FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 3 | Pages 849-862 | e-ISSN: 2347-470X

Dynamic Optimization in 5G Network Slices: A Comparative Study of Whale Optimization, Particle Swarm Optimization, and Genetic Algorithm

Geoffrey Okindo[1*], Prof. George Kamucha[2] and Dr. Nicholas Oyie[3]
[1]Student, Pan African University Institute for Basic Sciences, Technology, and Innovation, Nairobi; eoffreymogendi7@gmail.com
[2]Prof, University of Nairobi, Nairobi; gkamucha@uonbi.ac.ke
[3]Dr, Murang'a University of Technology, Nairobi; noyie@mut.ac.ke

*Correspondence:** Geoffrey Okindo; *geoffreymogendi7@gmail.com*

ABSTRACT- This study presents a comprehensive framework for optimizing 5G network slices using metaheuristic algorithms, focusing on Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and massive Machine Type Communications (mMTC) scenarios. The initial setup involves a MATLAB-based 5G New Radio (NR) Physical Downlink Shared Channel (PDSCH) simulation and OpenAir-Interface (OAI) 5G network testbed, utilizing Ubuntu 22.04 Long Term Support (LTS), MicroStack, Open-Source MANO (OSM), and k3OS to create a versatile testing environment. Key network parameters are identified for optimization, including power control settings, signal-to-noise ratio targets, and resource block allocation, to address the unique requirements of different 5G use cases. Metaheuristic algorithms, specifically the Whale Optimization Algorithm (WOA), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA), are employed to optimize these parameters. The algorithms are assessed based on their ability to enhance throughput, reduce latency, and minimize jitter within the network slices and the MATLAB simulation model. Each algorithm's performance is evaluated through iterative testing, with improvements measured against established pre-optimization benchmarks. The results demonstrate significant enhancements in network performance post-optimization. For eMBB, the GA shows the most substantial increase in throughput, while PSO is most effective in reducing latency for URLLC applications. In mMTC scenarios, GA achieves the most notable reduction in jitter, illustrating the potential of metaheuristic algorithms in fine-tuning 5G networks to meet diverse service requirements. The study concludes that the strategic application of these algorithms can significantly improve the efficiency and reliability of 5G network slices, offering a scalable approach to managing the complex dynamics of next-generation wireless networks.

Keywords: WOA, 5G, PSO, Genetic Algorithm, MATLAB, eMBB, URLLC, mMTC.

ARTICLE INFORMATION

Author(s): Geoffrey Okindo, George Kamucha and Dr. Nicholas Oyie;

Received: 26/04/2024; **Accepted**: 01/07/2024; **Published**: 30/07/2024;
e-ISSN: 2347-470X;
Paper Id: IJEER 2604-32;
Citation: 10.37391/ijeer.120316
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-12/ijeer-120306.html

crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

The emergence of 5G technology marks a significant transformation in telecommunications, featuring increased speeds, lower latency, and greater connectivity across devices. The key to exploiting these advancements is network slicing, which allows a single physical network to be divided into multiple virtual networks, each optimized for specific service requirements. However, the optimization of these network slices presents unique challenges, particularly in dynamic environments where traditional network management strategies falter.

In environments demanding high data rates as in Enhanced Mobile Broadband (eMBB), immediate and reliable communications as in Ultra-Reliable Low-Latency Communications (URLLC), and support for numerous low-power devices as in massive Machine Type Communications (mMTC), conventional network management frameworks often struggle [1]. These frameworks typically lack the flexibility required for dynamic resource allocation and fail to effectively balance the competing demands of diverse 5G applications [2]. This inadequacy is marked by suboptimal performance in throughput, latency, and jitter, particularly under varying bandwidth and latency conditions [3].

To address these challenges, this study proposes a novel optimization framework using metaheuristic algorithms, known for their robustness in navigating complex problem spaces through iterative refinement. The Whale Optimization Algorithm (WOA), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA) are evaluated for their effectiveness in a dual setting: a 5G network testbed and a MATLAB-based simulation environment. Our research focuses on the critical network parameters influencing eMBB, URLLC, and mMTC scenarios, aiming to demonstrate substantial improvements in network performance metrics [4][5][6].

1.1 Related Works

In the study by Dalila Boughaci [7], a comprehensive examination of optimization in 5G networks through metaheuristic algorithms is presented, addressing key issues such as resource management, energy efficiency, and network planning. This work emphasizes the complexity of optimizing 5G infrastructures and the efficacy of evolutionary algorithms and local search techniques in finding optimal network configurations. Our research builds upon Boughaci's foundational analysis, offering a detailed exploration of specific optimization challenges within 5G networks.

