Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 10s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Advanced Meta heuristic Strategies for Load Balancing in 5G C-RAN Environments

CH. Srilakshmi Prasanna[a], Dr. S. Zahoor-ul-Huq[b], Dr. P. Chenna Reddy[c]

[a]Research Scholar, Jawaharlal Nehru Technological University Anantapur, Ananthapuramu

[b]G. Pulla Reddy Engineering College, Kurnool, Affiliated to Jawaharlal Nehru Technological University Anantapur, Ananthapuramu

[c]JNTUA College of Engineering, Ananthapuramu, Constituent college of Jawaharlal Nehru Technological University Anantapur, Ananthapuramu

Email ID: sirikeshava9@gmail.com

Email ID: szahoor@gmail.com

Email ID: pcreddy1@rediffmail.com

Cite this paper as: CH. Srilakshmi Prasanna, Dr. S. Zahoor-ul-Huq, Dr. P. Chenna Reddy, (2025) Advanced Meta heuristic Strategies for Load Balancing in 5G C-RAN Environments. *Journal of Neonatal Surgery*, 14 (10s), 237-245.

ABSTRACT

With the rapid expansion of internet usage and continuous technological progress, mobile network operators are compelled to enhance their investments in network infrastructure. Emerging technologies such as Cloud Radio Access Networks (C-RAN) and Software Defined Networking (SDN) are increasingly viewed as viable solutions to lower operational costs and improve scalability in fifth-generation (5G) mobile networks. A typical base station comprises two critical components: the baseband unit (BBU) and the remote radio head (RRH). Variations in data traffic can lead to network inefficiencies, including issues like call drops and call blocking. As traffic patterns fluctuate, system performance may degrade if not properly managed. To address this, self-optimizing network strategies are essential for redistributing the load from heavily burdened eNodeBs, which experience high call blocking rates, to underutilized ones with spare capacity. The primary goal of these self-organizing networks is to balance the traffic load and minimize call blocking incidents. In this work, an improved version of the Cat Swarm Optimization (CSO) algorithm—referred to as Enhanced Cat Swarm Optimization (ECSO)—is introduced. Managed by the host controller, ECSO identifies the optimal BBU-RRH pairings by assessing quality-of-service (QoS) metrics from various configurations. The optimization process evaluates each user connection by analyzing QoS data for every potential BBU-RRH combination. Simulation outcomes indicate that ECSO outperforms existing Particle Swarm Optimization (PSO) and standard CSO methods by reducing blocking probability by 10%, increasing throughput by 8%, and decreasing response time by 7%.

1. INTRODUCTION

Mobile network operators are facing a significant surge in data traffic due to the growing number of mobile device users and the increasing demand for high-speed data services, particularly in multimedia applications. To accommodate this rising demand, wireless access technologies have evolved from third-generation (3G) to fifth-generation (5G) networks, providing widespread service access for a variety of mobile devices. Two major strategies shaping next-generation wireless networks are the deployment of small cells and the exploration of high-frequency radio bands. As smart devices proliferate and diverse air interface standards emerge, mobile internet traffic continues to rise. Consequently, operators are under pressure to increase both capital investments and operational expenditures to meet user expectations. However, the average revenue per user often falls short of covering these rising infrastructure costs. Addressing the challenges posed by increased traffic requires highly scalable connectivity solutions, especially for the expanding landscape of IoT-based mobile applications. Performance enhancements can be achieved through advanced technologies such as multiple-input multiple-output (MIMO) systems, cloud radio access networks (C-RAN) [1], and millimeter-wave communication, all of which offer greater scalability, flexibility, and efficiency. Another approach involves breaking down large cells into smaller units and reusing frequencies more intensively, which, although improving coverage, can lead to higher interference levels and increased network management and operational costs. Fifth-generation wireless systems are designed to support an ever-growing number of mobile devices with seamless service access. To achieve the vision of 5G, two critical technologies have been identified

: ultra-dense small cell deployment and distributed computing. Small cells, in particular, enhance the received signal quality for users by reducing the transmission distance, thereby improving bits-per-second (BPS) performance [2].

