areas. In this study, we used the O1 outdoor scenario from the DeepMIMO dataset, which corresponds to a 3D urban environment. This scenario was chosen due to its relevance to high-density user environments typical in 5G deployments. The simulations were conducted in MATLAB, where we modeled channel behavior, including path loss, for each user, and developed a traffic model to simulate varying user demands.

4.2. Simulations Scenarios

To effectively test our algorithms and simulate real-world conditions, several parameters must be considered. These include the base stations and their capacities, transmission power, noise power, the number of transmission (TX) antennas on the base station, user and antenna gain, and the available bandwidth, which is distributed among users to ensure equal resource allocation. Since all users in our scenario require the same resources, this uniform distribution is crucial. The DeepMIMO dataset can be completely defined by the chosen ray-tracing scenario and a set of parameters as outlined in Table 3, which enable the accurate definition and reproduction of the dataset. The site map in DeepMIMO is divided into rows that activate users in each specific row.

Table 3 Simulation Parameters

Simulation Parameters	***Value***
Base Station Bandwidth	400MHz
Subcarrier Spacing	120 KHz
NRB	264
Active Subcarriers	64
User Gain	0
User Antenna	1
Antenna Gain	21 dBi

Simulation Parameters	*Value*
Tx Antennas	1000
Transmission Power	45 dBm

To improve reproducibility, the same DeepMIMO scenario settings, simulation parameters, and extracted channel-related quantities were used across all examined algorithms. The simulations were implemented in MATLAB, using the DeepMIMO O1 and O2 scenarios as input datasets. The extracted parameters used by the allocation algorithms include pathloss, SNR values, user demand, base-station bandwidth capacity, transmission power, and antenna configuration. No additional randomization was introduced during the algorithmic comparison; therefore, each allocation method was evaluated using the same user configurations, base-station settings, and resource constraints. The Hungarian assignment step was implemented using MATLAB matchpairs, while the Minimum Cost Flow formulation was implemented as a graph-based user-to-base station allocation problem. These settings were kept consistent across the examined algorithms in order to ensure a reproducible and fair comparison.

The adopted simulation model focuses on assignment-based user-to-base station allocation using channel-quality and bandwidth-demand information. More specifically, the examined algorithms use SNR-related values, bandwidth requirements, and base-station capacity constraints in order to compare their behavior under common DeepMIMO-based assumptions. Therefore, the model should be interpreted as an assignment-level abstraction for comparative algorithm evaluation, rather than as a complete operational 5G Massive MIMO system model. Practical deployment aspects such as scheduling constraints, beamforming optimization, inter-cell interference coordination, user mobility, channel-estimation inaccuracies, traffic variability, QoS differentiation, and latency requirements are not explicitly modeled in the present evaluation. These aspects would require a broader system-level simulation framework and are considered important directions for future work.

The simulation parameters, including the SNR weight, bandwidth weight, base-station bandwidth, antenna configuration, and transmission power, are kept fixed across the evaluated scenarios in order to ensure a consistent comparison among the examined allocation algorithms. This fixed-parameter setup allows the relative behavior of the simple allocation, Hungarian, and Minimum Cost Flow methods to be evaluated under common assumptions. However, the results should be interpreted within the selected parameter configuration, since different values of these parameters may affect the relative performance of the algorithms. A broader sensitivity analysis over multiple weight combinations, bandwidth capacities, antenna configurations, and transmission power levels is not included in the present evaluation and is considered as future work.

The scenarios used in this paper are the O1 (Outdoor 1) and O2 Dynamic (Outdoor 2) Scenario from DeepMIMO with the operating frequency to be 60 GHz and 3.5 GHz accordingly. This 1st scenario consists of 18 base stations and 3 user grids with an overall of 1,184,923 users that are scattered through the site map (181 users per row) to provide an accurate representation of a real scenario. Figure 1 depicts the site map of the O1 scenario which was used to create one of the datasets needed for our algorithms testing. The 2nd scenario consists of 2 base stations and 3 user grids with 116,303 users scattered across the site map. In the 2nd and 3rd user grid of O2 each row consists of 191 users, but the 1st user grid includes 1891 users per row. Figure 2 illustrates the O2 scenario site map.

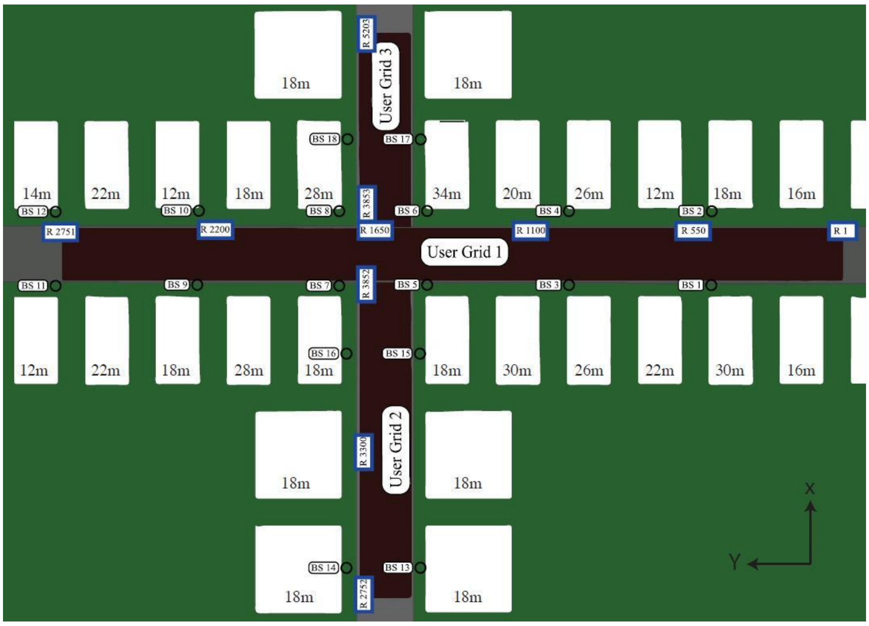

Fig. 1 Top view of the O1 scenario [26].

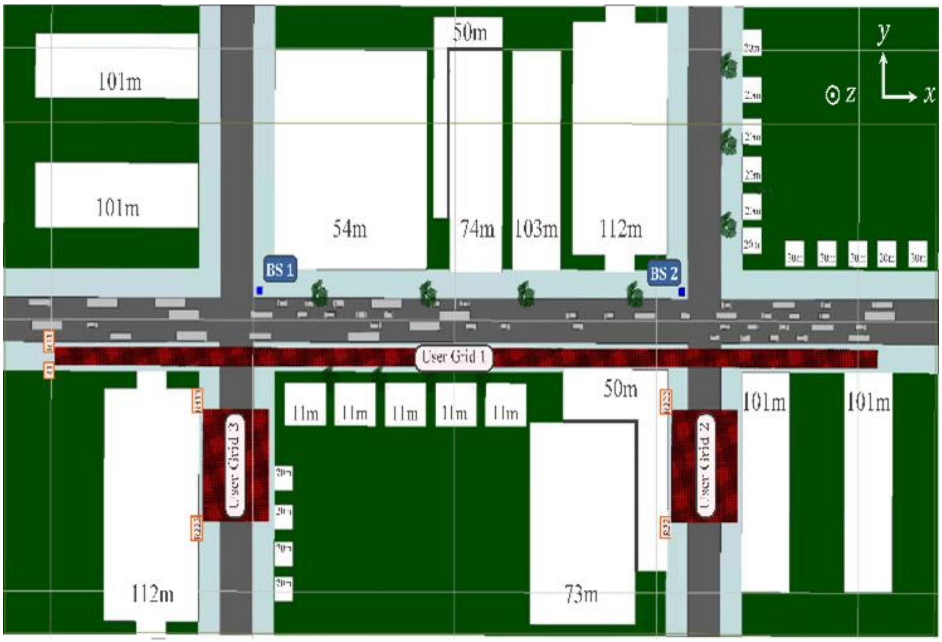

Fig. 2 Top view of the O2 scenario. [27]

5. Resource Allocation Analysis

In this section, the performance of the resource allocation algorithms in MM 5G networks presented in Section III are evaluated. First, Algorithm 1 is evaluated, which focuses on user assignment to a single base station. Simulation results are presented showing the impact of increasing user connections on throughput, SNR and overall network performance. In addition, the potential benefits of enabling multiple base stations to improve overall network throughput are validated. To improve resource allocation efficiency, the Hungarian and Minimum Cost Flow algorithm is utilized and compared with previous approaches. By evaluating these algorithms, we aim to identify the most optimized approach for resource allocation in MM 5G networks.

5.1. Algorithm 1 - User assignment to One Base Station

Simulations were run for 181, 362, 905, 1267, 2353, and 2534 users by selecting rows nearest to the active base station and adjusting the user number each time. Specifically, combinations of Rows 1090 to 1103 were used to activate the users. Figure 3 shows that as more users connect to the base station, their data rates decrease. The x-axis represents the number of users simultaneously connected to the base station, while the y-axis shows the average throughput in Mbps.

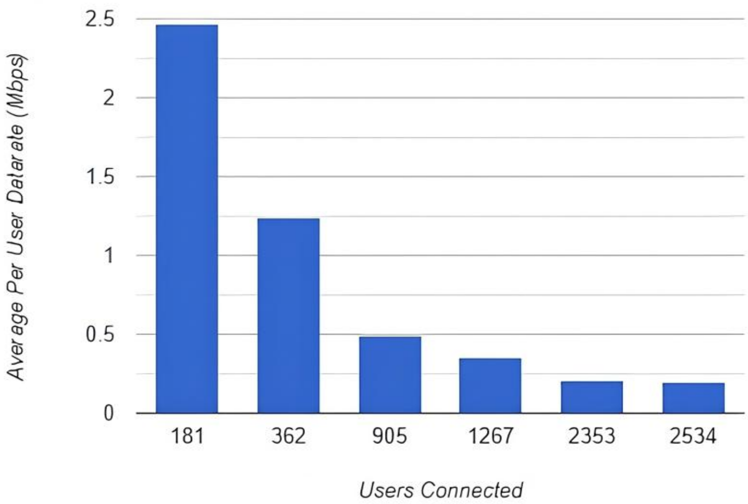

Fig. 3 Average per user throughput (1 active BS) – O1 scenario

In general, in a 5G MM scenario with one active base station, the use of multiple antennas allows for simultaneous transmission and reception of data, resulting in a higher per-user throughput compared to traditional single-antenna systems. However, as the number of users connected to the base station increases, the per-user throughput is likely to decrease. This decrease is due to a combination of factors: the finite resources of the base station need to be shared among more users, and increased interference from overlapping radio waves transmitted by multiple users. The addition of more users leads to a higher demand on the available bandwidth, and the overlapping transmissions create interference, which further impacts the data rate for each individual user. Also, the SNR is a very important aspect in determining the strength and quality of the received signal, so we also must examine how it changes with more users connecting. Since the signal strength is weakened, it lowers the SNR as well. Also, something that can also lead to an increase in the total throughput of the network is to activate more base stations since more coverage is provided and the load on existing base stations is reduced.