Rayner et al. [8] delve into meta-heuristic strategies for refining 5G network slicing, particularly emphasizing the Virtual Network Embedding (VNE) challenge, pivotal for the effective management of 5G's multifaceted demands. Their analysis sifts through existing studies to highlight typical network configurations for VNE evaluation, prevalent metaheuristic techniques, and critical Quality of Service (QoS) metrics in network slicing contexts. The study notably emphasizes the significance of Ant Colony Optimization (ACO), Genetic Algorithms (GA), and Particle Swarm Optimization (PSO) in addressing VNE complexities, stressing the necessity for adaptable and efficient algorithm choices in 5G optimization.

In the study by Hassana et al. [9], the focus is on optimizing the deployment of 5G Base Stations at millimetre wave frequencies using meta-heuristic algorithms, notably Particle Swarm Optimization (PSO), to reduce the number of base stations and enhance their placement for optimal user data rate provision in 5G networks. The research utilizes PSO for strategic Base Station (BS) placement at frequencies like 28GHz and 38 GHz, coupled with an iterative approach to remove surplus BSs, thereby enhancing network efficiency with fewer base stations and quicker computations for desired network QoS.

Rabia et al. [10] investigated optimizing energy efficiency and throughput in 5G heterogeneous networks simulated on MATLAB, focusing on device-to-device communications and separate downlink/uplink association strategies. Their work, using genetic algorithms and particle swarm optimization, revealed that decoupled access schemes significantly enhance network performance.

Our research extends the foundational analyses of Boughaci [7] by widening the application of metaheuristic algorithms in 5G networks, tackling advanced resource allocation and antenna positioning. Building upon Rayner et al. [8], we incorporate cutting-edge meta-heuristic methods, including the Whale Optimization Algorithm (WOA), to optimize essential network aspects like power control and resource block distribution more deeply. We also enhance the strategies introduced by Hassana et al. [9] by applying our optimization methods to network slicing, aiming for dynamic resource allocation across diverse network segments such as eMBB, URLLC, and mMTC. Furthermore, our methodology advances the work of Rabia et al. [10] by integrating these approaches to reduce latency and jitter, thereby improving throughput and energy efficiency across various service scenarios. This comprehensive approach significantly advances adaptive 5G network optimization, offering a framework that adapts to the dynamic and complex nature of modern 5G systems.

2. MATERIALS AND METHODS

2.1 Slicing Simulation Setup

The simulation setup for 5G network slicing involved leveraging widely used platforms and technologies to create a dynamic testing environment. Ubuntu 22.04 Long Term Support (LTS) was chosen for its stability and support, alongside Micro Stack for OpenStack virtualization, and Open-Source MANO (OSM) version 14 for managing network services. For virtual infrastructure management, k3OS, a lightweight Kubernetes distribution, was selected due to its suitability for edge and IoT applications, facilitating the orchestration of containerized network functions [11]. This configuration enabled the deployment of network slices incorporating the OAI CORE, gNodeB (gNB), and User Equipment (UE), followed by comprehensive performance testing. The process, illustrated in Figure 1, involved setting up Kubernetes-based Network Functions (KNF) and Network Service Descriptors (NSD), defining the network's functional characteristics [12].

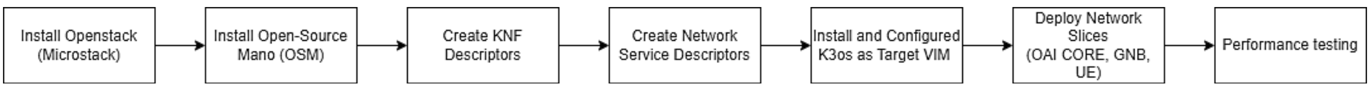

Figure 1. Test Bed Implementation

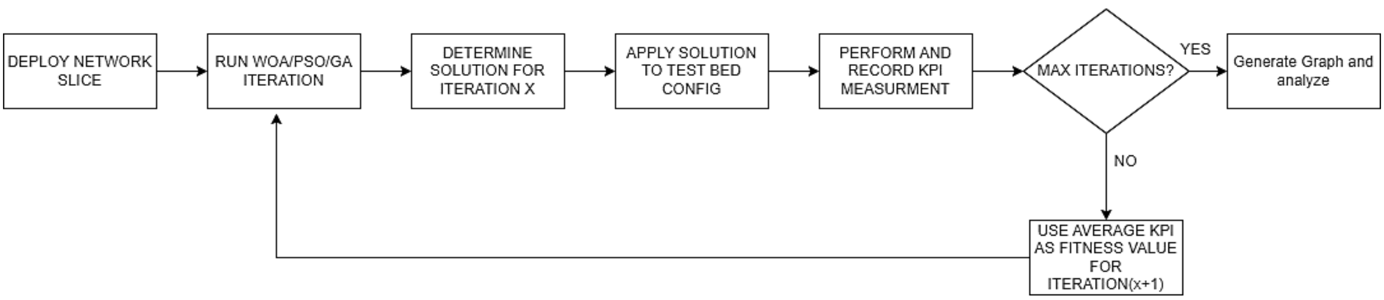

Figure 2. Metaheuristic Algorithm Application on OAI 5G testbed

FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 3 | Pages 849-862 | e-ISSN: 2347-470X

For the application of metaheuristic optimization algorithms (Whale Optimization Algorithm, Particle Swarm Optimization, and Genetic Algorithm), an iterative approach was employed to enhance network performance. As depicted in *figure 2*, this involved deploying a network slice, applying algorithm-generated solutions, and evaluating performance based on key indicators. This iterative process allowed for a systematic comparison of different strategies within the controlled 5G testbed environment.

