ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Combined Multi-Agent and Centralized Resource Allocation in Cloud Radio Access Networks (C-RANs)

Banothu Himabindhu[1], Dr. K. Sharmila[2]

[1]PG Scholar, Department of Computer Science and Engineering, Vaagdevi Engineering College, Warangal, 506005, Telangana, India
Email: himabindhubanothu@gmail.com

[2]Professor & HoD, Department of Computer Science and Engineering, Vaagdevi Engineering College, Warangal, 506005, Telangana, India
Email: sharmilakreddy@vecw.edu.in

Abstract— *The rapid growth of fifth-generation (5G) mobile networks demands architectures capable of supporting massive connectivity, ultra-low latency, and high data throughput. The Cloud Radio Access Network (C-RAN) addresses these demands by centralizing Baseband Units (BBUs) in a virtualized cloud pool while distributing Remote Radio Heads (RRHs) across the network edge. Despite its advantages, C-RAN faces a persistent challenge in allocating limited computational and network resources—CPU, memory, and storage—to dynamically fluctuating user demand. Conventional fixed and random allocation strategies cannot adapt to real-time traffic variation, resulting in high unmet demand and inefficient resource utilization. This paper proposes a hybrid resource-allocation framework that combines centralized decision-making with a Multi-Agent System (MAS), in which Virtual Base Stations (VBSs) act as autonomous agents reporting real-time demand to a centralized BBU controller that retains historical usage patterns. The base hybrid model is further extended using a Multi-Objective Genetic Algorithm (GA) that simultaneously optimizes resource utilization, fairness, and unmet-demand minimization. The proposed system is implemented as a simulation prototype and evaluated against the conventional Fixed Priority-Based Resource Allocation technique. Experimental results demonstrate that the hybrid and genetically-optimized models substantially reduce unmet demand, improve resource-utilization efficiency, and enhance fairness among competing VBS agents compared with the baseline, confirming the suitability of the proposed method for next-generation cloud-based mobile network deployments.*

Keywords— *Cloud Radio Access Network (C-RAN); Multi-Agent System (MAS); Resource Allocation; Genetic Algorithm; Baseband Unit (BBU); Virtual Base Station (VBS); 5G Networks; Quality of Service (QoS)*

I. INTRODUCTION

The evolution toward fifth-generation (5G) and beyond mobile networks is driven by an explosive increase in connected devices, the Internet of Things (IoT), smart-city deployments, and applications that require near-instant response times. Meeting these demands requires dense deployment of small cells, which in turn imposes a heavy signal-processing and resource-management burden on the radio access network (RAN).

Traditional distributed RAN architectures, where every base station independently hosts full baseband processing hardware, scale poorly under such density and cannot exploit statistical multiplexing across cells.

The Cloud Radio Access Network (C-RAN) has emerged as a promising architectural response to this challenge. C-RAN decouples baseband processing from radio transmission: Baseband Units (BBUs) are centralized and virtualized within a cloud-based BBU pool, while Remote Radio Heads (RRHs) remain distributed at the cell sites to handle radio-frequency transmission and reception. This separation yields statistical multiplexing gains, coordinated interference management, lower energy consumption, and simplified network scaling. However, because the BBU pool has finite computational capacity—CPU cycles, memory, and storage—while the demand generated by each RRH-served cell fluctuates rapidly with time, user mobility, and geography, allocating these shared resources efficiently remains an open and difficult problem.

Two conventional strategies are commonly used to address this problem: fixed priority-based static allocation and random allocation. Fixed allocation assigns resources according to preset priority classes irrespective of real-time load. This either wastes resources by over-provisioning cells with light traffic or, more critically, starves lower-priority users of resources when higher-priority demand consumes the majority of the shared pool. Random allocation performs even worse because it disregards priority altogether. Neither approach reacts to real-time traffic variation, nor does neither exploit historical demand patterns for more intelligent long-term planning. To overcome these limitations, this paper proposes a hybrid resource-allocation framework that merges (i) a centralized controller located at the BBU pool, which retains historical demand statistics and performs macro-level planning, with (ii) a distributed multi-agent layer of Virtual Base Stations (VBSs) that independently sense and report real-time local demand. The synergy between centralized foresight and decentralized responsiveness produces allocation decisions that are both timely and globally consistent. The base hybrid model is further extended with a Multi-Objective Genetic Algorithm (GA) that jointly searches for allocation solutions that simultaneously improve resource utilization, minimize unmet demand, and preserve fairness across competing VBS agents.