5.2. *Algorithm 2 - With Many Base Stations Active*

To investigate the impact of increasing the number of users on per-user throughput and user allocation in a 5G MM network, simulations were conducted with three active base stations and user numbers ranging from 181 to 2534. Base stations BS3, BS4 and BS5 (Figure 1) were used from the O1 scenario and combinations from lines 1090 to 1103 for the activation of users. Figure 4 shows that as the number of users increases, the average throughput per user decreases. The x-axis refers to the number of users that were active and, on the y-axis, the average throughput in Mbps is shown. Additionally, Figure 5 presents how the users were allocated for each scenario to each one of the 3 base stations used for this simulation. It is shown that BS3 and BS4 occupied most of the users and the rest were allocated to BS5 but not in an efficient and balanced way. When the scale of the problem got bigger, base stations 3 and 4 were filled fully and only a few users were left for the 3rd base station (BS5).

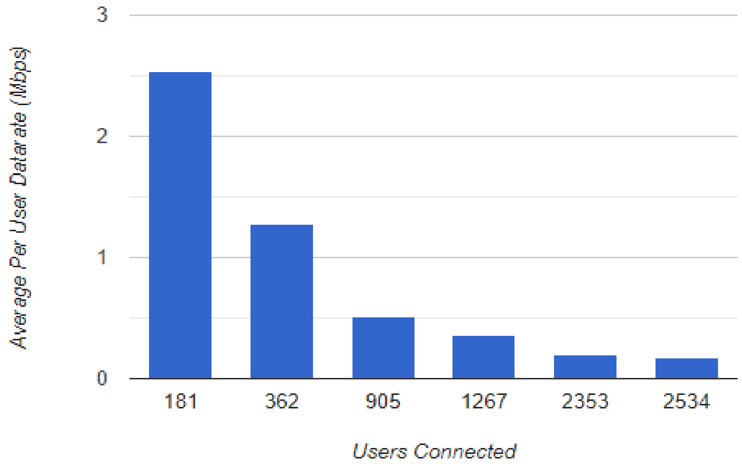

Fig. 4 Average per user throughput to User count Graph (3 active BS)

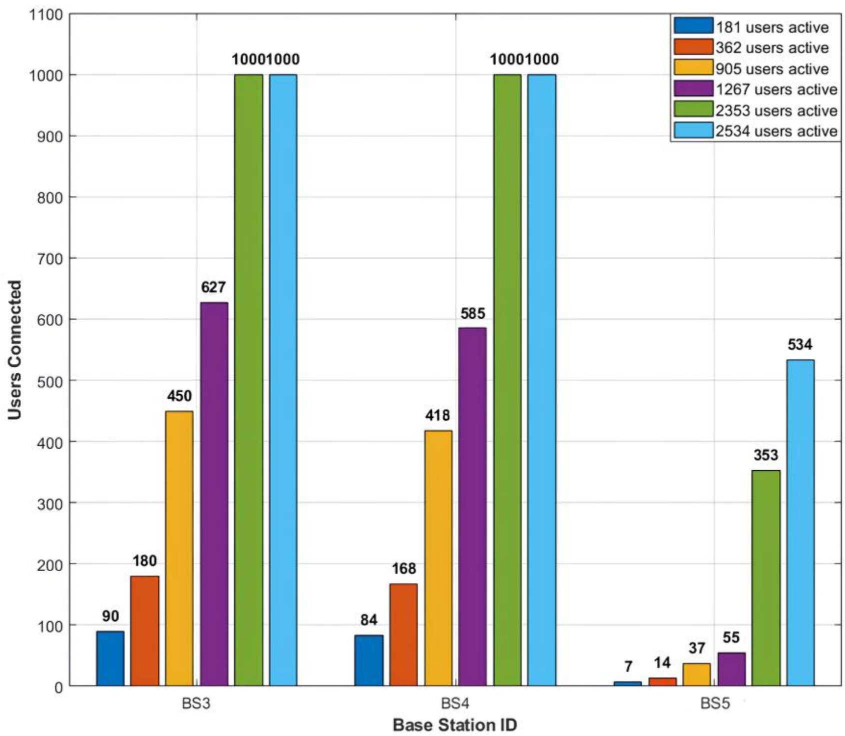

Fig. 5 User allocation for each base station (3 active BS) – O1 scenario

When three base stations are active, it is observed that the average per user data rate increases. Moreover, running the same simulation with four active base stations it was observed that there was a very small (0.1%) increase in the average data rate compared to having three base stations active. In Figure 6 the reduction in the average per-user throughput is almost linearly proportional to the increase in the number of users. Also, in Figure 7 we see how the users were allocated for 4 active base stations. BS3 and BS4 also show a steep increase in user connections as the total number of active users rises. For instance, with 2353 and 2534 active users, both BS3 and BS4 reach their maximum capacity, connecting 1000 users each. This capacity is due to the Tx Antennas of each base station which were defined as 1000 in Section IV. This uneven distribution highlights a critical issue in the allocation strategy of Algorithm 2. Similarly with 3 active base stations, BS3 and BS4 occupied most of the users and left the rest for BS5 and BS6. We can therefore assume that Algorithm 2 does not distribute the users efficiently.

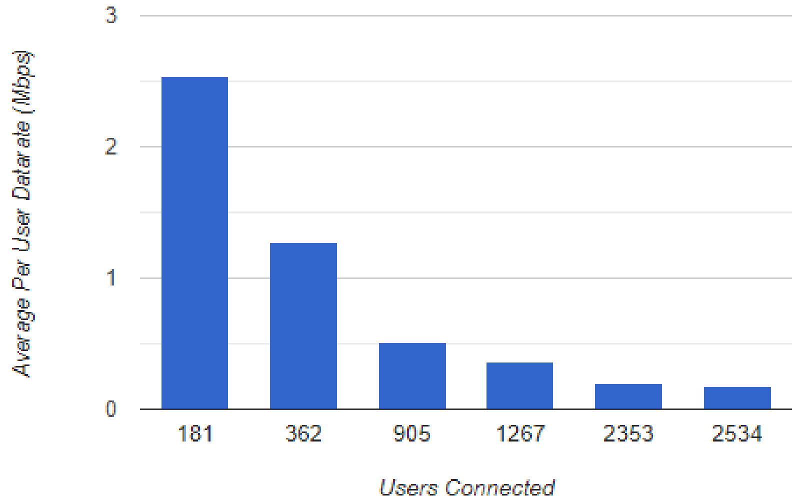

Fig. 6 Average per user throughput to User (4 active BS) – O1 scenario

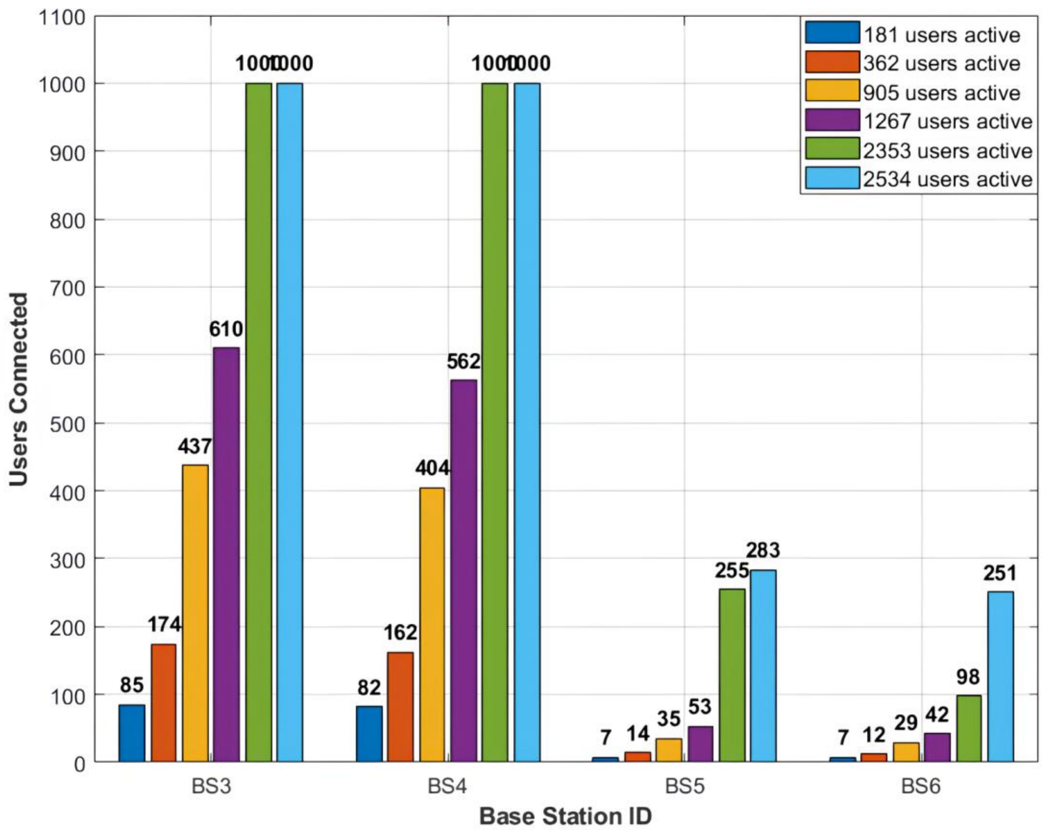

Fig. 7 User allocation for each base station (4 active BS) – O1 scenario

We also run simulations for the dynamic part of this algorithm presented in Section III.B. Figures 8 and 9 illustrate the number of users assigned to each base station and the bandwidth distribution for each simulation of active users. Table 4 summarizes the results. The 5th row (overflow) represents the users that did not connect to a base station because they were all full. It is important to mention that the use of this algorithm has as result some of the users not being able to connect and receive service due to the inefficient management of network resources by the proposed algorithm. As Table 4 shows, for user counts up to 1267, the bandwidth utilization for BS1 and BS2 show a steady increase with the rising number of users, meaning that the initial allocation is efficient. The average bandwidth utilization across both base stations remains balanced, highlighting the algorithm's capacity to manage resources effectively. So, in cases with no overflow, ensuring that all users are served within the capacity limits of the base stations. In the last two experiments with 2353 and 2534 users, a total of 216.52MHz and 254.97MHz in user needs, could not be served by the network. This overflow suggests that the algorithm's distribution strategy fails to scale on par with the increasing demand. The root cause of this inefficiency lies in the non-optimal distribution of users across the base stations, leading to some base stations being overloaded while others have underutilized capacity.