Parameter optimization focused on key radio resource management and signal transmission aspects, such as power control, signal-to-noise ratio targets, and receiver sensitivity. These parameters were fine-tuned using the metaheuristic algorithms to achieve optimal throughput, latency, and jitter, essential for diverse 5G applications requiring real-time communication and reliable connectivity. The iterative testing and evaluation process facilitated a balanced optimization, considering network efficiency and operational stability.

2.1.1. Testbed Deployment

The testbed was deployed according to the architecture illustrated in *figure 3* and following the steps outlined below:

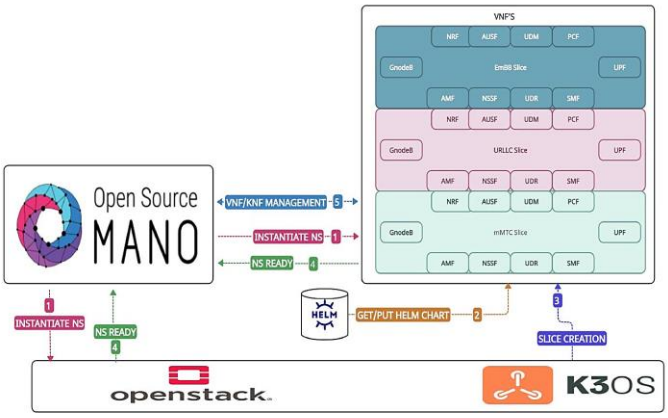

Figure 3. Logical architecture

MicroStack Installation: We initialized our localized OpenStack environment with MicroStack on an Ubuntu 22.04 LTS platform, ensuring system compatibility with requisite hardware and software specifications. The installation process involved a sequence of stages, including Snap installation, host system preparation, cloud bootstrapping, and subsequent configuration tailored to our simulation needs.

Open-Source MANO Installation*:* OSM, vital for our Network Function Virtualization (NFV) orchestration, was deployed within the MicroStack environment. This setup involved executing a shell script from the OSM repository, which automated the deployment process, and configuring environment variables for OSM client interaction.

k3OS Installation*:* We opted for k3OS, a lightweight Kubernetes distribution, to establish an efficient platform for our simulated 5G network slices. The installation process entailed downloading the k3OS ISO image, creating a Virtual Machine (VM) within MicroStack, and configuring network settings to ensure seamless communication within the simulation environment.

Network Functions and Service Descriptors*:* Our approach included the creation of Virtual Network Function Descriptors (VNFDs) and Network Service Descriptors (NSDs) for configuring and orchestrating the 5G network components. These descriptors defined the structure, deployment, and operational behavior of the network functions, facilitating the instantiation of network slices tailored to specific service requirements.

Slice Creation and Deployment*:* We employed OSM to instantiate network slices by executing a series of CLI commands, each corresponding to different components of the 5G network architecture. The deployment process encompassed the core network, gNB, and UE slices, essential for establishing a functional 5G network simulation [13].

Performance Monitoring: To ensure Service Level Agreement (SLA) compliance and optimal resource utilization, we integrated Prometheus and Grafana for monitoring the performance of Kubernetes-based Network Functions (KNFs) within our simulation environment. This setup provided real-time insights into various performance metrics, crucial for the effective management and orchestration of network slices[14].

2.1.2 Metaheuristic Algorithm Integration

The integration process begins with the development of Python scripts that implement the metaheuristic algorithms. These scripts are designed to interact with the Kubernetes API, enabling the dynamic modification of network slice configurations based on the optimization results of the algorithms.

Each algorithm brings a unique approach to tuning key parameters like PO_NominalWithGrant and SSPBCH_BlockPower detailed in Table I, aiming to maximize efficiency in different use cases like enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC), and massive Machine Type Communications (mMTC).