The 5G network architecture integrates two key technologies to address modern communication challenges: (i) ultra-dense small cell deployment, which minimizes the distance between users and their respective base stations, thereby improving signal reception, and (ii) cloud computing, which helps mitigate inter-cell interference often caused by dense cell layouts. By utilizing Cloud Radio Access Networks (Cloud-RAN), base station functionalities are virtualized, allowing for dynamic load distribution and enhancing both energy and spectrum efficiency. This approach effectively addresses critical concerns such as resource allocation, security, system monitoring, and load balancing. Among these, load balancing plays a crucial role as it directly influences computational resource utilization and fairness in service delivery. Improved load distribution contributes to enhanced Quality of Service (QoS) by optimizing resource management and minimizing response times. Since users submit tasks with varying resource demands, cloud data centers need adaptive strategies to handle fluctuating workloads and prevent QoS degradation [3-5]. Load balancing techniques often employ virtual machine (VM) scheduling, where VMs are allocated to different hosts to ensure an even workload. In metaheuristic algorithms, a centralized controller manages operations like mutation, crossover, and swapping processes to optimize VM placement, aiming for efficient host assignment and reduced system load.

Cloud-RAN (Radio Access Network) represents a modern approach to base station functionality by leveraging cloud computing to virtualize key operations. Instead of processing baseband and upper-layer functions locally at individual wireless access points, these tasks are handled by centralized, general-purpose processors within a cloud infrastructure. In this setup, the access points are simplified to primarily perform radio functions, eliminating the need for traditional, fully-equipped base station components. This results in an alternative network architecture where low-cost units, known as Remote Radio Heads (RRHs), manage wireless access and are centrally controlled by a cloud-based management system, often referred to as the control unit (CU). At its core, Cloud-RAN is considered a key element for implementing network performance virtualization (NPV) techniques. Through high-capacity backhaul links connecting base stations and centralized processing units, Cloud-RAN facilitates joint encoding and decoding across multiple cells, improving overall network efficiency. As 5G networks increasingly rely on smaller cell sizes to support higher data rates, Cloud-RAN plays a critical role in addressing inter-cell interference and ensuring efficient system performance [6-8]. Additionally, Coordinated Multi-Point (CoMP) transmission, commonly known as a multi-input multi-output (MIMO) system, further enhances network capabilities by improving coverage and throughput. By optimizing load balancing within Cloud-RAN architectures, service providers can significantly lower both capital and operational expenditures in multicast network environments. In 5G Cloud-RAN systems, techniques such as statistical multiplexing and dynamic load balancing are key to maximizing efficiency and performance. This study introduces an improved cat swarm optimization algorithm specifically designed to enhance load balancing and reduce call blocking in Cloud-RAN environments [9].

2. LITERATURE SURVEY

Researchers [10-12] have developed an architectural framework for 5G Cloud-RAN that integrates computational intelligence with software-defined networking (SDN). Within this system, dynamic mapping between Baseband Units (BBUs) and Remote Radio Heads (RRHs) was optimized through load balancing techniques based on both objective and subjective Quality of Experience (QoE) predictions. This adaptive mapping was crucial for supporting applications like ultra-high-definition (UHD) video streaming. The intelligent SDN-based system forecasted the users' Mean Opinion Score (MOS) during UHD video transmissions over the Cloud-RAN infrastructure. Load balancing policies were implemented within the SDN framework to enhance the QoE for video streaming services, while the BBU pool was organized through SDN control components. Simulation outcomes demonstrated significant improvements, with performance gains of 130% in scenarios involving new BBU activation, and 60% without additional activation.

The authors in [13] introduced an inter-BBU load balancing mechanism that relied on virtual base station (VB) live migration. Their approach considered two BBUs within a single pool, enabling timely resource allocation with predicted completion times and migrations driven by utilization rates. This strategy effectively addressed load imbalances by increasing BBU utilization through efficient resource distribution and load balancing. The use of live migration, combined with cloud computing, minimized both server downtime and migration delays.