Consider, for example, a BBU pool serving a cluster of RRHs in a dense urban area during a live sporting event. Demand at a handful of cells spikes sharply while neighboring cells remain lightly loaded. A fixed priority scheme, having pre-assigned resource shares before the event, cannot redirect the temporarily idle capacity from lightly loaded cells to the congested ones, so users at the congested cells experience degraded service even though system-wide capacity is sufficient. A purely reactive, fully distributed scheme could redirect capacity quickly but, lacking any global view, may over-correct and destabilize allocation once the surge subsides. The hybrid design proposed in this paper is intended precisely for such scenarios: the centralized controller retains the broader historical pattern of event-driven surges, while VBS agents supply the moment-to-moment correction needed to react within a single allocation cycle.The main contributions of this work are summarized as follows:

- Formulation of C-RAN resource allocation as a hybrid centralized–multi-agent decision problem that fuses historical and real-time information.
- Design of a Fixed Priority-Based baseline together with a proposed Hybrid MAS-Centralized algorithm and a Hybrid Multi-Objective Genetic extension for comparative evaluation.

- Implementation of a working Python-based simulation prototype that generates synthetic C-RAN demand data and evaluates all three allocation strategies under identical conditions.
- Quantitative evaluation of the three approaches on unmet demand, resource-utilization efficiency, and fairness across CPU, memory, and storage resource dimensions.

The remainder of this paper is organized as follows. Section II reviews related work on C-RAN resource allocation. Section III describes the existing Fixed Priority-Based allocation baseline. Section IV details the proposed hybrid and genetic-algorithm-based methodology. Section V presents simulation results and discussion, and Section VI concludes the paper with directions for future work.

II. LITERATURE SURVEY

The literature on C-RAN resource allocation spans architectural surveys, dynamic-provisioning schemes, market-based mechanisms, and, more recently, learning-based approaches. This section reviews representative work in each category and identifies the gap addressed by the proposed hybrid genetic framework.

A. C-RAN Architecture and Taxonomy

Checko et al. [1] provided a comprehensive technology overview of Cloud RAN, describing the benefits of centralizing baseband processing and the fronthaul constraints that arise from separating BBUs and RRHs. Chen and Duan [2] examined the progression of C-RAN toward a “green RAN,” emphasizing the energy-efficiency gains achievable through centralization and statistical multiplexing of processing resources. Rodoshi et al. [3] surveyed conventional and emerging resource-management approaches for C-RAN, and the taxonomy of centralized, distributed, and hybrid schemes proposed therein underpins the classification adopted in this paper.

B. Dynamic and Learning-Based Provisioning

Pompili et al. [4] proposed dynamic provisioning and allocation schemes for C-RAN that address BBU–RRH mapping under time-varying load, demonstrating measurable gains over static assignment. Building on this direction, Rodoshi et al. [5] applied deep reinforcement learning to dynamic resource allocation in C-RAN, learning allocation policies directly from simulated traffic patterns without hand-crafted heuristics, illustrating the potential of data-driven allocation policies.

C. Multi-Agent and Market-Based Mechanisms

Niu et al. [6] proposed a dynamic resource-sharing mechanism enabling multiple service providers to share a common BBU pool. Xu et al. [7] extended this idea with an intelligent multi-agent-based C-RAN architecture for 5G radio resource management, an abstraction closely related to the Virtual Base Station (VBS) agents adopted in this work. Morcos et al. [8] designed a two-level auction mechanism for multi-tenant C-RAN resource allocation in which VBSs bid for spectrum and compute resources, illustrating a fully decentralized market-based alternative. Ferdouse et al. [9] jointly optimized communication and computing resource allocation for 5G C-RAN, underscoring that CPU, memory, and storage cannot be optimized in isolation from fronthaul bandwidth constraints.