Table 1. Network Tuning Parameters

Parameter	Description
P0_NOMINALWITHGRANT	Nominal power with grant (in dBm)
SSPBCH_BLOCKPOWER	Power of the broadcast channel block
PUSCH_TARGETSNRX10	Target SNR for PUSCH
PUCCH_TARGETSNRX10	Target SNR for PUCCH
MAX_RXGAIN	Maximum receiver gain
PRACH_DTX_THRESHOLD	PRACH Detection Threshold
NROF_UPLINK_SYMBOLS	Number of uplink symbols
RSRP_THRESHOLD_SSB	RSRP Threshold for SSB
SESSION_AMBR_UL0	Session AMBR for uplink (in Mbps)
SESSION_AMBR_DL0	Session AMBR for downlink (in Mbps)

Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 3 | Pages 849-862 | e-ISSN: 2347-470X

2.1.3 KPI Measurement

Key Performance Indicators (KPIs) such as throughput, latency, and jitter are measured using test traffic generated within the Kubernetes cluster. Network Function Pods included performance measurement tools ping and iperf which were used to collect the metric data. The Python scripts orchestrate the execution of performance tests and collect KPI metrics, which are then fed back into the metaheuristic algorithms to guide the optimization process.

2.1.4 Dynamic Resource Optimization

The dynamic resource optimization process involves the continuous monitoring of network performance and the adaptive adjustment of network parameters. The metaheuristic algorithms, integrated with the Python scripts, respond to changes in network conditions by recalculating optimal parameter values. The Kubernetes Application Program Interface (API) facilitates the application of these updated configurations to the network slices in real time, ensuring optimal performance under varying network loads and conditions.

The integration of metaheuristic algorithms with Kubernetes for the dynamic optimization of 5G network slices represents a sophisticated approach to network management. By leveraging the computational power of WOA, PSO, and GA, alongside the flexibility and scalability of Kubernetes, this framework provides a powerful tool for achieving superior network performance tailored to the specific requirements of eMBB, URLLC, and mMTC services. This innovative integration paves the way for the development of intelligent, self-optimizing 5G networks.

2.2 MATLAB Simulation Setup

The flowchart in *figure 4* represents an overview of the MATLAB 5G Toolbox simulation for the New Radio Physical Downlink Shared Channel (NR PDSCH). The MATLAB 5G Toolbox simulation for the NR PDSCH encompasses the end-to-end data transmission process in a 5G network, from DLSCH encoding and PDSCH mapping to signal reception and decoding [15]. This model is vital for evaluating 5G NR link performance, particularly PDSCH throughput and latency, according to 3GPP standards [16].

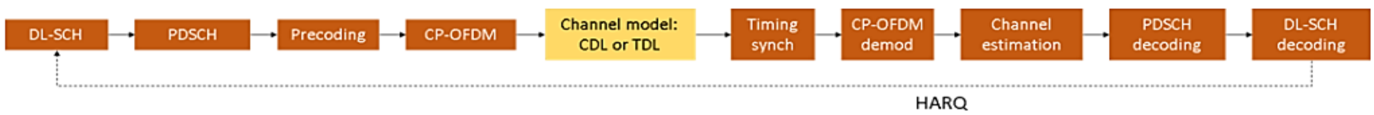

Figure 4. MATLAB 5G NR PDSCH Model [20]

2.2.1. Simulation Workflow

The procedure starts with initializing key parameters, followed by signal generation, propagation, reception, and processing. This systematic approach includes channel estimation, signal decoding, and throughput calculation, with the option for parallel execution to enhance efficiency.

2.2.2. Metaheuristic Algorithm Application

We integrate the WOA, PSO, and GA to optimize the 5G NR PDSCH link described above. As depicted in *figure 5.* these algorithms are applied in an iterative process aimed at enhancing KPIs, specifically throughput and latency, by finetuning the 5G link model parameters.

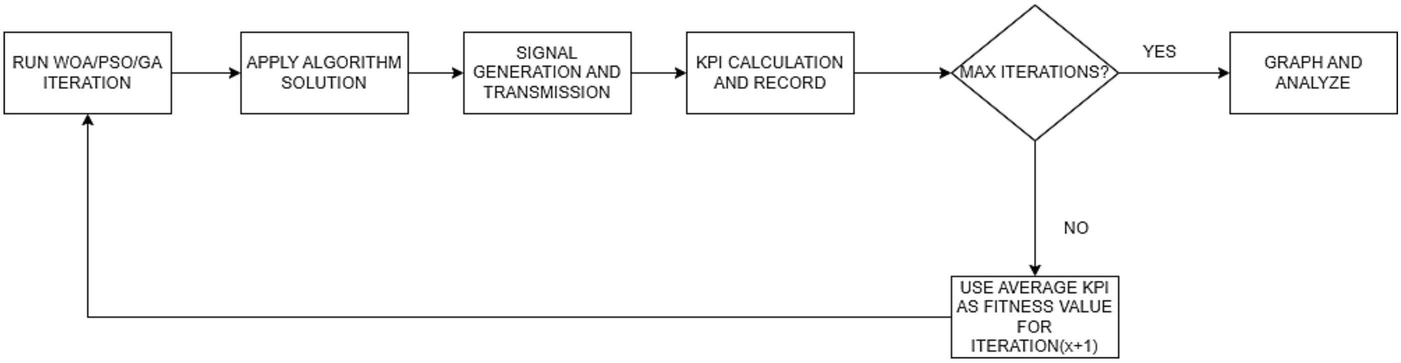

Figure 5. Metaheuristic Algorithm Application

The optimization process initiates with each algorithm generating candidate solutions, which represent potential sets of parameters for the 5G NR PDSCH link. These solutions are implemented in the simulation environment to assess their impact on the 5G link's performance. Subsequently, the system evaluates the performance by calculating the KPIs, which serve as fitness values guiding the algorithms toward optimal parameter sets.