In another study [14], researchers proposed a Heterogeneous Radio Access Network (H-RAN) architecture that incorporated newly designed edge nodes and local processing units within the RRHs. These enhancements led to greater network capacity and reduced latency. In the H-RAN model, centralized BBU control and processing tasks were handled using multiple VBs situated in the local processing units. A dynamic base station switching algorithm was introduced to optimize load distribution and power consumption. Performance evaluations between Cloud-RAN and H-RAN, under identical traffic conditions, demonstrated that the addition of local processing units and an intermediate edge layer at the RRH reduced both call blocking probability and waiting times in comparison to conventional Cloud-RAN systems.

The authors [15] explored the use of swarm intelligence algorithms to improve load balancing and minimize call blocking within BBU networks. Their implementation of the Particle Swarm Optimization (PSO) algorithm began with a group of candidate solutions, each representing a particle with random speed and position. Communication among these particles helped identify both local and global optima for balancing loads across different BBUs and RRHs. The simulation results showed that this approach effectively minimized blocked calls and achieved more uniform load distribution, thereby improving overall Quality of Service (QoS).

Additionally, an edge computing framework [16] leveraging cloud computing principles was proposed to reduce latency in mobile networks. This Mobile Edge Computing (MEC) solution aimed to offload computational tasks from resource-limited edge devices to edge servers within the network. A key challenge addressed was the design of an offloading strategy for multi-user, resource-constrained environments. In industrial applications, task migration from edge devices was optimized using PSO algorithms to allocate workloads to edge servers with higher energy efficiency and reduced delays. The multi-objective optimization strategy focused on minimizing energy consumption, task execution costs, and response times. Each particle in the PSO algorithm evaluated the total cost associated with task offloading to different MEC servers. Simulations demonstrated that the proposed PSO-based strategy outperformed both genetic algorithms and simulated annealing techniques, achieving lower server delay, better energy consumption balance, and more efficient resource allocation.

3. SYSTEM ARCHITECTURE

The primary goal of this study is to address the issue of load imbalance and minimize call blocking within the network. To achieve this, an improved Cat Swarm Optimization (CSO) algorithm is introduced to enhance load balancing efficiency. The system architecture of the proposed Cloud-RAN model is depicted in Figure 1.

In the Cloud-RAN environment, cloud computing plays a crucial role in optimizing the performance of the 5G network infrastructure. The system architecture is composed of virtual Baseband Units (BBUs), physical BBUs, cloud controllers, Remote Radio Heads (RRHs), and general-purpose processors (GPPs). Digital IQ data is transmitted to RRHs via a standardized public radio interface. The BBUs, which are pooled together, are interconnected and managed by a centralized host controller. This host manager is responsible for monitoring the load distribution across the BBUs and selecting the optimal BBU-RRH pairing to ensure efficient resource utilization.

Each BBU manages multiple sectors simultaneously, with each sector comprising several RRHs. The BBUs' resources are fully utilized to serve the RRHs assigned within their sectors. An RRH is dedicated to one sector at a time, and the load on a BBU is determined by the number of active users it supports. The system imposes a hard capacity limit for each sector, restricting the number of active users based on the underlying hardware or software capabilities.

The BBU pool functions [19] as a centralized data processing hub, hosting shared and virtualized units that allocate resources dynamically according to fluctuating user demand. These virtual BBUs are interconnected and can support numerous RRHs simultaneously. Communication between BBUs is facilitated through the X2 interface, while connectivity with the core mobile network is established via the S1 interface, also known as the backhaul link. Real-time virtualization is employed on high-performance processors within the BBUs, and these BBUs operate on virtual machines hosted in a cloud data center, utilizing physical computing cores to deliver scalable and efficient network services [20].