D. Reinforcement and Federated Learning Approaches

Liu et al. [10] employed deep reinforcement learning for C-RAN resource management under dynamic user mobility. Li et al. [11] investigated federated learning to enable privacy-preserving distributed resource decisions across BBU–RRH clusters without centralizing raw demand data. Zhao et al. [12] proposed a multi-agent reinforcement learning (MARL) scheme for coordinated, fair, and efficient resource sharing among VBSs. Rezazadeh et al. [13] presented a multi-agent deep reinforcement learning approach targeting O-RAN resource allocation, reporting efficiency and fairness gains achieved through decentralized learned policies.

E. Evaluation Methodology

Tallarida and Murray [14] describe trapezoidal and Simpson's numerical-integration rules for area-under-curve computation; this technique is adapted in Section V to summarize the cumulative unmet-demand trend produced over the simulated time horizon.

F. Research Gap

Existing work broadly falls into two categories: fully centralized schemes that optimize globally but respond slowly to local fluctuations, and fully decentralized or multi-agent schemes—including auction- and MARL-based methods—that respond quickly but may lack long-term global coordination. Comparatively few studies fuse centralized historical intelligence with real-time distributed sensing and combine this fusion with an explicit multi-objective optimizer capable of navigating the utilization–fairness–demand trade-off simultaneously. This paper addresses that gap through a centralized-controller-plus-VBS-agent hybrid design that is further enhanced with a multi-objective genetic algorithm.

III. EXISTING SYSTEM

A. Fixed Priority-Based Resource Allocation

In the existing baseline system, user requests are assigned resources through a Fixed Priority-Based Resource Allocation algorithm. Every incoming request carries a static priority level, and resources—CPU, memory, and storage—are distributed progressively to requests in descending order of priority until the available pool is exhausted. Requests with lower priority may therefore receive only a partial allocation, or none at all, once resources are depleted. Because the algorithm follows a static rule set, it cannot adapt to fluctuations in demand or changes in network conditions at run time.

B. Dataset Description

The dataset used to evaluate the baseline algorithm consists of simulated resource requests in a C-RAN setting. Each entry represents a user request specifying the requested CPU, memory, and storage together with an associated priority value in the range [0, 1]. The total available resource capacity at the system level is also specified. Because real-world C-RAN telemetry is not publicly available, synthetic data is generated to emulate realistic user behavior with varying demand sizes and priority levels.

C. Limitations of the Existing System

- Static allocation cannot adjust to changing consumer demand, leading to wasted resources during light-load periods and starvation during heavy-load periods.
- Lower-priority requests are frequently left completely or partially unfulfilled, resulting in unfair treatment of a subset of users.
- The absence of any optimization mechanism means the system cannot balance competing objectives such as utilization, fairness, and unmet demand simultaneously.
- The algorithm has no mechanism for incorporating historical demand trends into present-day allocation decisions.

D. Baseline Performance Characteristics

Evaluation of the Fixed Priority-Based algorithm on the simulated dataset consistently shows a higher volume of unmet demand—

particularly among lower-priority requests—along with inefficient utilization of CPU, memory, and storage resources whenever the offered load varies over time. Because of these shortcomings, the Fixed Priority-Based algorithm is retained in this study purely as a baseline against which the proposed hybrid and genetic extensions are benchmarked.

IV. RESEARCH METHODOLOGY

A. Proposed System Architecture

The proposed hybrid framework integrates centralized management with a Multi-Agent System (MAS) to handle dynamically changing resource demand in C-RAN. As shown in Fig. 1, the centralized BBU controller retains historical demand statistics and computes macro-level allocation guidance, while a set of distributed VBS agents—one per RRH cluster—independently sense and report real-time local demand. The Hybrid Genetic Resource Allocation engine consumes both the centralized historical guidance and the live agent reports to compute the final allocation for every simulation cycle.