This iterative cycle continues until a predefined number of iterations is reached, ensuring a comprehensive exploration of the solution space. The effectiveness of the parameter adjustments is then analysed, highlighting the most advantageous configurations identified through the metaheuristic optimization process.

For the metric optimization component, we focus on key parameters critical to 5G NR performance:

- *NSizeGrid*: Influences bandwidth and data rate by determining the count of Physical Resource Blocks (PRBs) [17].

- *Subcarrier Spacing:* Affects the signal's resilience to delay spread and Doppler spread by adjusting frequency domain granularity[18].

- *Maximum Doppler Shift:* Addresses the channel's time selectivity, crucial for maintaining stable connections in high-mobility scenarios [19].

These parameters are iteratively adjusted within defined ranges by the algorithms, leveraging mathematical models inspired by natural phenomena and genetics to identify the configurations that maximize throughput and minimize latency, thereby addressing the demands of eMBB and URLLC.

3. THEORY AND FORMULATIONS

3.1 Whale Optimization Algorithm (WOA) Integration

WOA simulates humpback whales' social behavior, particularly their bubble-net feeding strategy, to identify optimal network configurations[4]. The algorithm's iterative process involves the following steps:

1. *Initialize* the whale population.

2. *Calculate* the fitness of each whale.

3. *Update* the position of each whale using either the encircling prey mechanism, the bubble-net attacking method, or the search for prey, based on probabilistic rules.

4. *Repeat steps 2 and 3 until a termination* criterion is met (e.g., maximum number of iterations or a satisfactory fitness level).

5. *Encircling Prey:* Potential solutions are encircled, akin to how whales encircle fish. This is represented by adjusting network parameters towards the best current solution:

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \quad (1)$$

where $\vec{X}^*(t)$ is the position of the best solution at time t, $\vec{A}$ and $\vec{D}$ are coefficient vectors guiding the search.

6. *Bubble-Net Attacking Method:* This involves spirally decreasing the area around the prey, simulated by:

$$\vec{X}(t+1) = \vec{D}' \cdot e^{b \cdot l} \cdot cos\,(2\pi l) + \vec{X}^*(t) \quad (2)$$

where $\vec{D}'$ is the distance to the best solution, b is a constant defining the spiral shape, and l is a random number in $[-1,1]$.

3.2 Particle Swarm Optimization (PSO) Integration

PSO mimics birds flocking, with each particle representing a potential network configuration [20]. The algorithm's iterative process involves the following steps:

1. Initialize a swarm of particles with random positions and velocities within the search space.

2. Evaluate the fitness of each particle.

3. Update each particle's personal best position if the current position is better.

4. Update the global best position if any particle's position is better than the current global best.

5. Update the velocity and position of each particle according to the equations mentioned above.

6. Repeat steps 2-5 until a termination criterion is met (*e.g.*, maximum number of iterations, or a satisfactory fitness level is achieved).

7. The update of particle velocity and position is governed by:

Velocity Update:

$$\vec{v}(t+1) = w \cdot \vec{v}(t) + c_1 \cdot \vec{r}_1 \cdot (\vec{p}_{\text{best}} - \vec{x}) + c_2 \cdot \vec{r}_2 \cdot (\vec{g}_{\text{best}} - \vec{x}) \quad (3)$$

where w is the inertia weight, c_1 and c_2 are learning factors, $\vec{r}_1$ and $\vec{r}_2$ are random vectors, $\vec{p}_{\text{best}}$ is the particle's best-known position, and $\vec{g}_{\text{best}}$ is the global best position.

Position Update:

$$\vec{x}(t+1) = \vec{x}(t) + \vec{v}(t+1) \quad (4)$$

- $\vec{x}(t+1)$: Represents the position of a particle in the swarm at the next time step $(t+1)$. The position of a particle corresponds to a potential solution in the search space of the optimization problem.
- $\vec{x}(t)$: Denotes the current position of the particle at time t. It indicates where the particle is in the search space now.
- $\vec{v}(t+1)$: This is the velocity of the particle at the next time step $(t+1)$. The velocity dictates how far and in what direction the particle will move in the search space.

The update rule for velocity, as provided, combines several components, including the particle's inertia, cognitive component (attraction to its personal best position), and social component (attraction to its global best position).