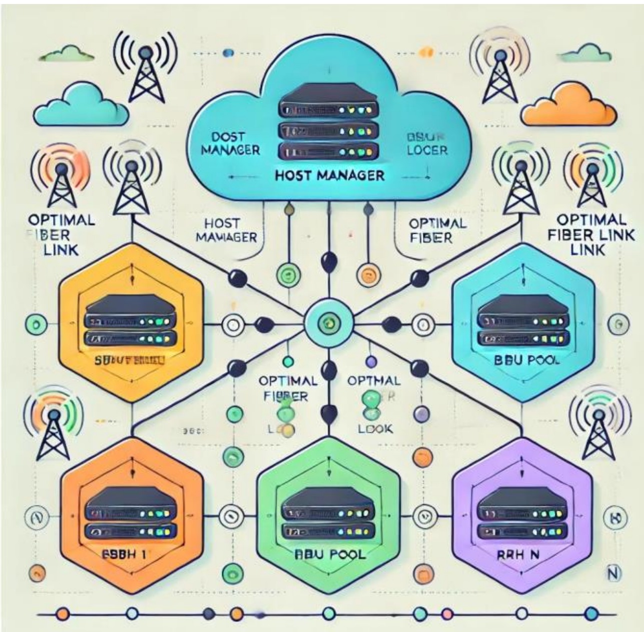

Figure 1: Cloud Radio Access Network System Architecture

In 5G networks, efficient load balancing is essential for delivering real-time, carrier-grade services within a cloud-RAN environment. To address this, an improved Cat Swarm Optimization (CSO) algorithm has been introduced. The eNodeB control center is responsible for managing all Layer-2 (L2) control functions and maintains data for a significant number of users. Various applications operate within the L2 layer, all managed by a centralized control unit. Based on user requirements, appropriate frequency bands are assigned. These requirements are represented by the user equipment (UE) context [18].

Instead of dedicating a separate physical machine to each cell in the BBU pool of cloud-RAN, processing tasks are handled by a group of virtual machines hosted on centralized servers. Each virtual machine is allocated its own local cache, which stores the UE context, ensuring quicker data access and reducing latency. Effective load balancing hinges on properly assigning each UE context to the appropriate virtual machine. This ensures minimized processing delays and faster response times. Several sophisticated and intelligent load-balancing algorithms have been developed for distributed systems and network infrastructures.

The main focus of this study is to analyze and understand the nature of actual network traffic loads in terms of UE contexts to optimize cloud-RAN deployment in 5G networks. To tackle the challenges in load balancing, an Advanced Cat Swarm Optimization (ACSO) algorithm has been applied, as illustrated in Algorithm 1.

Algorithm 1: Load Balancing and Traffic Allocation for RRHs in Cloud-RAN

Step 1: Start the process.

Step 2: For each Remote Radio Head (RRH), calculate its total capacity, denoted by C.

Step 3: Determine the total traffic load I generated in each time period (e.g., per year).

Step 4: Select an RRH, referred to as RRHj, for evaluation.

Step 5: Compare the capacity Cj of RRHj with its traffic load Ij:

- If Cj ≥ Ij (the RRH is underloaded or balanced), → Proceed to Step 6.
- Else, → Reallocate excess users to other RRHs with available capacity, then return to Step 4.

Step 6: Assign the traffic load to RRHj.

Step 7: Forward the traffic for processing at the Baseband Unit (BBU).

Step 8: End the process.

1. Advanced Meta Heuristic Algorithm

In the proposed algorithm, the two primary behaviors observed in cats are tracking and seeking are used as the foundation for two distinct sub-models. By combining these two behavioral modes in a user-defined ratio, the Advanced Cat Swarm Optimization (ACSO) algorithm achieves improved efficiency. The first step in Advanced involves determining the number of cats to be used in the optimization process. Each cat functions as a potential solution in the problem-solving mechanism of the algorithm.

Every cat's position is defined in an M-dimensional space, where each dimension is assigned its own velocity and fitness value. These parameters help determine the cat's state, whether it should enter the seeking mode or the tracing mode, based on its evaluation of the fitness function. The process continues through iterative updates, gradually moving toward identifying the optimal position held by one of the cats.