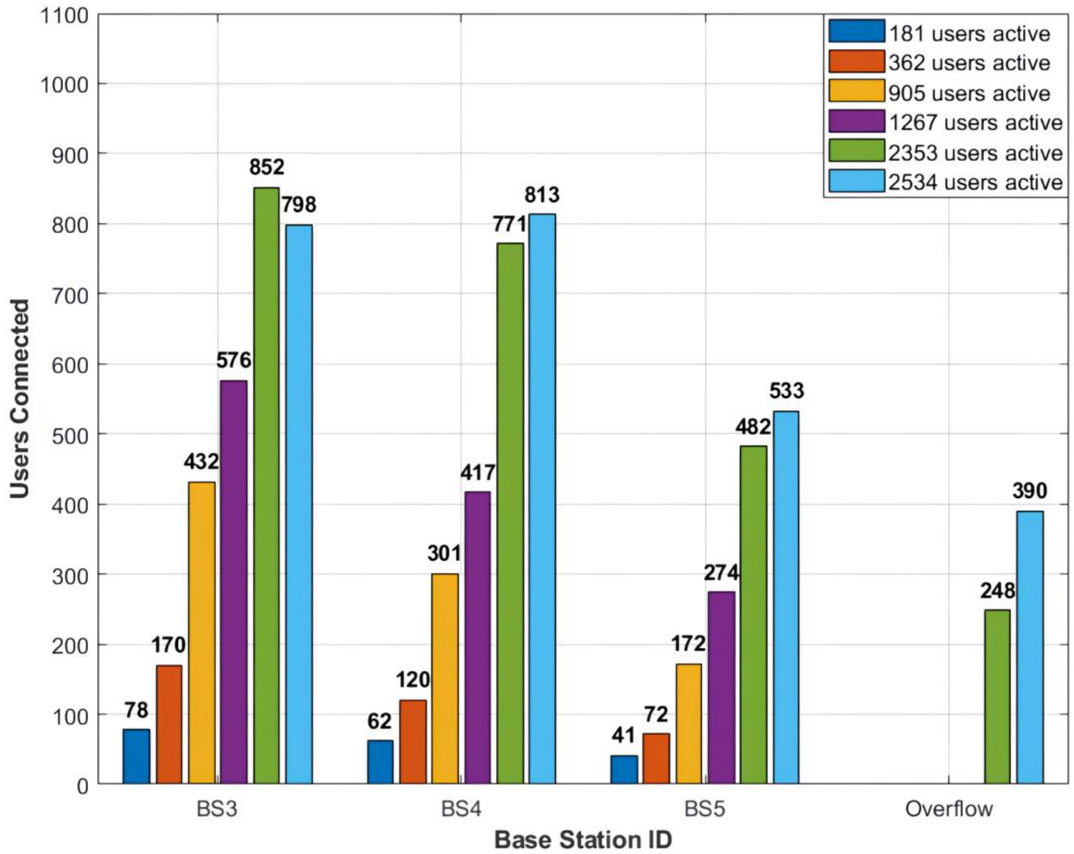

Fig. 8 User allocation for Algorithm 2 – O1 scenario

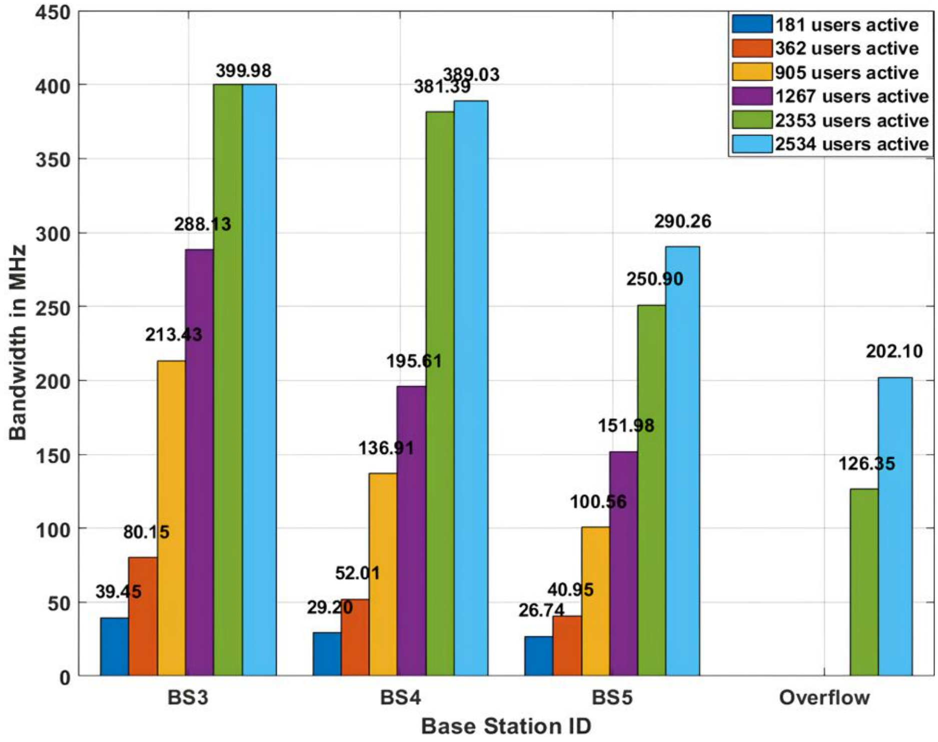

Fig. 9 Bandwidth Distribution for Algorithm 2 – O1 scenario

Table 4 Algorithm 2 Bandwidth – O1 Scenario

# Users	181	362	905	1267	2353	2534
BW of BS3	39.45 MHz	80.15 MHz	213.43 MHz	288.13 MHz	399.98 MHz	399.98 MHz
BW of BS4	29.2 MHz	52.01 MHz	136.91 MHz	195.61 MHz	381.39 MHz	398.03 MHz
BW of BS5	26.74 MHz	40.95 MHz	100.56 MHz	151.98 MHz	250.9 MHz	290.26 MHz
Average BS BW	31.78 MHz	57.7 MHz	150.3 MHz	211.9 MHz	344.09 MHz	362.76 MHz
Overflow BW	0 MHz	0 MHz	0 MHz	0 MHz	126.35 MHz	202.1 MHz

Similar simulations were run for the O2 scenario including the dynamic requirements for each user. Scene 1 and Rows 19 to 20 were used to activate 3782 users. A reprocessing was made to use the same number of users as the O1 scenario. Figure 10 illustrates the number of users assigned to each base station (BS1, BS2) and the overflow for varying numbers of active users. Here, BS1 consistently accommodates the highest number of users, reaching up to 1000 users in the scenario with 2353 and 2534 active users. This is followed by BS2, which handles 795 and 955 users for these same scenarios. As user density increases, a significant number of users are left unallocated, resulting in substantial overflow. In the scenarios with 2353 and 2534 users, the overflow reaches 558 and 579 users, respectively. This indicates that the algorithm fails to evenly distribute the load across available base stations, leading to inefficiencies and service degradation. Figure 11 and Table 5 detail the bandwidth distribution for each base station and the overflow for different user densities. Because of the users allocated, BS1 has the highest bandwidth usage followed by BS2. There is also a high overflow bandwidth which highlights the algorithm's inability to efficiently allocate resources to meet user demands. Algorithm 2 has left many users in the overflow state instead of re-distributing them.

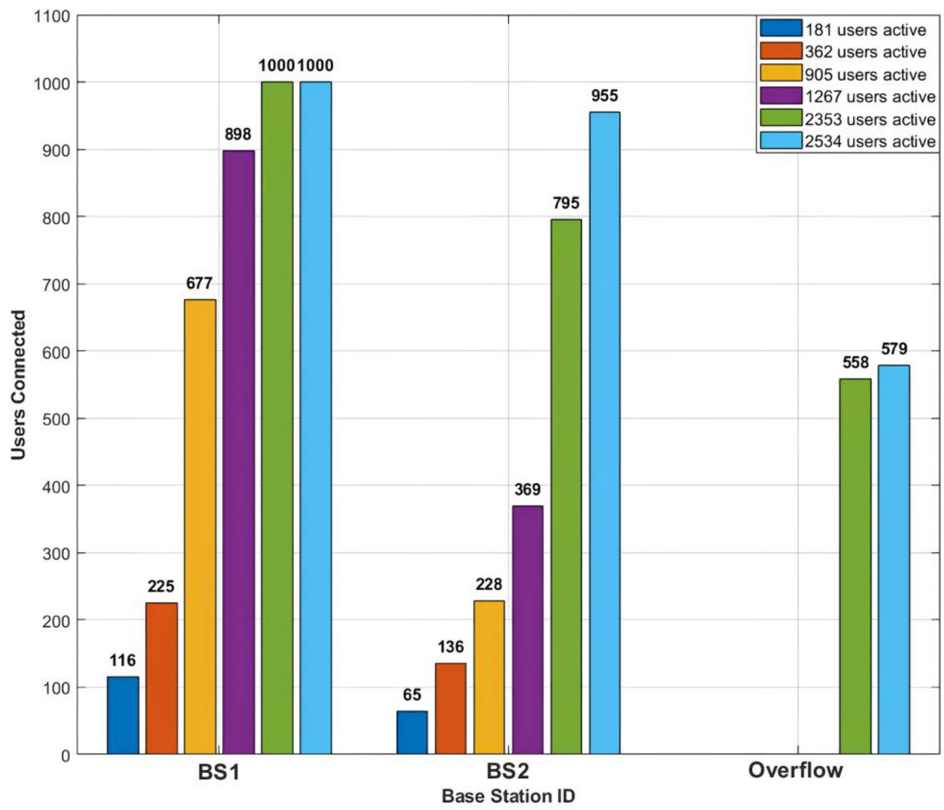

Fig. 10 User allocation for Algorithm 2 – O2 scenario

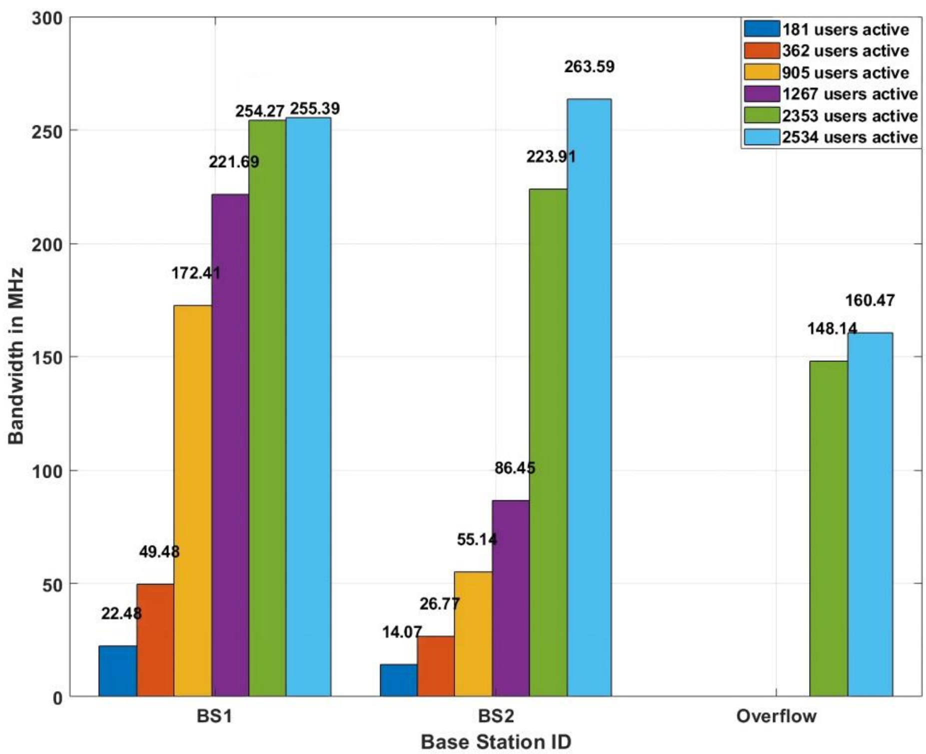

Fig. 11 Bandwidth Distribution for Algorithm 2 – O2 scenario

Table 5 Algorithm 2 Bandwidth – O2 Scenario

# Users	181	362	905	1267	2353	2534
BW of BS1	22.28 MHz	49.48 MHz	172.41 MHz	221.69 MHz	254.27 MHz	255.39 MHz
BW of BS2	14.07 MHz	26.77 MHz	55.14 MHz	86.45 MHz	223.91 MHz	263.59 MHz
Average BS BW	18.18 MHz	38.13 MHz	113.78 MHz	154.08 MHz	239.09 MHz	259.49 MHz
Overflow BW	0 MHz	0 MHz	0 MHz	0 MHz	148.14 MHz	160.47 MHz

Additionally, the difficulties in sustaining high performance in larger wireless networks can be seen by the nearly linear relationship between the number of users and the decrease in throughput. These discoveries may have significant effects on wireless network design and optimization, especially in densely populated areas where numerous users must be occupied simultaneously. However, managing and connecting users to the most efficient base station to achieve the best throughput per user can get easier by implementing more efficient algorithms. So, we will test the same scenario with the Minimum Cost Flow and Hungarian algorithms and compare all the results to determine which one produces the most optimized ones.