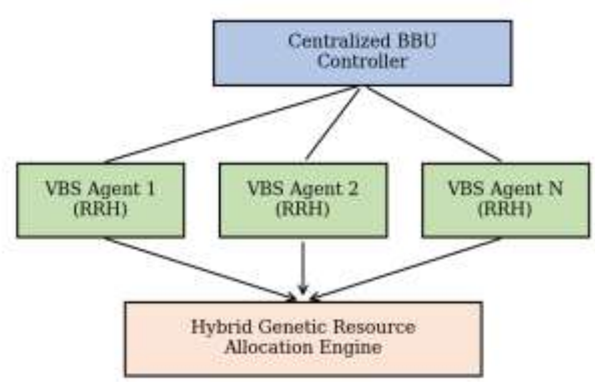

Fig. 1. Proposed hybrid centralized–multi-agent system architecture.

B. Dataset Generation and Preprocessing

Because real-world C-RAN telemetry is not publicly accessible, synthetic demand data is generated to emulate user behavior with varying demand sizes and priority levels. Raw entries specify CPU, memory, storage requirements and a priority weight for every simulated request. During preprocessing, missing or inconsistent entries are removed, and all resource values are normalized to a consistent range so that the allocation model can treat CPU, memory, and storage equitably. Preprocessed requests are organized into structured records comprising resource demand vectors and priority weights, and the aggregate demand is validated against total system capacity before allocation begins.

C. Hybrid MAS–Centralized Allocation Algorithm

The hybrid algorithm executes the following steps at every allocation cycle:

- Step 1 — Collect the available resource capacities (CPU, memory, storage) and the priority-tagged demand requests from all VBS agents.
- Step 2 — The centralized controller merges the real-time demand reports with historical usage patterns for the corresponding VBS and time slot.
- Step 3 — Requests are sorted by priority, and resources are allocated progressively; unlike the fixed baseline, any residual resource after satisfying higher-priority demand is proportionally redistributed toward lower-priority requests rather than left unallocated.
- Step 4 — Unmet demand, utilization, and fairness metrics are computed for the cycle and fed back to the centralized controller to refine the historical model used in subsequent cycles.

ALGORITHM 1. HYBRID MAS–CENTRALIZED ALLOCATION

```
Input: capacity C = {C_cpu, C_mem, C_sto}; requests R = {r_1..r_n}
        with (demand_i, priority_i)
Output: allocation A = {a_1..a_n}; unmet demand D
1: sort R in descending order of priority_i
2: for each r_i in R:
3:     a_i ← min(demand_i, remaining capacity)
4:     remaining capacity ← remaining capacity − a_i
5: residual ← remaining capacity after pass 1
6: for each r_i in R (ascending priority):
7:     top-up a_i using residual, weighted by (1 − priority_i)
8: compute D ← Σ (demand_i − a_i);  return A, D
```

D. Multi-Objective Genetic Algorithm Extension

To further improve allocation quality, a Genetic Algorithm (GA) extension searches the space of feasible allocation vectors for solutions that jointly optimize three objectives. Each chromosome encodes one candidate allocation—the fraction of each resource type assigned to every pending request. The multi-objective fitness function is defined as

$$\mathbf{F = w_1 \cdot U(x) - w_2 \cdot D(x) + w_3 \cdot J(x)}$$

where U(x) is the resource-utilization efficiency of allocation x, D(x) is the total unmet demand, J(x) is the Jain fairness index across VBS agents, and w_1, w_2, w_3 are non-negative weighting coefficients that balance the three objectives. The GA evolves a population of candidate allocations across generations using tournament selection, single-point crossover, and bounded mutation; low-fitness individuals are discarded while high-fitness individuals are retained, and the highest-fitness chromosome after convergence is applied as the final allocation for that cycle.