3.3 Genetic Algorithm (GA) Integration

GA uses processes analogous to natural selection and genetics to evolve solutions [6]. Key steps include:

1. *Initialization:* Generate an initial population of N chromosomes randomly.

2. *Evaluation:* Evaluate the fitness f(C) of each chromosome C in the population.

3. *New Population:* Create a new population by repeating the following steps until the new population is complete:

4. *Selection:* The best-performing configurations are selected to form a mating pool.

5. *Crossover:* Pairs of configurations exchange segments of their encoding to produce new configurations.

6. *Mutation*: Random changes are introduced to new configurations to explore a wider solution space.

7. *Replace*: Use the newly generated population for a further run of the algorithm.

8. *Termination*: If the end condition is satisfied, stop, and return the best solution in the current population.

9. *Loop*: Go to step 2.

By employing these algorithms, the framework ensures continuous optimization of network configurations, adapting to evolving network conditions and demands. This approach demonstrates the potential of algorithmic optimization in dynamic network slice management, enhancing both network performance and reliability.

3.4 MATLAB Model

3.4.1. Simulation Essentials

- *SNR Loop:* Essential for evaluating signal strength performance across various noise levels.

- *Channel Estimation:* Both ideal and practical estimations are considered, leveraging the PDSCH Demodulation Reference Signal (DM-RS).

- *Configurations:* Critical parameters such as bandwidth, subcarrier spacing, and modulation are defined.

3.4.2. Core Mathematical Models [23]

- CP-OFDM Modulation is fundamental, representing the baseband signal $s(t)$ as:

$$s(t) = \sum_{k=0}^{N-1} X[k] e^{\frac{j2\pi kt}{T}} \quad (5)$$

where $X[k]$ are data symbols, N the number of subcarriers, and T the symbol duration.

- MIMO Processing is captured by:

$$y = Hx + n \quad (6)$$

where x is the transmitted signal vector, H the channel matrix, y the received signal vector, and n the noise.

- Channel Response $h(t, \tau)$ reflects multipath effects:

$$h(t, \tau) = \sum_{l=0}^{L-1} g_l(t) \delta(\tau - \tau_l) \quad (7)$$

with $g_l(t)$ as the complex gain and τ_l the delay for each path.

- Precoding involves Singular Value Decomposition (SVD) of H :

$$\mathrm{H} = \mathrm{U\Sigma V^H} \quad (8)$$

$$\mathrm{W} = \mathrm{V} \quad (9)$$

where U, Σ, and V are the matrices from the SVD of H and W is the precoding matrix chosen from the SVD of the channel matrix.

- Channel Coding: LDPC or Turbo codes are applied, and modulation varies from QPSK to 256QAM based on conditions.

- Channel Estimation utilizes least squares (LS) or minimum mean square error (MMSE) techniques:

$$\hat{H} = Y \cdot (X^H (X \cdot X^H)^{-1}) \quad (10)$$

where Y is the received signal and X is the known transmitted reference signal.

- Equalization employs MMSE:

$$\hat{s} = \left(\mathrm{H}^H \mathrm{H} + \frac{\sigma^2}{E_s} \mathrm{I}\right)^{-1} \mathrm{H}^H \mathrm{r} \quad (11)$$

where H is the channel matrix, r the received signal, σ^2 the noise variance, and E_s the symbol energy.

- HARQ Process efficiency is defined by:

$$\eta = \frac{D}{D + H} \quad (12)$$

where D is the successfully received data and H is the overhead due to retransmissions.

- Throughput and SNR: The Signal-to-Noise Ratio (SNR) is defined per resource element (RE), considering signal and noise across all antennas. Throughput is then calculated based on the proportion of successfully decoded bits over the total transmitted bits.

- Throughput calculation for each SNR point:

$$\text{Throughput}_i = \frac{D_{\text{succ},i}}{D_{\text{total},i}} \quad (13)$$

Where: $D_{\text{total},i}$ is the total number of bits transmitted at the i-th SNR point, considering the number of frames, slots per frame, Resource Blocks, symbols per slot (S), resource elements per Resource Block, modulation order, and transmission layers.

$D_{\text{succ},i}$ is the number of successfully decoded bits at the i-th SNR point, derived from the simulation results.

- Average throughput over all SNR points:

$$\text{Average Throughput} = \frac{1}{N_{\text{SNR}}} \sum_{i=1}^{N_{\text{SNR}}} \text{Throughput}_i \quad (14)$$

Throughput $_i$ is the throughput at the i-th SNR point, calculated as the ratio of successfully decoded data to the total transmitted data. Average Throughput is the average of throughputs calculated across all SNR points.