ACSO operates as a continuous and single-objective optimization algorithm inspired by the contrasting behaviors of cats when they are resting and when they are actively pursuing a target. While typically stationary and alert, cats react swiftly when a target appears, shifting to an active pursuit. This concept is mirrored in the two modes of ACSO.

Initialization:

- Define the population size of cats, denoted by N. Each cat represents a candidate solution in the search space.
- Each cat is initialized with:
  - Position Vector:

    $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$,

    where i = 1, 2, ..., N, and M is the dimensionality of the problem (number of BBUs or RRHs).

  Velocity Vector:

  $V_i = (v_{i1}, v_{i2}, \ldots, v_{im})$

- Mode Flags are assigned for each cat to determine whether it operates in Seeking Mode (SM) or Tracking Mode (TM).
- Set parameters:
    - SMP (Seeking Memory Pool)
    - SRD (Seeking Range of the Selected Dimension)
    - CDC (Counts of Dimension to Change)
    - MR (Mixture Ratio) determines the proportion of cats in SM and TM.

Fitness Function:

The fitness function evaluates the load balancing efficiency across all BBUs. It considers both load distribution and resource utilization. One common fitness function for load balancing can be:

Fitness Function Equation:

$f(X_i) = 1 / (1 + (\alpha \times LB + \beta \times CBR))$

Where:

- LB: Load Balancing index, calculated as the standard deviation of the loads across all BBUs. $LB = \sqrt{((1 / M) \Sigma (L_j - \bar{L})^2)}$
    - L_j: Load on BBU *j*
    - $\bar{L}$: Average load across all BBUs
- CBR: Call Blocking Rate, defined as: CBR = (Number of Blocked Calls) / (Total Number of Call Requests)
- α, β: Weight coefficients to balance the importance of LB and CBR.

The goal is to maximize the fitness, which minimizes both LB and CBR.

Seeking Mode (SM):

In Seeking Mode, cats search locally around their current positions to find a better solution. Steps are:

Step 1: Generate SMP copies of the current position.

Step 2: For each copy:

- Randomly select CDC dimensions.
- Modify these dimensions by a random value within the range defined by SRD. $x_{ij}' = x_{ij} \times (1 + r \times SRD)$
    - *r* is a random number in the range [-1, 1].

Step 3: Evaluate the fitness of each candidate position.

Step 4: Select the best candidate position and update the cat's position accordingly.

Tracking Mode (TM):

In Tracking Mode, cats move toward the best global position found so far.

Velocity Update Equation

$$v_{ij}(t+1) = v_{ij}(t) + r_1 \times c \times (X\_best_j - x_{ij}(t))$$

Where:

- $v_{ij}(t)$: Current velocity of cat *i* in dimension *j*.
- r_1: Random number in the range [0, 1].
- c: Acceleration coefficient.
- X_best_j: Best position found in dimension *j*.

Position Update Equation $x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$

This guides the cat towards the global best solution, promoting global exploration.

Termination:

- Repeat the SM and TM updates for a predefined number of iterations or until convergence criteria are met.

- The best solution obtained from all cats is selected as the optimal load balancing configuration in the Cloud-RAN system.

4. RESULTS EVALUATION

This section presents the simulation outcomes of the proposed approach, which were generated and analyzed using MATLAB R2020a [17]. In the model, each BBU is responsible for managing three sectors, with the number of active users in each cell varying between 40 and 70. Additionally, the maximum capacity for each sector is set at 150 users. A traffic shaping mechanism is implemented to restrict traffic flow once it surpasses the predefined threshold. The evaluation focuses on a single hexagonal cell comprising 10 RRHs. The idle power consumption of the General Purpose Processor (GPP) is set at 120 Watts, while its peak power consumption is 215 Watts. Within the cell, a total of 75 users are considered for the simulation.

Blocking Probability:

Blocking probability is an important metric used to evaluate the efficiency of resource allocation in a network. It indicates the proportion of service requests that are denied access due to insufficient available resources. It is calculated by dividing the number of blocked service requests by the total number of service requests received. The formula is expressed as:

Blocking Probability = Sbs / Srs

Where:
Sbs = Number of blocked service requests Srs = Total number of service requests received

Response Time and Throughput:

Response time, measured in milliseconds (ms), refers to the time difference between when a user makes a service request and when the system responds to that request.