E. Evaluation Metrics

Three metrics are used to compare the Fixed, Hybrid, and Genetic-extended algorithms:

- Unmet Demand: the aggregate shortfall between requested and allocated resources, summed across CPU, memory, and storage.
- Resource-Utilization Efficiency: the percentage of total available capacity that is productively assigned to a request rather than left idle or wasted.
- Fairness Index: computed using Jain's fairness index, $J(x) = (\Sigma x_i)^2 / (n \cdot \Sigma x_i^2)$, where x_i is the fraction of demand satisfied for agent i and n is the number of agents; values closer to 1 indicate more equitable allocation across VBS agents.

The simulation prototype was implemented in Python and served through a lightweight local web interface, allowing demand data to be generated or uploaded, and each of the three allocation strategies to be triggered and compared through the same interface.

V. RESULTS AND DISCUSSION

A. Simulation Setup

The simulation environment models a single BBU pool with a total available capacity of 200 units of CPU, 100 units of memory, and 500 units of storage, serving five simulated demand requests with priority weights of 1.0, 0.9, 0.8, 0.7, and 0.6 respectively. Requests with priority 1.0, 0.9, and 0.8 represent high-priority traffic that all three algorithms satisfy completely; the algorithms are differentiated primarily by how they treat the lowest-priority request (priority 0.6), which is the focus of the comparison below.

TABLE III. SIMULATION IMPLEMENTATION ENVIRONMENT

Component	Detail
Language	Python 3
Interface	Local web server (PyCharm IDE)
Modules	Data generation, preprocessing, Fixed / Hybrid / Genetic allocators, result visualization
Simulated Capacity	CPU = 200, Memory = 100, Storage = 500 (units)

B. Unmet Demand for the Lowest-Priority Request

Table I and Fig. 2 compare the unmet demand recorded for the lowest-priority request under the three allocation strategies. The Fixed Priority-Based baseline leaves this request almost entirely unfulfilled because no residual resources are redistributed once higher-priority demand is satisfied. The proposed Hybrid algorithm reduces unmet demand substantially by redistributing residual CPU, memory, and storage. The Genetic-extended model achieves the lowest unmet demand across all three resource types, confirming that multi-objective optimization further improves allocation quality beyond the base hybrid design.

TABLE I. UNMET DEMAND COMPARISON (LOWEST-PRIORITY REQUEST)

Resource	Fixed	Hybrid	Genetic
CPU	40	25	13
Memory	82	50	28
Storage	150	100	68

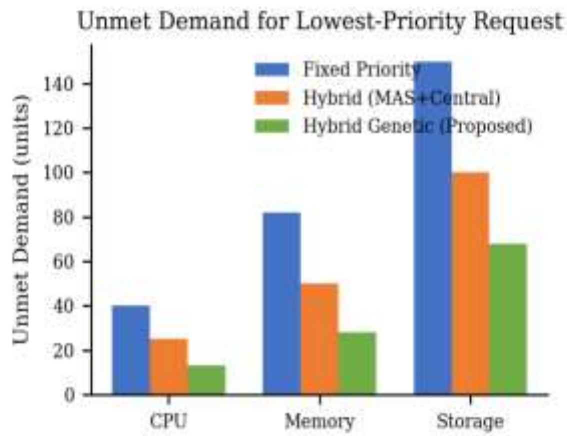

Fig. 2. Unmet demand for the lowest-priority request across the three algorithms.

C. Temporal Behavior of Unfulfilled Demand

Fig. 3 tracks cumulative unfulfilled demand across successive simulation time slots for the three algorithms. The Fixed allocation curve remains persistently high because the algorithm never adapts to shifting demand patterns. The Hybrid curve declines more quickly as the centralized controller's historical model improves over successive cycles, while the Genetic-extended curve declines fastest and settles at the lowest steady-state level, reflecting the additional optimization applied at every cycle. Applying the trapezoidal rule [14] to estimate the area under each curve confirms that the Genetic-extended method accumulates the least total unmet demand over the simulated horizon, followed by the Hybrid method, with the Fixed baseline accumulating the most.