5.3. Algorithm 3 - Hungarian Algorithm

In our previous simulations for algorithms 1 and 2, we tested the simulation with 181, 362, 905, 1267, and 2353 users. To conduct these simulations, a cost matrix was created by extracting data from the DeepMIMO scenario, using parts of algorithms 1 and 2 to gather the necessary information for the linear assignment problem. The simulation results mirrored those of algorithm 2. Compared with the simple allocation procedure, the Hungarian algorithm provides a structured assignment-based formulation. Its computational cost is polynomial and depends on the assignment-matrix size, as discussed in Section 3.3. Therefore, the comparison in this section focuses on allocation behavior and performance trends rather than hardware-dependent runtime superiority.

Through careful optimization, this method achieved significant improvements in average throughput per user and bandwidth allocation as well as a better distribution of users through the base stations. The results are particularly noticeable in Figure 12 and 13, showing a more balanced and fair distribution of users and bandwidth across base stations. Table 6 presents the results. Using this algorithm, we have unassigned users only in the 2534 users scenario and the unassigned users are only 94 compared with the previous algorithm which has 390 unassigned users. Moreover, using this algorithm the bandwidth of the unassigned users is only 94MHz compared to 202.1MHz of the previous algorithm.

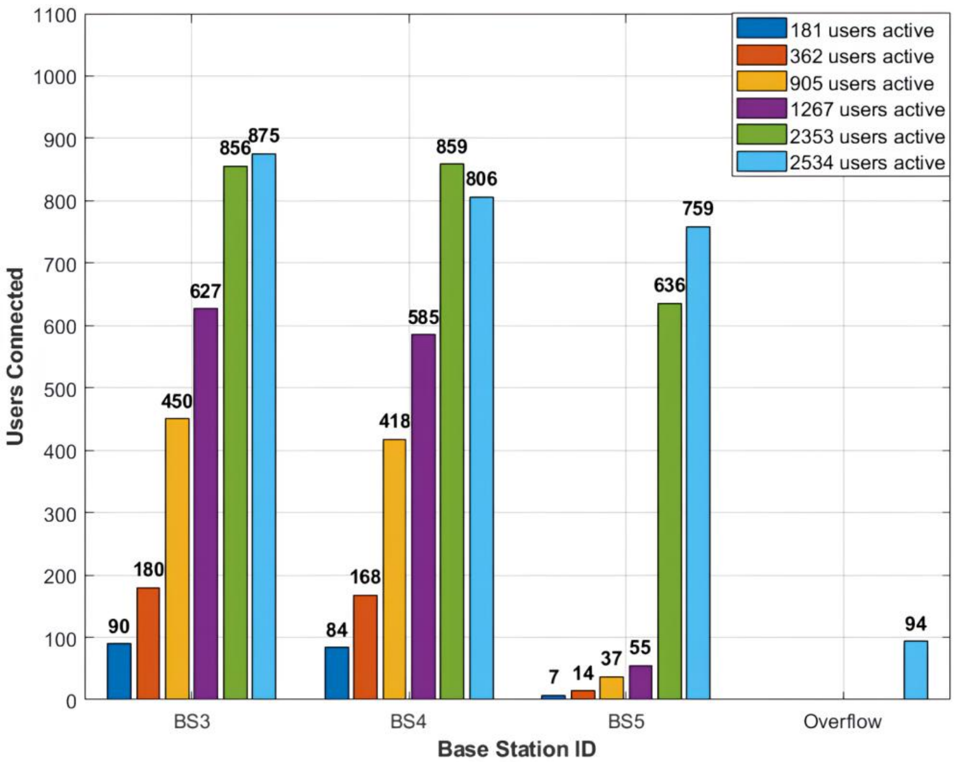

Fig. 12 User allocation for Hungarian Algorithm – O1 scenario

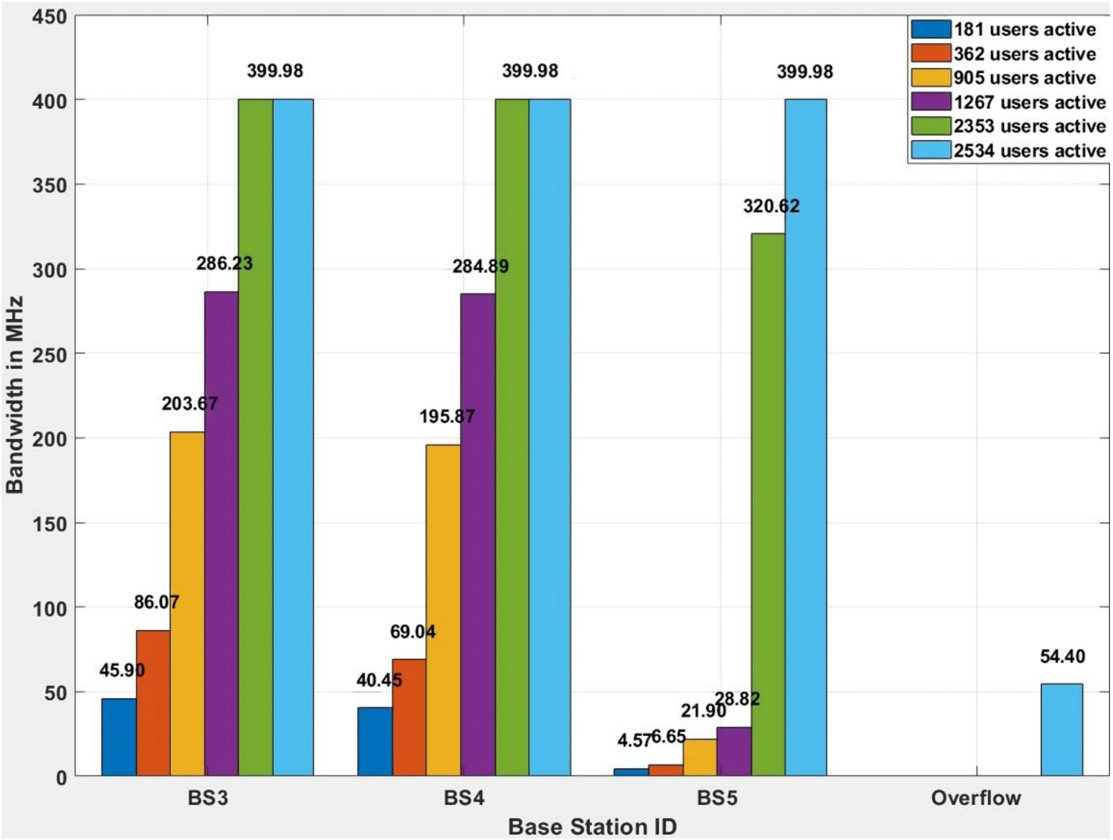

Fig. 13 Bandwidth Distribution for Hungarian Algorithm – O1 scenario

Table 6 Hungarian Algorithm Bandwidth – O1 Scenario

# Users	181	362	905	1267	2353	2534
BW of BS3	45.9 MHz	86.07 MHz	203.67 MHz	286.23 MHz	399.98 MHz	399.98 MHz
BW of BS4	40.45 MHz	69.04 MHz	195.87 MHz	284.89 MHz	399.98 MHz	399.98 MHz
BW of BS5	4.56 MHz	6.65 MHz	21.9 MHz	28.82 MHz	320.62 MHz	399.98 MHz
Average BS BW	30.30 MHz	53.92 MHz	140.48 MHz	199.98 MHz	373.53 MHz	399.98 MHz
Overflow BW	0 MHz	0 MHz	0 MHz	0 MHz	0 MHz	54.4 MHz

Simulations were run for the O2 scenario for 181, 362, 905, 1267 and 2353 using the Hungarian Algorithm. Figures 14 and 15 illustrate the results. Figure 14 displays the user distribution for the 2 base stations and Figure 15 the bandwidth distribution. They seem to follow a similar distribution for resource allocation and bandwidth as the experiments for the O1 scenario. The Hungarian Algorithm exhibited a higher bandwidth distribution, peaking at 399.99 MHz for both base stations, compared to Algorithm 2. However, it resulted in more overflowing users, indicating a trade-off between user allocation and bandwidth efficiency. Despite the increased number of overflowed users, the Hungarian Algorithm managed to allocate bandwidth more effectively across the network, ensuring that base stations operated closer to their optimal capacity and maintaining overall network performance for the O2 scenario. Table 7 displays the bandwidth distribution results of the Hungarian Algorithm. The Hungarian Algorithm strives to achieve a more equitable and effective distribution of bandwidth and a balanced allocation of users. It ensures that each base station can distribute a fair amount of bandwidth, connecting with as many users as possible while maximizing bandwidth utilization. Although it results in more overflowing users, this approach leaves some capacity at each base station, enabling the network to perform well and accommodate additional users as the problem scale increases.