Throughput is the amount of data successfully transmitted over a network within a specific time period. It is measured in Megabits per second (Mbps) and is calculated using the following formula:

Throughput = Transmitted data (bytes) / Time taken for transmission

The Advanced Cat Swarm Optimization (ACSO) algorithm has been assessed to determine the global optimal cat position by adjusting cat velocity across different scenarios. The evaluation covers blocking probabilities between 0.05 and 0.5. Additionally, traffic load levels vary from 0.1 to 0.9, during which a noticeable decrease in blocking probability is observed, as presented in Table 1. The ACSO algorithm demonstrates improved response times when tested under traffic load intensities ranging from 0.1 to 0.95, detailed in Table 2. Furthermore, the throughput performance increases significantly, rising from 900 Mbps to 1700 Mbps, as the traffic load intensity extends from 0.1 up to 1.2, as shown in Table 3.

Table 1. Relationship between Blocking Probability and Traffic Load.

Blocking Probability	Traffic Load (PSO)	Traffic Load (CSO)	Traffic Load (ACSO)
0.05	0.22	0.15	0.10
0.10	0.53	0.39	0.20
0.15	0.57	0.46	0.25
0.20	0.61	0.47	0.30
0.25	0.66	0.53	0.42
0.30	0.71	0.55	0.45
0.35	0.78	0.72	0.67
0.40	0.85	0.77	0.70
0.45	0.89	0.80	0.76
0.50	1.10	1.00	0.90

Table 2. Traffic load corresponding to response time.

Response Time (s)	Traffic Load (PSO)	Traffic Load (CSO)	Traffic Load (ACSO)
0.098	0.45	0.38	0.10
0.099	0.54	0.41	0.15
0.100	0.66	0.53	0.23
0.101	0.72	0.60	0.29
0.102	0.77	0.68	0.36
0.103	0.83	0.75	0.41
0.104	0.89	0.81	0.59
0.105	0.95	0.90	0.63
0.106	1.03	0.99	0.78
0.107	1.10	1.05	0.81
0.108	1.15	1.08	0.95

Table 4. Traffic load corresponding to throughput.

Throughput (Mbps)	Traffic Load (PSO)	Traffic Load (CSO)	Traffic Load (ACSO)
900	0.05	0.08	0.10
1000	0.12	0.15	0.20
1100	0.23	0.39	0.40
1200	0.59	0.65	0.70
1300	0.64	0.73	0.80
1400	0.78	0.81	0.95
1500	0.81	0.94	0.97
1600	0.90	0.99	1.08
1700	1.00	1.10	1.20

The proposed approach implements the ACSO optimization algorithm, in contrast to the existing techniques that utilize CSO and PSO algorithms. A comparative evaluation of the proposed ACSO method with the existing methods is illustrated in Table 1, highlighting the relationship between blocking probability and traffic load.

Table 1 illustrates a comparison of blocking probability between the proposed approach and the existing techniques, namely CSO and PSO. The blocking probability in the experimental analysis ranges from 0.05 to 0.5. As the blocking probability increases, the corresponding traffic load also rises. At a blocking probability of 0.05, the traffic load for the proposed method is observed to be 0.1, whereas for the existing PSO and CSO methods, the traffic loads are 0.22 and 0.154, respectively. When the blocking probability is 0.15, the proposed method achieves a traffic load of 0.25, while the PSO and CSO methods reach traffic loads of 0.57 and 0.46. Additionally, when the blocking probability is 0.2, the proposed method demonstrates a traffic load of 0.3, compared to 0.47 and 0.61 for CSO and PSO, respectively. At a blocking probability of 0.3, the traffic loads are 0.45 for the proposed approach, and 0.55 and 0.71 for CSO and PSO, respectively. These results clearly show that the proposed method delivers superior performance when compared to the existing techniques. Furthermore, Table 2 presents a comparative evaluation of response time relative to traffic load for both the proposed and existing methods.