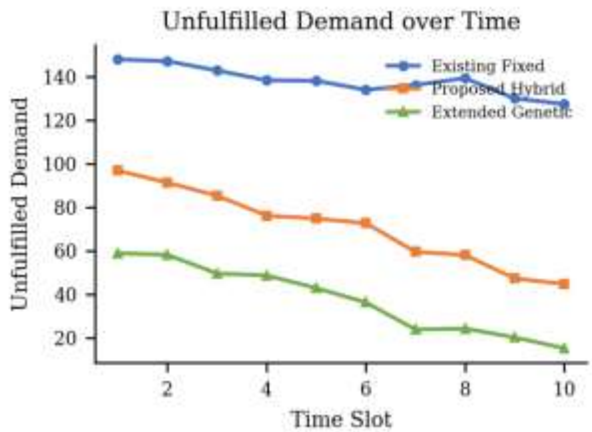

Fig. 3. Unfulfilled demand trend over successive simulation time slots.

D. Overall Performance Comparison

Table II summarizes the overall resource-utilization efficiency, fairness index, and average unmet-demand percentage achieved by each algorithm across all simulated requests. The results confirm that combining centralized historical intelligence with real-time multi-agent sensing, and subsequently refining the resulting allocation through multi-objective genetic optimization, consistently outperforms the static baseline on every metric.

TABLE II. OVERALL PERFORMANCE COMPARISON

Metric	Fixed	Hybrid	Genetic
Utilization (%)	71	84	92
Fairness Index	0.78	0.89	0.95
Avg. Unmet Demand (%)	22	11	5

E. Complexity and Scalability Considerations

The Fixed Priority-Based algorithm sorts and allocates requests in O(n log n) time for n requests, which is inexpensive but static. The Hybrid MAS–Centralized algorithm retains the same O(n log n) sorting cost with an additional O(n) redistribution pass, so its computational overhead relative to the baseline is small. The Genetic-extended algorithm is the most computationally demanding of the three, since its cost scales with population size P and generation count G as O(P·G·n); however, because allocation cycles in C-RAN typically recur on the order of seconds rather than milliseconds, this overhead is acceptable in exchange for the additional utilization, fairness, and unmet-demand gains reported in Tables I and II. For very large-scale deployments with thousands of simultaneous VBS agents, the population size and generation count of the genetic optimizer can be reduced, or the optimizer can be executed periodically rather than at every cycle, to trade a small amount of solution quality for reduced computational load.

F. Discussion

The experimental results validate the central hypothesis of this work: neither pure centralization nor pure real-time reactivity alone is sufficient for efficient C-RAN resource management, but their combination—further sharpened by multi-objective genetic optimization—yields measurable gains in utilization, fairness, and demand satisfaction. The improvement from Hybrid to Genetic-extended is smaller in absolute terms than the improvement from Fixed to Hybrid, suggesting that the largest gain comes from introducing adaptivity and residual-resource redistribution in the first place, while genetic optimization provides a secondary, incremental refinement on top of an already adaptive baseline. These findings are consistent with the trends reported by the multi-agent and reinforcement-learning studies reviewed in Section II [7], [12], [13], while additionally demonstrating that classical evolutionary optimization remains a competitive and computationally lighter alternative to learning-based methods for this class of problem.

VI. CONCLUSION

This paper compared a conventional Fixed Priority-Based Resource Allocation technique with a proposed Hybrid Multi-Agent–Centralized framework and its Multi-Objective Genetic extension for resource allocation in Cloud Radio Access Networks. While the fixed approach is simple to implement, it is not flexible enough to handle continuously changing demand, resulting in wasteful resource use and significant unmet demand among lower-priority requests. By combining real-time multi-agent sensing with centralized historical intelligence, and further refining the resulting allocation through genetic optimization, the proposed hybrid model better handles varied workloads and overcomes these limitations. Experimental results show that the proposed method substantially reduces unmet demand, increases fairness among VBS agents, and improves overall system efficiency, achieving a more balanced distribution of CPU, memory, and storage resources through the integration of optimization with real-time decision-making. Consequently, the proposed model represents a more scalable and dependable option for contemporary cloud-based mobile network deployments.