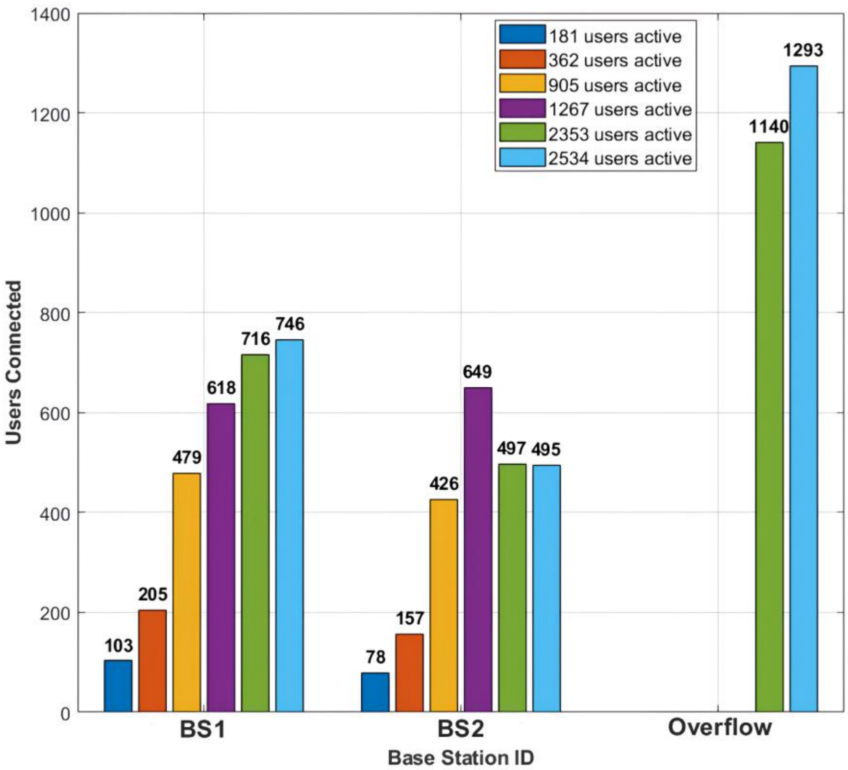

Fig. 14 User allocation for Hungarian Algorithm – O2 scenario

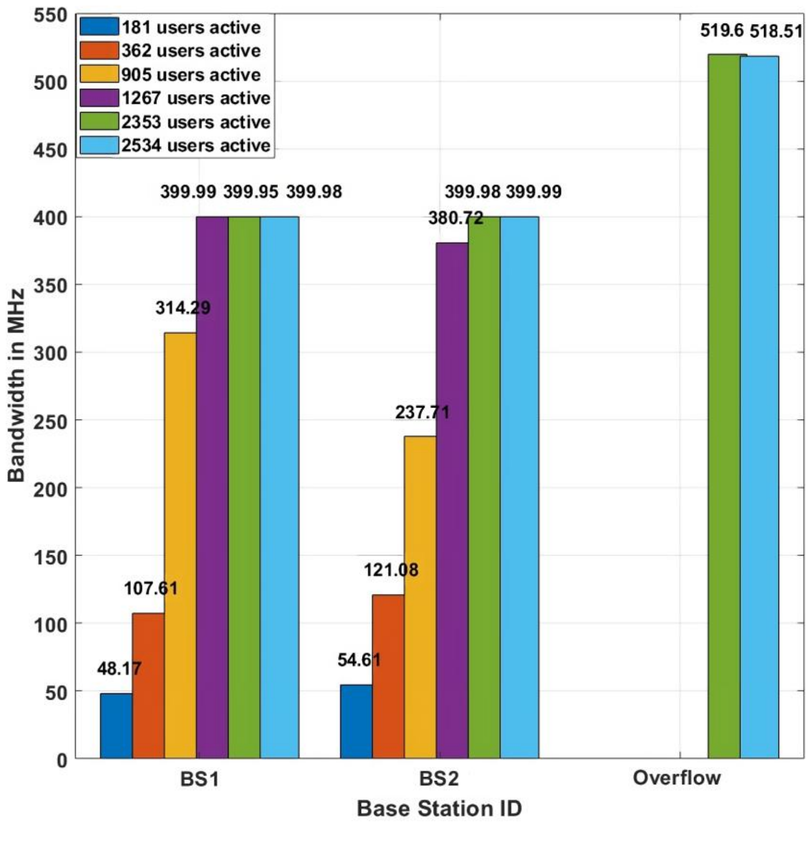

Fig. 15 Bandwidth Distribution for Hungarian Algorithm – O2 scenario

Table 7 Hungarian Algorithm Bandwidth – O2 Scenario

# Users	181	362	905	1267	2353	2534
BW of BS1	48.17 MHz	107.61 MHz	314.29 MHz	399.99 MHz	399.95 MHz	399.98 MHz
BW of BS2	54.61 MHz	121.08 MHz	237.71 MHz	380.72 MHz	399.98 MHz	399.99 MHz
Average BS BW	51.39 MHz	114.345 MHz	276 MHz	390.355 MHz	399.97 MHz	399.99 MHz
Overflow BW	0 MHz	0 MHz	0 MHz	0 MHz	519.6 MHz	518.51 MHz

5.4. Algorithm 4 - Minimum Cost Flow Algorithm

In our simulations with user counts of 181, 362, 905, 1267, 2353, and 2534, the Minimum Cost Flow algorithm demonstrated remarkable efficiency in maximizing per-user throughput and achieving a distribution similar to that of the Hungarian algorithm. Figures 16 and 17 present the resource allocation results and bandwidth distribution for Minimum Cost Flow, while Table 8 summarizes the bandwidth outcomes. Unassigned users appeared only in the 2534 users scenario, with 142 unassigned users, who were allocated a bandwidth of just 75.97 MHz.

For the O2 scenario, Figures 18 and 19 illustrate the user and bandwidth distribution, with Table 9 summarizing the bandwidth results. The Minimum Cost Flow algorithm surpassed the Hungarian algorithm in execution time and computational efficiency. As user numbers increase, the Hungarian algorithm's time complexity grows cubically, leading to longer execution times. In contrast, the Minimum Cost Flow algorithm scales more efficiently, reducing computational load and expediting resource allocation.

For the Minimum Cost Flow formulation, the user-to-base station allocation problem is modeled as a graph, where users and base stations are represented as nodes and feasible user-to-base station associations are represented as edges. Let U denote the number of users, B the number of base stations, V the number of graph nodes, and E the number of feasible edges. In this formulation, V depends on U and B, while E depends on the possible user-to-base station associations. The computational complexity depends on the selected Minimum Cost Flow solver and graph structure. For a successive-shortest-path implementation, the complexity can be expressed as $O(E + F * E \log V)$, where F denotes the amount of assigned flow. Therefore, the complexity discussion is expressed in terms of V, E, and F, and is used to support the qualitative scalability comparison of the examined assignment-based methods.

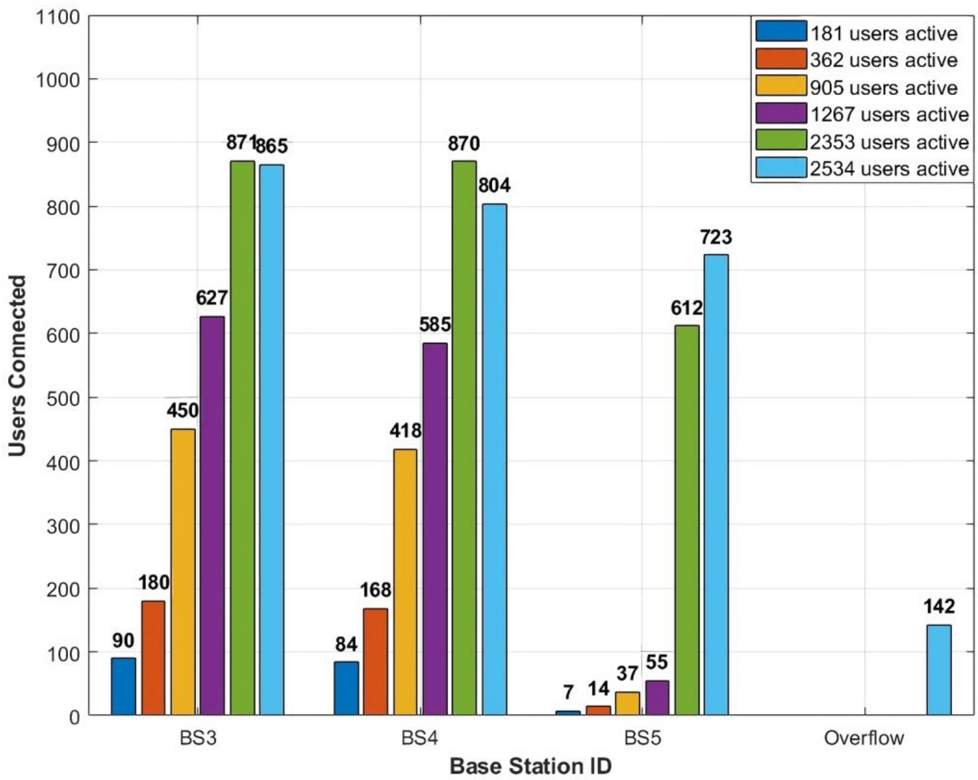

Fig. 16 User Distribution for Minimum Cost Flow Algorithm – O1 scenario

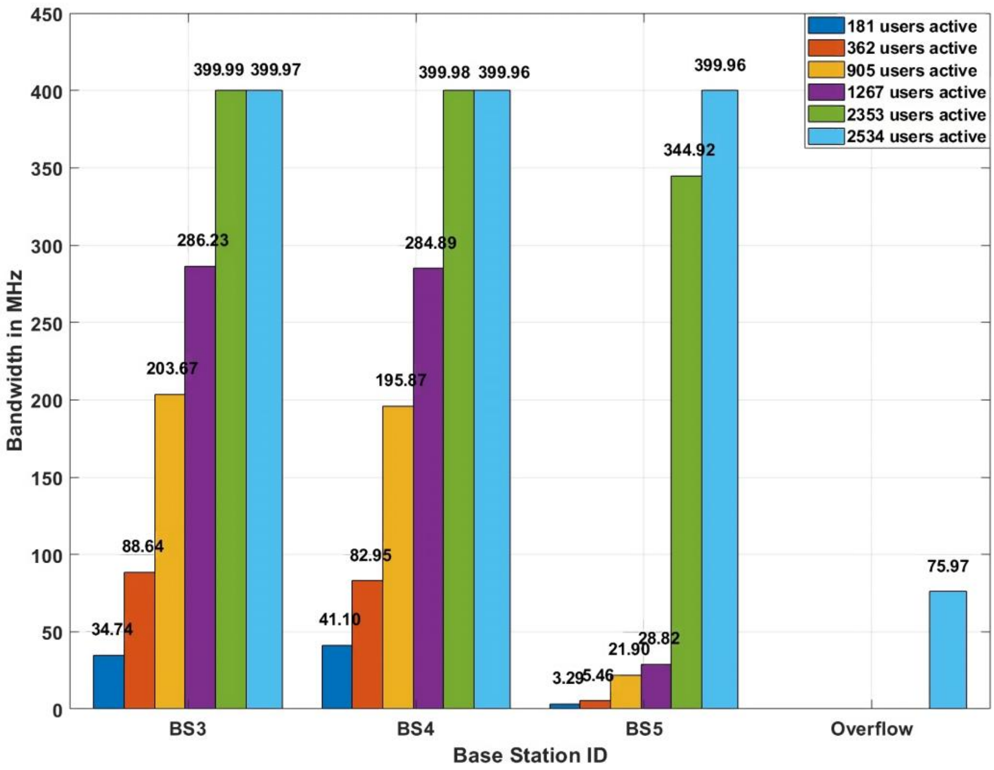

Fig. 17 Bandwidth Distribution for Minimum Cost Flow Algorithm – O1 scenario

Table 8 Minimum-Cost Flow Bandwidth – O1 scenario

# Users	181	362	905	1267	2353	2534
BW of BS3	34.74 MHz	88.64 MHz	203.67 MHz	286.23 MHz	399.99 MHz	399.97 MHz
BW of BS4	41.1 MHz	82.95 MHz	195.87 MHz	284.89 MHz	399.98 MHz	399.96 MHz
BW of BS5	3.29 MHz	5.46 MHz	21.9 MHz	28.82 MHz	334.92 MHz	399.96 MHz
Average BS BW	26.38 MHz	59.02 MHz	140.48 MHz	199.98 MHz	378.3 MHz	399.96 MHz
Overflow BW	0 MHz	0 MHz	0 MHz	0 MHz	0 MHz	75.97 MHz