In addition, a comparative study of throughput versus traffic load for both the proposed and existing methods is illustrated in Table 3. This comparison highlights the performance differences in terms of throughput between the proposed approach and the conventional methods. The throughput is measured for the entire network and is calculated at the BBU pool. During the experimental analysis, throughput values range from 900 Mbps to 1700 Mbps. When the throughput is 900 Mbps, the corresponding traffic load values are 0.1 for the proposed method, 0.05 for PSO, and 0.08 for CSO. At a traffic load of 0.4 in the proposed method, the existing methods record traffic loads of 0.23 (PSO) and 0.39 (CSO), with the throughput measured at 1100 Mbps. For a throughput of 1300 Mbps, the traffic load is 0.8 for the proposed approach, while PSO and CSO yield 0.64 and 0.73, respectively. At a traffic load of 0.7 in the proposed method, PSO and CSO report higher traffic loads of 0.81 and 0.94, respectively, when the throughput reaches 1500 Mbps. Finally, at 1700 Mbps throughput, the proposed method achieves a traffic load of 1.2, whereas PSO and CSO reach 1 and 1.1, respectively. These results demonstrate that the proposed method provides significantly improved throughput compared to the existing techniques.

5. CONCLUSION

The proposed approach effectively addresses the issue of blocked calls and enhances load balancing by utilizing an Advanced Cat Swarm Optimization (ACSO) algorithm. This enhanced optimization technique focuses on maximizing Quality of Service (QoS) by reducing the call blocking probability. The current BBU-RRH (Baseband Unit–Remote Radio Head) architecture is first evaluated to assess QoS metrics, which are then leveraged in the optimization process to select optimal BBU-RRH configurations tailored to individual user requirements. The advanced CSO algorithm implemented in this study successfully minimizes call blocking while maximizing QoS. Experimental results demonstrate the superiority of the proposed method when compared to traditional Particle Swarm Optimization (PSO) and standard CSO algorithms. Specifically, the Advanced CSO (ACSO) reduces call blocking probability by 10% in comparison to PSO and CSO. Additionally, the throughput is improved by 8%, ranging from 900 Mbps to 1700 Mbps, outperforming existing approaches. The proposed algorithm also achieves a 7% reduction in response time under increasing traffic load conditions when compared to conventional methods. Future work will focus on further optimizing load balancing in BBU-RRH combinations by incorporating advanced cooperative techniques.

REFERENCES

[1] Dai, B., & Yu, W. (2014). Sparse beamforming and user-centric clustering for downlink cloud radio access network. *IEEE Access*, 2, 1326–1339. https://doi.org/10.1109/ACCESS.2014.2363830

[2] Hung, S.-C., Hsu, H., Lien, S.-Y., & Chen, K.-C. (2015). Architecture harmonization between cloud radio access networks and fog networks. *IEEE Access*, 3, 3019–3034. https://doi.org/10.1109/ACCESS.2015.2496272

[3] Khan, M., Alhumaima, R. S., & Al-Raweshidy, H. S. (2017). QoS-aware dynamic RRH allocation in a self-optimized cloud radio access network with RRH proximity constraint. *IEEE Transactions on Network and Service Management*, 14(3), 730–744. https://doi.org/10.1109/TNSM.2017.2729682

[4] Li, J., Peng, M., Cheng, A., Yu, Y., & Wang, C. (2014). Resource allocation optimization for delay-sensitive traffic in fronthaul constrained cloud radio access networks. *IEEE Systems Journal*, 11(4), 2267–2278. https://doi.org/10.1109/JSYST.2015.2414478

[5] Liu, L., Patil, P., & Yu, W. (2016). An uplink-downlink duality for cloud radio access network. In *2016 IEEE International Symposium on Information Theory (ISIT)* (pp. 1606–1610). IEEE. https://doi.org/10.1109/ISIT.2016.7541551