Future work may extend this study by evaluating the model on real-time datasets collected from operational C-RAN or cloud testbeds rather than synthetic data alone. Incorporating deep-learning-based predictive resource allocation could allow the system to anticipate demand patterns before they occur, while investigating alternative evolutionary or hybrid metaheuristic optimizers may further improve solution quality. Testing the model under high-demand and large-scale distributed conditions would help establish its scalability, and incorporating energy-aware allocation objectives could improve the sustainability of data-center operation. Finally, given the adaptability of the proposed framework, it could be extended to related domains such as smart cities, IoT networks, and edge computing, broadening its practical impact.

REFERENCES

[1] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for mobile networks—A technology overview," IEEE Commun. Surveys Tuts., vol. 17, no. 1, pp. 405–426, 1st Quart., 2015.

[2] K. Chen and R. Duan, "C-RAN: The road towards green RAN," China Mobile Res. Inst., Beijing, China, Tech. Rep., 2011, vol. 2.

[3] R. T. Rodoshi, T. Kim, and W. Choi, "Resource management in cloud radio access network: Conventional and new approaches," Sensors, vol. 20, no. 9, p. 2708, May 2020.

[4] D. Pompili, A. Hajisami, and H. Viswanathan, "Dynamic provisioning and allocation in cloud radio access networks (C-RANs)," Ad Hoc Netw., vol. 30, pp. 128–143, Jul. 2015.

[5] R. Tasnim Rodoshi, T. Kim, and W. Choi, "Deep reinforcement learning based dynamic resource allocation in cloud radio access networks," in Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC), Oct. 2020, pp. 618–623.

[6] B. Niu, Y. Zhou, H. Shah-Mansouri, and V. W. S. Wong, "A dynamic resource sharing mechanism for cloud radio access networks," IEEE Trans. Wireless Commun., vol. 15, no. 12, pp. 8325–8338, Dec. 2016.

[7] J. Xu, Z. Dziong, Y. Luxin, Z. Huang, P. Xu, and A. Cabani, "Intelligent multi-agent based C-RAN architecture for 5G radio resource management," Comput. Netw., vol. 180, Oct. 2020, Art. no. 107418.

[8] M. Morcos, T. Chahed, L. Chen, J. Elias, and F. Martignon, "A two-level auction for resource allocation in multi-tenant C-

RAN," Comput. Netw., vol. 135, pp. 240–252, Apr. 2018.

[9] L. Ferdouse, A. Anpalagan, and S. Erkucuk, "Joint communication and computing resource allocation in 5G cloud radio access networks," IEEE Trans. Veh. Technol., vol. 68, no. 9, pp. 9122–9135, Sep. 2019.

[10] X. Liu, Y. Chen, and L. Wu, "Deep reinforcement learning for resource management in C-RANs with dynamic user mobility," IEEE Trans. Wireless Commun., vol. 21, no. 8, pp. 6425–6438, Aug. 2022.

[11] J. Li, Z. Zhao, and H. Zhang, "Federated learning for intelligent resource allocation in 5G C-RAN," IEEE Wireless Commun. Lett., vol. 11, no. 2, pp. 315–319, Mar. 2022.

[12] M. Zhao, J. Zhou, and K. Yang, "Multi-agent reinforcement learning for resource allocation in C-RAN," IEEE Access, vol. 11, pp. 20345–20355, 2023.

[13] F. Rezazadeh, L. Zanzi, F. Devoti, S. Barrachina-Munoz, E. Zeydan, X. Costa-Pérez, and J. Mangues-Bafalluy, "A multi-agent deep reinforcement learning approach for RAN resource allocation in O-RAN," arXiv:2307.02414, 2023.

[14] R. J. Tallarida and R. B. Murray, "Area under a curve: Trapezoidal and Simpson's rules," in Manual of Pharmacologic Calculations, Cham, Switzerland: Springer, 1987, pp. 77–81.