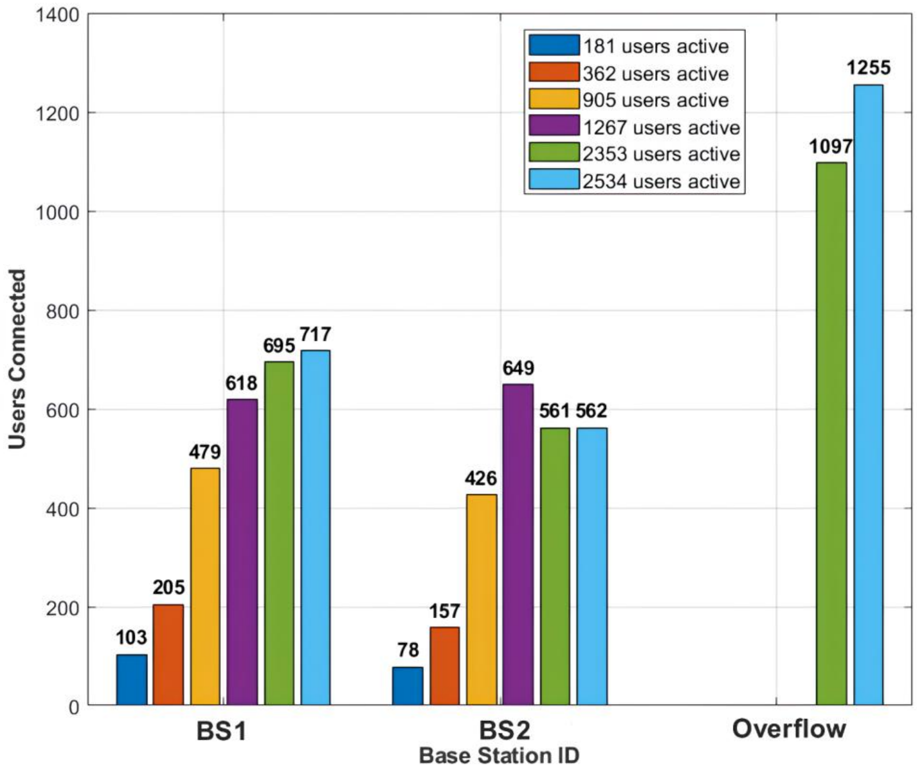

Fig. 18 User Allocation for Minimum Cost Flow Algorithm – O2 scenario

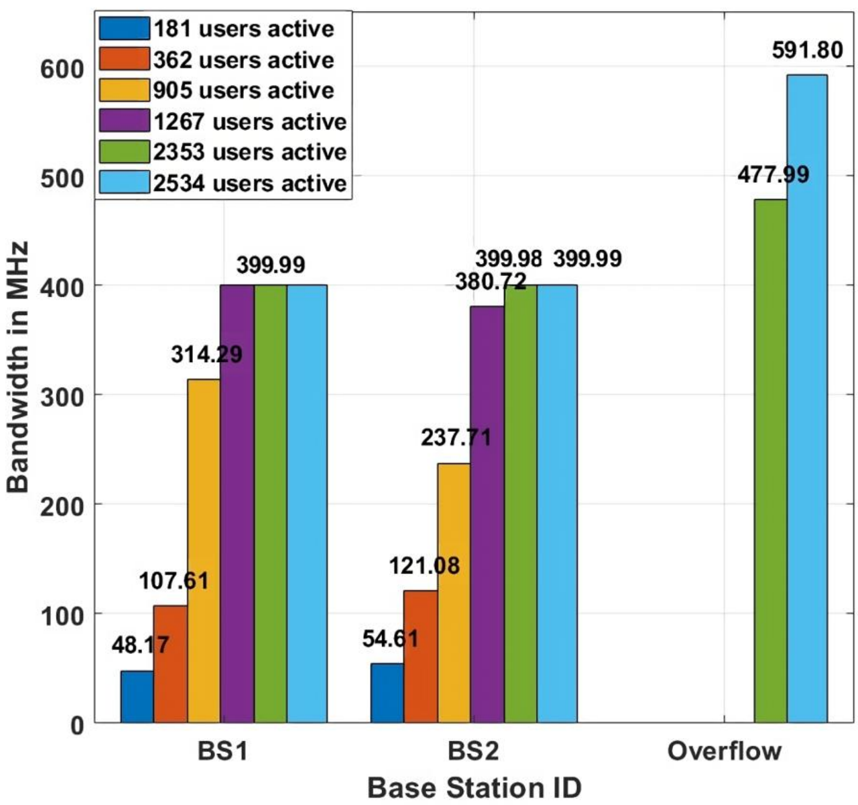

Fig. 19 Bandwidth Distribution for Minmum Cost Flow Algorithm – O2 scenario

Table 9 Minimum-Cost Flow Bandwidth – O2 scenario

# Users	181	362	905	1267	2353	2534
BW of BS1	48.17 MHz	107.61 MHz	314.29 MHz	399.99 MHz	399.99 MHz	399.99 MHz
BW of BS2	54.61 MHz	121.08 MHz	237.71 MHz	380.72 MHz	399.98 MHz	399.99 MHz
Average BS BW	51.39 MHz	114.345 MHz	276 MHz	390.36 MHz	399.99 MHz	399.99 MHz
Overflow BW	0 MHz	0 MHz	0 MHz	0 MHz	477.99 MHz	591.8 MHz

The overall comparison of the three algorithms shows that all three performed similarly when the number of users was 181, 362, 905, and 1267. However, as the number of users increased to 2353 and 2534, significant differences became apparent. Both the Minimum Cost Flow algorithm and the Hungarian algorithm demonstrated better throughput per user compared to the simple allocation method. The Hungarian algorithm, although having a higher number of overflow users, allocated the highest bandwidth, peaking at 399.99 MHz. The Minimum Cost Flow algorithm had slightly fewer overflow users than the Hungarian algorithm and achieved a similar peak bandwidth of 399.99 MHz. This indicates that while the Hungarian algorithm maximizes bandwidth utilization, the Minimum Cost Flow algorithm strikes a balance between reducing overflow users and maintaining high bandwidth distribution. In contrast, the simple allocation method lagged behind both in throughput per user and bandwidth allocation. Comparing the total given bandwidth across the three algorithms, the Minimum Cost Flow algorithm allocated the most, followed by the Hungarian algorithm and then the simple allocation method.

6. Energy Consumption Analysis

The final experiment examines the transmit-power-based consumption patterns of the examined allocation algorithms across different base stations and user-density scenarios. The power analysis is based on the conversion of simulated transmission power values from dBm to Watts using (2). Therefore, the reported values should be interpreted as a transmit-power-based comparative indicator across the examined allocation algorithms, rather than as a complete hardware-level base-station energy model. The same conversion formula, DeepMIMO-based scenario assumptions, user-density settings, and allocation outputs are used for all examined algorithms, allowing a consistent relative comparison.

$$\mathrm{P}\,(\mathrm{W}) \;=\; 10\text{^}((\mathrm{dBm} \,-\, 30)\,/\,10) \qquad (2)$$

Table 10 Power-consumption model and evaluation parameters.

Item	Setting / Description
Power quantity	Simulated transmit power values
Conversion formula	$\mathrm{P}\,(\mathrm{W}) = 10\text{^}((\mathrm{dBm} - 30)/10)$
Evaluation scenario	DeepMIMO O1 scenario
Compared algorithms	Simple allocation, Hungarian algorithm, Minimum Cost Flow algorithm
User-density cases	181, 362, 905, 1267, 2353, and 2534 users
Model scope	Transmit-power-based comparison only; circuit power, cooling, idle/sleep modes, and hardware-specific energy consumption are not included

This model is suitable for comparing the relative transmit-power behavior of the examined allocation algorithms under identical simulation assumptions. However, it does not represent a complete hardware-level 5G base-station energy model, since additional components such as circuit power, cooling consumption, power amplifier efficiency, and idle or sleep modes are not considered. A more detailed hardware-aware energy model is therefore outside the scope of the present work and is considered as future work.

Figures 20, 21 and 22 illustrate the power consumption for Algorithm 2, the Hungarian and Minimum Cost Flow Algorithm from the O1 Scenario for the same parameters and results from the second study.

Figure 20 suggests that for Algorithm 2, BS3 and BS4 exhibit relatively linear power consumption across different user densities, with values ranging from 27 to 246 Watts for BS3 and 19 to 230 Watts for BS4. BS5 shows lower power consumption compared to BS3 and BS4, ranging from 4 to 57 Watts. This suggests that BS5 is less loaded and underutilized in the O1 scenario with Algorithm 2.

In Figure 21, the Hungarian Algorithm, which focuses more on a balanced user allocation based on SNR, results in higher power consumption for BS3 (42 to 365 Watts) and BS4 (38 to 341 Watts). Again, BS5 shows lower consumption compared to BS3 and BS4 but higher consumption compared to the previous algorithm’s consumption, ranging from 2 to 148 Watts.

The Minimum Cost Flow Algorithm (Figure 22) showcased that since it focuses on the best per user throughput, it can be costly in terms of energy consumption, especially when the scale increases (2353 and 2534 users scenarios). It exhibits balanced power consumption compared to Algorithm 2 and the Hungarian Algorithm, with BS3 consuming 42 to 354 Watts, BS4 consuming 38 to 324 Watts, and BS5 consuming 2 to 113 Watts.

The simpler algorithm (Algorithm 2) offers a balanced approach with lower energy consumption, making it suitable for scenarios where power efficiency is critical. Its consistent power usage across different user densities highlights its capability to manage resources effectively. The lower power usage, especially at BS5, indicates that this base station may be underutilized, possibly due to a conservative user allocation strategy that does not maximize the station's capacity.

The Hungarian Algorithm prioritizes optimal SNR, resulting in higher power consumption. The significant power usage at BS3 and BS4 can be attributed to the need to maintain high-quality signals for a large number of users, especially under high-density conditions.

Finally, the Minimum Cost Flow Algorithm focuses on minimizing the cost associated with power usage while maintaining network quality. The steep power consumption at higher user scenarios reflects the algorithm's drive to maximize throughput, requiring substantial energy resources.

These results show that as user density increases, power consumption for all algorithms rises. This is expected as more users demand more resources and higher power to maintain network performance. Also, the underutilization of BS5 in the first algorithm (Figure 20) suggests potential for optimizing user allocation to balance the load more evenly across base stations, potentially reducing overall power consumption.