[6] Niu, B., Zhou, Y., Shah-Mansouri, H., & Wong, V. W. S. (2018). A dynamic resource sharing mechanism for cloud radio access networks. *IEEE Transactions on Wireless Communications*, 12(16), 8325–8338. https://doi.org/10.1109/TWC.2018.2849395

[7] Simeone, O., Maeder, A., Peng, M., Sahin, O., & Yu, W. (2016). Cloud radio access network: Virtualizing wireless access for dense heterogeneous systems. *Journal of Communications and Networks*, 18(2), 135–149. https://doi.org/10.1109/JCN.2016.000024

[8] Suresh, K., & Kumaratharan, N. (2020). Performance modelling of service function chaining in distributed controllers secure black SDN with NFV architecture. *Solid State Technology*, 63(6), 7558–7567.

[9] Suresh, K., & Kumaratharan, N. (2021). SDN controller allocation and assignment based on multicriterion

chaotic salp swarm algorithm. *Intelligent Automation & Soft Computing*, 27(1), 89–102. https://doi.org/10.32604/iasc.2021.015539

[10] Tang, J., Tay, W. P., & Quek, T. Q. S. (2015). Cross-layer resource allocation with elastic service scaling in cloud radio access network. *IEEE Transactions on Wireless Communications*, 14(9), 5068–5081. https://doi.org/10.1109/TWC.2015.2431683

[11] Tran, T. X., Kazemi, F., Karimi, E., & Pompili, D. (2017). Mobee: Mobility-aware energy-efficient coded caching in cloud radio access networks. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)* (pp. 461–465). IEEE. https://doi.org/10.1109/MASS.2017.86

[12] Tran, T. X., Le, D. V., Yue, G., & Pompili, D. (2018). Cooperative hierarchical caching and request scheduling in a cloud radio access network. *IEEE Transactions on Mobile Computing*, 17(11), 2729–2743. https://doi.org/10.1109/TMC.2018.2802907

[13] Ugur, Y., Awan, Z. H., & Sezgin, A. (2016). Cloud radio access networks with coded caching. In *2016 20th International ITG Workshop on Smart Antennas (WSA)* (pp. 1–5). IEEE. https://doi.org/10.1109/WSA.2016.7455335

[14] Wang, X., Wang, K., Wu, S., Di, S., Yang, K., & Jin, H. (2016). Dynamic resource scheduling in cloud radio access network with mobile cloud computing. In *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)* (pp. 1–6). IEEE. https://doi.org/10.1109/IWQoS.2016.7590378

[15] Wu, J., Zhang, Z., Hong, Y., & Wen, Y. (2015). Cloud radio access network (C-RAN): A primer. *IEEE Network*, 29(1), 35–41. https://doi.org/10.1109/MNET.2015.7053728

[16] Chen, J., Chen, H., & Wang, B. (2021). Intelligent resource allocation for 5G cloud radio access networks based on deep reinforcement learning. *IEEE Access*, 9, 13766–13776. https://doi.org/10.1109/ACCESS.2021.3052007

[17] Xu, W., Li, Y., & Han, Z. (2023). Joint resource allocation and user association for ultra-dense cloud radio access networks. *IEEE Transactions on Communications*, 71(2), 1096–1110. https://doi.org/10.1109/TCOMM.2022.3229264

[18] Zheng, K., Wu, Q., Zhang, T., & Long, K. (2022). Energy-efficient and delay-aware resource allocation for C-RANs with fronthaul constraints. *IEEE Transactions on Green Communications and Networking*, 6(3), 1374–1384. https://doi.org/10.1109/TGCN.2022.3154317

[19] Zhang, X., Shen, J., & Chen, Y. (2020). An efficient machine learning-based RRH selection scheme for C-RANs. *IEEE Access*, 8, 172525–172535. https://doi.org/10.1109/ACCESS.2020.3025193

[20] Al-Turjman, F., & Alturjman, S. (2021). 6G and e-health: Emerging technologies, opportunities, and challenges. *Future Generation Computer Systems*, 118, 315–323. https://doi.org/10.1016/j.future.2020.12.049