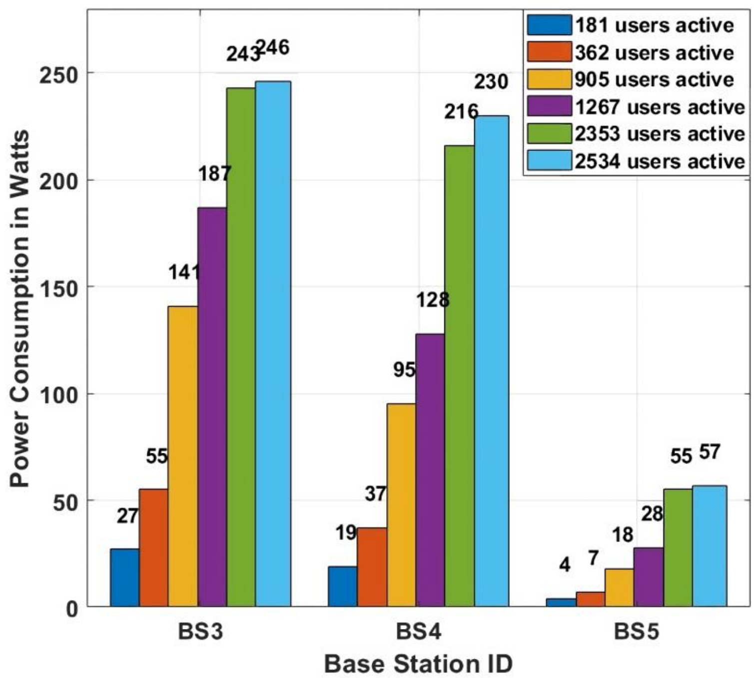

Fig. 20 Total Power Consumption Of BSs– Algorithm 2 – O1 scenario

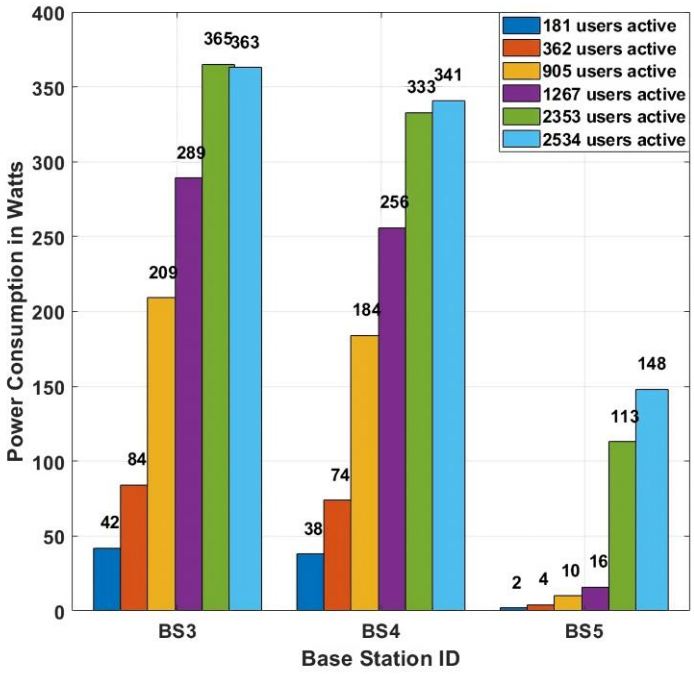

Fig. 21 Total Power Consumption Of BSs– Hungarian Algorithm – O1 scenario

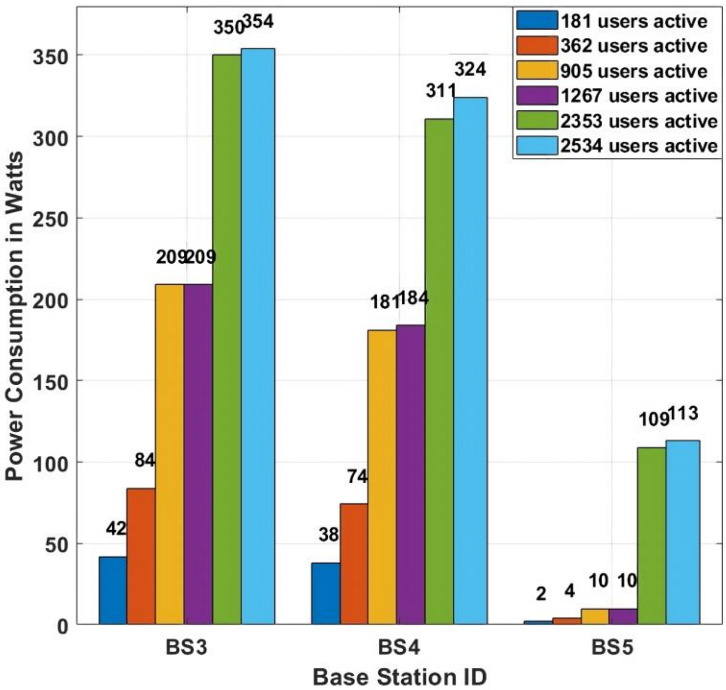

Fig. 22 Total Power Consumption Of BSs– Minimum Cost Flow Algorithm – O1 scenario

Figures 23, 24, and 25 illustrate the power consumption for the three algorithms under the O2 scenario, which requires significantly more power consumption due to the greater SNR this scenario provides.

Figure 23 suggests that Algorithm 2, BS1 and BS2 show significant power consumption over different user densities, with values ranging from 89 to 4433 Watts for BS1 and from 71 to 2437 Watts for BS2. Notably, Algorithm 2 peaks at 2353 users, not at 2534 users. This indicates that Algorithm 2 consumes a considerable amount of power in the O2 scenario, especially under high user densities.

In Figure 24, the Hungarian Algorithm, which focuses on adjusted user allocation based on SNR, results in slightly lower power consumption compared to Algorithm 2. BS1's power consumption ranges from 80 to 4411 Watts, while BS2's consumption ranges from 16 to 1519 Watts. The significant power usage at BS1 can be attributed to the need to maintain high-quality signals for a large number of users, especially under high-density conditions.

Figure 25 shows that the Minimum Cost Flow Algorithm, which focuses on maximizing throughput per user, results in power consumption similar to the Hungarian Algorithm. BS1 consumes between 80 and 4195 Watts, while BS2 consumes between 16 and 1709 Watts. This reflects that both the Hungarian and Minimum Cost Flow algorithms have comparable power consumption profiles.

The three algorithms exhibit distinct power consumption profiles reflective of their operational priorities and algorithmic efficiency in the high-SNR O2 scenario.

Algorithm 2 shows the highest power consumption, particularly under higher user densities, peaking at 4433 Watts for BS1 and 2437 Watts for BS2. This high power consumption indicates that while Algorithm 2 can handle substantial user loads, it may not be as efficient in terms of energy usage, making it less suitable for power-constrained environments.

The Hungarian Algorithm prioritizes optimal service quality, resulting in power consumption ranging from 80 to 4411 Watts for BS1 and from 16 to 1519 Watts for BS2. This higher power consumption is justified in high-value, performance-sensitive applications where maintaining high-quality signals is critical.

The Minimum Cost Flow Algorithm shows power consumption similar to the Hungarian Algorithm, with BS1 consuming between 80 and 4195 Watts and BS2 consuming between 16 and 1709 Watts. This algorithm balances throughput and power costs, making it suitable for environments where a balance between performance and cost-efficiency is needed.

The distinct power consumption patterns observed in each algorithm are a result of their specific design philosophies and operational goals. Algorithm 2's significant power consumption suggests a less efficient approach to managing user loads, focusing more on handling user density than conserving energy. The Hungarian Algorithm's higher power consumption is indicative of its commitment to optimizing SNR and service quality, which inherently requires more energy. In contrast, the Minimum Cost Flow Algorithm strikes a balance, optimizing both throughput and power costs, leading to moderate power consumption that scales with user density. The high-SNR O2 scenario amplifies these differences, as the increased SNR allows for higher data rates and better service quality but at the cost of greater power consumption. Each algorithm's performance in this scenario highlights its strengths and potential trade-offs, providing valuable insights into their suitability for different applications.

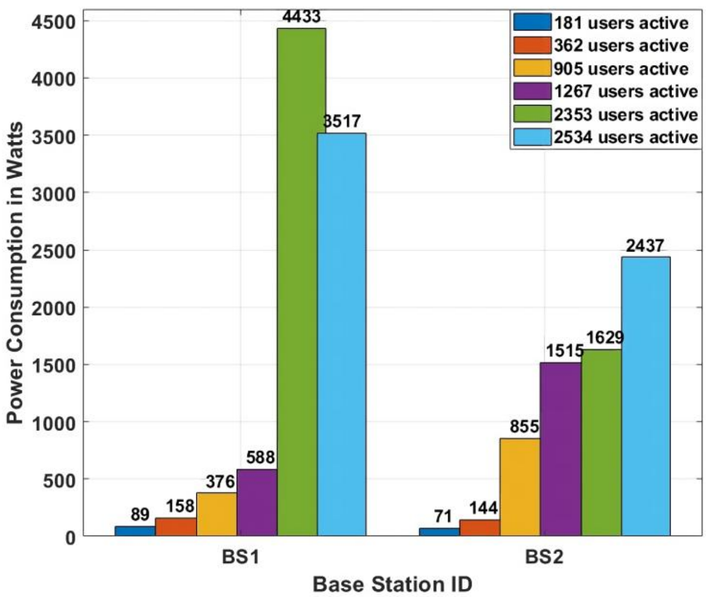

Fig. 23 User Power Consumption – Algorithm 2 – O2 scenario

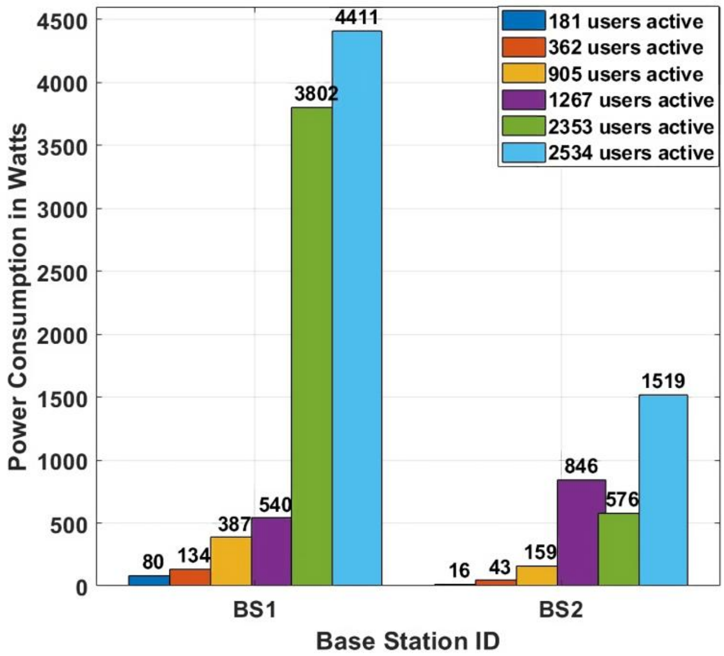

Fig. 24 User Power Consumption – Hungarian Algorithm – O2 scenario

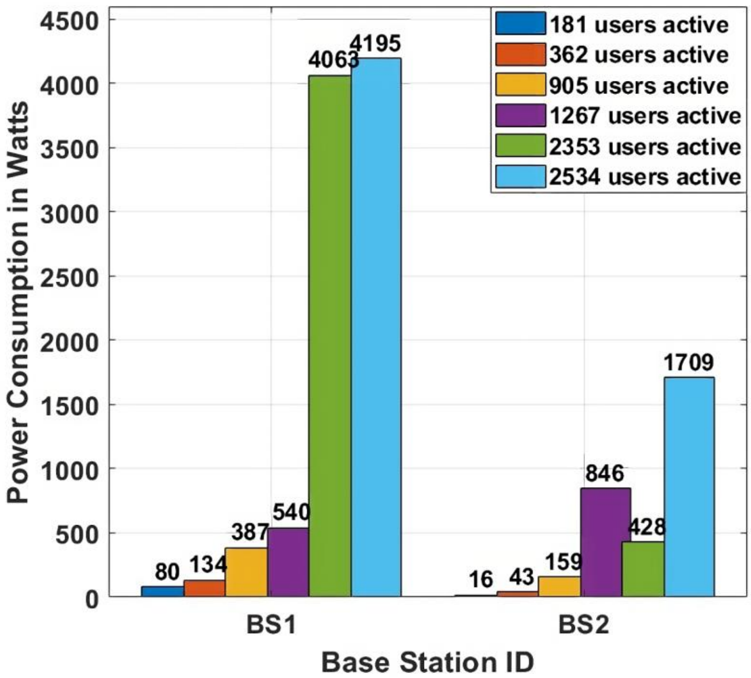

Fig. 25 User Power Consumption – Minimum Cost Flow Algorithm – O2 scenario

7. Conclusions

The consolidated research provides a path for future advancements in 5G technology, emphasizing the need for energy-efficient and adaptive resource allocation strategies. The studies' findings offer invaluable insights into developing sustainable 5G infrastructures capable of supporting the growing demands of wireless communication while minimizing environmental impact.

In summary, our research delved into resource allocation techniques in MM 5G networks, evaluating their performance through extensive simulations. First there was a focus on resource allocation for both single and multiple active base stations, revealing insights into throughput degradation with increasing user connections and the potential benefits of activating additional base stations. The study introduced the Hungarian algorithm and Minimum Cost Flow algorithm, highlighting their efficiency and scalability in optimizing user assignments. Further an analysis was provided for these resource allocation algorithms, emphasizing their strengths and limitations in different dynamic scenarios where each base station had a bandwidth capacity. While all algorithms performed comparably with a large user base, the Hungarian and Minimum Cost Flow algorithms demonstrated superior per-user throughput in scenarios with fewer users. Moreover, the Minimum Cost Flow Algorithm demonstrated the best resource allocation and bandwidth distribution out of all the algorithms used for the O1 scenario. For the O2 scenario the Hungarian Algorithm showed better resource allocation and bandwidth distribution from the simple algorithm.

The power consumption analysis reveals significant differences in the efficiency of the three algorithms under the O1 scenario. Algorithm 2 demonstrates the highest power consumption, making it a less efficient choice for energy-conscious network operations. The Hungarian and Minimum Cost Flow algorithms, while ensuring optimal signal quality and network performance, require moderate energy usage, making them more suitable for environments where a balance between performance and energy efficiency is needed. The Hungarian Algorithm's focus on optimal service quality results in higher power consumption, which could be justified in high-value, performance-sensitive applications.

In contrast, the Minimum Cost Flow Algorithm balances throughput and power costs, showing power consumption similar to the Hungarian Algorithm but still lower than Algorithm 2. Additionally, the proposition of implementing machine learning algorithms to address overflow issues presents a promising avenue for future exploration. These findings underscore the importance of aligning resource allocation algorithms with specific objectives and requirements, considering factors such as network scalability and dynamic resource allocation for maximizing system performance in MM 5G networks.

The conclusions of this study are limited to the examined assignment-based baselines, namely simple allocation, Hungarian assignment, and Minimum Cost Flow, under the selected DeepMIMO-based simulation assumptions. Therefore, the results should not be interpreted as demonstrating superiority over all existing resource allocation approaches, such as reinforcement learning, game-theoretic, scheduling-based, or heuristic load-balancing methods.

The reported results should be interpreted within the scope of the evaluated DeepMIMO scenarios and fixed simulation settings. Since the same scenario configuration, user settings, and resource constraints are used across the examined algorithms, the results provide a deterministic comparison under common assumptions. However, repeated-run statistical analysis, confidence intervals, variance estimation, statistical significance testing, sensitivity analysis, alternative user distributions, different traffic profiles, and additional DeepMIMO configurations are not included in the present evaluation. These aspects would require an additional robustness-oriented experimental campaign and are therefore considered important future work.

8. Future Work

For future work we will focus on enhancing the scalability and adaptability of our resource allocation algorithms to better meet the evolving needs of 5G networks. We plan to explore more advanced machine learning techniques to further optimize user-to-base station assignments dynamically. This includes developing models that can predict user demands and network conditions in real-time, allowing for more proactive and efficient resource distribution.

Additionally, we aim to investigate the integration of our algorithms with emerging technologies such as edge computing and network slicing to improve overall network performance and user experience. Expanding the scope of our simulations to include more varied and complex network scenarios will also be a priority, ensuring that our solutions are robust and versatile enough to handle the diverse challenges of future 5G and beyond networks. We will also extend our research to include the use of drones in 5G networks. Drones can act as mobile base stations, providing coverage in hard-to-reach areas or during events that require temporary network expansion.

Finally, we will delve into energy efficiency enhancements, striving to minimize the power consumption of base stations without compromising on performance, thereby contributing to more sustainable network operations.

Acknowledgment

The research project was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the “2nd Call for H.F.R.I. Research Projects to support Faculty Members & Researchers” (Project Number: 02440).

References

[1] Adnan, N. H. M., Rafiqul, I. M., & Alam, A. H. M. Z. (2016). Massive MIMO for fifth generation (5G): Opportunities and challenges. 2016 International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, Malaysia, 47–52. https://doi.org/10.1109/ICCCE.2016.23

[2] Bouras, C., Diasakos, D., Gkamas, A., Kokkinos, V., Pouyioutas, P., & Prodromos, N. (2023). Evaluation of resource allocation techniques in massive MIMO 5G networks. 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM), Istanbul, Turkiye, 1–6. https://doi.org/10.1109/WINCOM59760.2023.10322955

[3] Miuccio, L., Panno, D., & Riolo, S. (2021). A DNN-based estimate of the PRACH traffic load for massive IoT scenarios in 5G networks and beyond. *Computer Networks, 201*, Article 108608. https://doi.org/10.1016/j.comnet.2021.108608

[4] Riolo, S., Panno, D., & Miuccio, L. (2021). Modeling and analysis of tagged preamble transmissions in random access procedure for mMTC scenarios. *IEEE Transactions on Wireless Communications, 20*(7), 4296–4312. https://doi.org/10.1109/TWC.2021.3057932

[5] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. Naval Research Logistics Quarterly, 2(1–2), 83–97. https://doi.org/10.1002/nav.3800020109

[6] Goldberg, A. V. (1997). An efficient implementation of a scaling minimum-cost flow algorithm. Journal of Algorithms, 22(1), 1–29. https://doi.org/10.1006/jagm.1995.0805

[7] Athanasiou, G., Weeraddana, P. C., & Fischione, C. (2013). Auction-based resource allocation in millimeter wave wireless access networks. IEEE Communications Letters, 17(11), 2108–2111. https://doi.org/10.1109/LCOMM.2013.100713.131781

[8] Buzzi, S., D'Andrea, C., Fresia, M., Zhang, Y.-P., & Feng, S. (2021). Pilot assignment in cell-free massive MIMO based on the Hungarian algorithm. IEEE Wireless Communications Letters, 10(1), 34–37. https://doi.org/10.1109/LWC.2020.3020003

[9] Manini, M., Gueguen, C., Legouable, R., & Lagrange, X. (2019). Study of MIMO channel matrices correlation to optimize resource allocation algorithms in multi-user 5G. 2019 12th IFIP Wireless and Mobile Networking Conference (WMNC), Paris, France, 162–166. https://doi.org/10.23919/WMNC.2019.8881567

[10] Keshav, K., Pradhan, A. K., Srinivas, T., & Venkataram, P. (2021). Bandwidth allocation for interactive multimedia in 5G networks. 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 840–845. https://doi.org/10.1109/ICCES51350.2021.9488978

[11] Marwaha, S., Jorswieck, E. A., López-Pérez, D., Geng, X., & Bao, H. (2022). Spatial and spectral resource allocation for energy-efficient massive MIMO 5G networks. ICC 2022 – IEEE International Conference on Communications, Seoul, Korea, 135–140. https://doi.org/10.1109/ICC45855.2022.9838648

[12] Liu, H. (2022). Research on resource allocation and optimization technology in 5G communication network. 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 209–212. https://doi.org/10.1109/ICCECE54139.2022.9712674

[13] Sarder, M. A. I., Tajrian, F., Rafique, M., Anzum, M., & Shams, A. B. (2021). Configuring antenna system to enhance the downlink performance of high-velocity users in 5G MU-MIMO networks. 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), Rajshahi, Bangladesh, 1–6. https://doi.org/10.1109/ACMI53878.2021.9528218

[14] Prasad, K. N. R. S. V., Hossain, E., & Bhargava, V. K. (2017). Energy efficiency in massive MIMO-based 5G networks: Opportunities and challenges. IEEE Wireless Communications, 24(3), 86–94. https://doi.org/10.1109/MWC.2016.1500374WC

[15] Alkhateeb, A. (2019). DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications. In Proceedings of The Information Theory and Applications Workshop (ITA), San Diego, CA, USA, February 2019. https://doi.org/10.48550/arXiv.1902.06435

[16] Bouras, C., Diasakos, D., Gkamas, A., Kokkinos, V., Pouyioutas, P., & Prodromos, N. (2024). Dynamic bandwidth allocation in MIMO 5G networks. The 20th International Wireless Communications & Mobile Computing Conference (IWCMC 2024), Ayia Napa, Cyprus, 27–31 May 2024. https://doi.org/10.1109/IWCMC61514.2024.10592577

[17] Mughees, A., Tahir, M., Sheikh, M. A., Amphawan, A., Meng, Y. K., Ahad, A., & Chamran, K. (2023). Energy-efficient joint resource allocation in 5G HetNet using multi-agent parameterized deep reinforcement learning. Physical Communication, 61, Article 102206. https://doi.org/10.1016/j.phycom.2023.102206

[18] Korrai, P., Lagunas, E., Sharma, S. K., & Chatzinotas, S. (2023). Energy-efficient RAN slicing and beam selection for multiplexing of heterogeneous services in 5G mmWave networks. arXiv Preprint. https://doi.org/10.48550/arXiv.2306.07518

[19] Lin, X., Xu, F., Fu, J., & Wang, Y. (2022). Resource allocation for TDD cell-free massive MIMO systems. Electronics, 11(12), Article 1914. https://doi.org/10.3390/electronics11121914

[20] Liu, X., Zheng, J., Zheng, K., Liu, J., Taleb, T., & Shiratori, N. (2026). AoI minimization in heterogeneous MEC networks: A federated learning-assisted hybrid DRL and convex approach. *IEEE Transactions on Mobile Computing*. Advance online publication. https://doi.org/10.1109/TMC.2026.3674155

[21] Liu, X., Xu, J., Zheng, K., Zhang, G., Liu, J., & Shiratori, N. (2025). Throughput maximization with an AoI constraint in energy harvesting D2D-enabled cellular networks: An MSRA-TD3 approach. *IEEE Transactions on Wireless Communications, 24*(2), 1448–1466. https://doi.org/10.1109/TWC.2024.3509475

[22] Zheng, K., Luo, R., Liu, X., Qiu, J., & Liu, J. (2024). Distributed DDPG-based resource allocation for age of information minimization in mobile wireless-powered Internet of Things. *IEEE Internet of Things Journal, 11*(17), 29102–29115. https://doi.org/10.1109/JIOT.2024.3406044

[23] MathWorks. (n.d.). Matchpairs function documentation in MATLAB. Retrieved November 26, 2024, from https://www.mathworks.com/help/matlab/ref/matchpairs.html

[24] Remcom. (n.d.). Wireless Insite. Retrieved November 26, 2024, from https://www.remcom.com/wireless-insite-em-propagation-software

[25] Taha, A., Alrabeiah, M., & Alkhateeb, A. (2019). Deep learning for large intelligent surfaces in millimeter wave and massive MIMO systems. 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 1–6. https://doi.org/10.1109/GLOBECOM38437.2019.9013256

[26] DeepMIMO O1 scenario – Top view. DeepMIMO. https://www.deepmimo.net/scenarios/o1-scenario/

[27] DeepMIMO O2 scenario – Top view. *DeepMIMO*. https://www.deepmimo.net/scenarios/o2-scenario/

9. Statements & Declarations

9.1. Funding

The research project was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the “2nd Call for H.F.R.I. Research Projects to support Faculty Members & Researchers” (Project Number: 02440).

9.2. Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

9.3. Author Contributions

All authors contributed to the study conception and design. Nikolaos Prodromos and Damianos Diasakos were involved in the implementation of simulation environments. Vasileios Kokkinos and Apostolos Gkamas contributed to the design of the algorithms and the evaluation of the simulation results. Christos Bouras and Philippos Pouyioutas contributed to the evaluation of the simulation results and to the scientific coordination of the research work.

9.4. Data Availability

The datasets generated during and/or analyzed during the current study are not publicly available but are available from the corresponding author on reasonable request.