3.4.3 Latency calculation in the model

Latency is a key performance metric in the 5G NR PDSCH model, representing the signal delay from source to destination. It is influenced by the OFDM symbol duration, the channel delay spread, and processing time. OFDM Symbol Duration: The duration of an OFDM symbol, T_{symbol}, including the cyclic prefix, is influenced by the subcarrier spacing and the cyclic prefix length [24]. It is calculated as:

$$T_{\text{symbol}} = \frac{1}{\text{Subcarrier Spacing}} + T_{\text{CP}} \quad (15)$$

where T_{CP} is the cyclic prefix duration, added to each OFDM symbol to mitigate inter-symbol interference. Wider subcarrier spacing results in shorter symbol durations, but the cyclic prefix extends each symbol's duration, impacting latency.

Channel Delay Spread Impact: The channel delay spread, significant in multipath environments, affects latency by varying the arrival times of signal components [25]. The maximum channel delay, maxChDelay, represents the longest multipath component delay.

$$\text{maxChDelay} = \lceil \max(\text{PathDelays}) \times \text{SampleRate} \rceil + \text{ChannelFilterDelay} \quad (16)$$

A higher value of maxChDelay indicates greater latency, which is particularly critical in scenarios requiring timely signal transmission.

Where:

- PathDelays represents an array or list of delays for each path the signal can take in a multipath environment. Each element in the PathDelays array corresponds to the time delay experienced by the signal on that path.
- SampleRate is the rate at which the signal is sampled.
- ChannelFilterDelay refers to the delay introduced by any filtering processes applied to the signal in the channel.

Processing Time Measurement: The processing time, processingTime, measures the duration for executing simulation steps like generating OFDM symbols and decoding:

$$\text{processingTime} = \text{toc}(\text{processingStartTime}) \quad (17)$$

Where:

- *toc* is a function used in MATLAB to measure the elapsed time. It typically records the time now it is called and calculates the difference from a previously set start time, which is set by a corresponding 'tic' function.
- Processing Start Time: This variable is the timestamp recorded at the beginning of the process or operation whose duration is being measured. It is typically set by a tic function or equivalent, marking the start of the processing or simulation activities.

This metric reflects the computational efforts in the simulation, distinct from the physical transmission time in an actual network but crucial for assessing algorithmic efficiency.

Latency Calculation: The total latency, L_{total}, combines the OFDM symbol duration, channel delay spread, and processing time:

$$L_{\text{total}} = N \times T_{\text{symbol}} + \text{maxChDelay} + \text{processingTime} \quad (18)$$

where N is the number of OFDM symbols transmitted, representing the overall latency in the model.

4. RESULTS AND DISCUSSION

4.1.OAI 5G testbed Slice Optimization Results

Pre-optimization Performance Benchmarking: We established baseline performance metrics for latency, throughput, and jitter within our 5G network slicing testbed to evaluate the impact of metaheuristic optimizations. These initial benchmarks, crucial for quality of service (QoS) assessment in 5G networks, were obtained under standard operational conditions without optimization algorithm interventions.

Throughput was gauged using the iPerf tool between the UE and UPF, simulating eMBB scenarios requiring high data rates. Latency measurements were taken via ICMP echo requests between the UE and UPF, reflecting URLLC requirements for minimal round-trip times. For Jitter, the assessment was carried out using iPerf to analyse packet inter-arrival times, which is crucial for understanding the tolerance of mMTC scenarios to variations in packet timing, as these applications are generally less sensitive to delay variations.

The collected metrics serve as a reference for evaluating the enhancements brought by the WOA, PSO, and GA. *Table 2* summarizes these pre-optimization performance metrics, providing a snapshot of network performance before optimization.

Table 2. Reference Performance Metrics

Metric	Average
Latency (ms)	0.82
Throughput (Mbits/sec)	18.75
Jitter (ms)	0.1575

These benchmarks offer a clear comparative baseline for assessing the optimization algorithms' impact on network

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 3 | Pages 849-862 | e-ISSN: 2347-470X

performance, highlighting the potential enhancements in QoS enabled by each algorithmic approach.

4.1.1. Metaheuristic Optimization for eMBB

The graph in *figure 6* shows the performance of three metaheuristic algorithms WOA, PSO, and GA over thirty iterations in optimizing the throughput of a 5G network slice. These algorithms aim to find the optimal set of network parameters that maximize throughput, a key performance indicator for Enhanced Mobile Broadband (eMBB) services. The core findings are summarized in *table 3*:

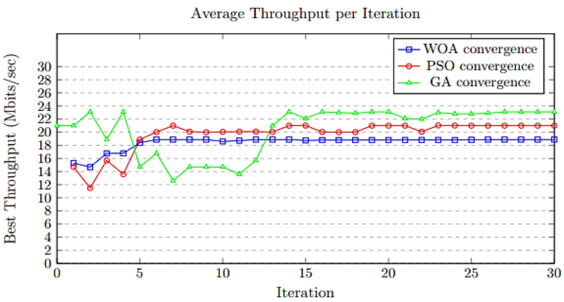

Figure 6. Metaheuristic Applications for Throughput Optimization

Table 3. Summary of Metaheuristic Optimization Results for eMBB

Algorithm	Early Iteration Performance	Stability After Convergence	Average Throughput (Mbits/sec)
WOA	Rapid improvement, peaks by the 5th iteration	Stable, minimal fluctuations	~ 18.88
PSO	Variable early performance, stabilizes after 6th iteration	Consistent performance from 10th iteration stability	~ 23.10
GA	High throughput from the start, stable	High Stability and Consistency	~ 23.10

Key Insights:

- GA demonstrated superior performance in throughput optimization, achieving the highest final throughput value of approximately 23.1 Mbits/sec. Its stable convergence behavior and the effective balance between exploration and exploitation were notable.
- PSO exhibited significant improvement after initial exploration, with adaptive mechanisms leading to a final throughput of about 21Mbits/sec. Its performance was marked by good stability and the ability to find near-optimal solutions.
- WOA, despite showing rapid initial improvements, appeared to converge prematurely to suboptimal solutions, resulting in a lower final throughput of approximately 18.88 Mbits/sec. However, its consistent performance post-convergence could be advantageous in scenarios requiring stability over maximum throughput.
- *Throughput Improvement for eMBB:* The application of the GA exhibited the most notable improvement in throughput, which is of particular importance for eMBB services. The GA's final iteration achieved an average throughput of 23.1 Mbits/sec, compared to the pre-optimization average of 18.75 Mbits/sec. This represents an increase of approximately 23.2%, enhancing the network's capability to handle high data rate services. PSO also showed an improvement, stabilizing at 21 Mbits/sec, indicating a 12% increase. WOA, while showing the least improvement, still enhanced the throughput to an average of 18.88 Mbits/sec, a slight increment from the baseline.

These results show the effectiveness of GA in the context of 5G network slicing for eMBB, with PSO also showing promising outcomes. The choice of algorithm may depend on specific network requirements, including the need for rapid convergence or high stability in throughput performance.

4.1.2. Metaheuristic Optimization for URLLC

The graph in *figure 7* illustrates the performance of three metaheuristic algorithms- WOA, PSO, and GA in terms of their ability to reduce Round Trip Time (RTT), a critical metric for URLLC in a 5G network slicing environment. The graph spans thirty iterations, and the objective is to minimize the average RTT, with lower values indicating better performance. The core findings are summarized in *table 4*.

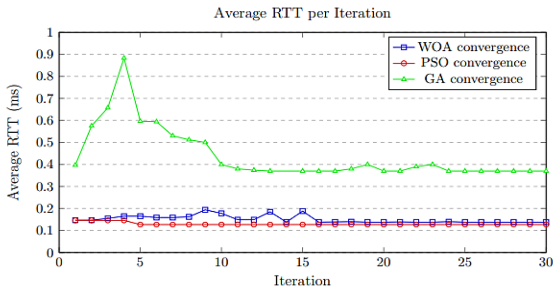

Figure 7. Metaheuristic Applications for Latency Optimization

Table 4: Summary of Metaheuristic Optimization Results for URLLC

Algorithm	Early Iteration Performance	Stability After Convergence	RTT (ms)
WOA	Fluctuating, starting at 0.147, peaking at 0.194	Highly stable, maintains optimal RTT	~ 0.138
PSO	Rapid convergence, starting at ~ 0.127	Highly stable, maintains optimal RTT	~ 0.127
GA	Divergent, starting from 0.397 to 0.883	Improves to ˜0.37, less stable than WOA/PSO	~ 0.37

Key Insights:

- PSO exhibited outstanding performance, rapidly achieving, and consistently maintaining the lowest RTT of about 0.127 ms, indicating its efficiency in quickly identifying and sustaining optimal solutions, making it particularly suited for URLLC applications in 5G networks.

- WOA demonstrated initial exploration with fluctuating RTT values but eventually converged to a stable and low RTT solution of about 0.138 ms, reflecting its capability to balance exploration and exploitation effectively.

- GA showed a considerable range in RTT values during initial iterations, suggesting significant exploration. Despite improvements, it stabilized at a higher RTT than WOA and PSO of about 0.37 ms, indicating it might be less optimal for minimizing latency in URLLC scenarios.

- *Latency Reduction for URLLC:* The URLLC use case, where latency is a critical metric, benefited significantly from PSO optimization. PSO converged to an average RTT of 0.127 ms , a substantial reduction from the pre-optimization average of 0.82 ms, indicating an 84.5% decrease in latency. WOA and GA also reduced latency to 0.138 ms and 0.37 ms , respectively, but PSO's optimization results were the most impactful, potentially transforming the network's suitability for real-time applications.

For the specific goal of RTT reduction in a URLLC 5G network slicing context, PSO demonstrates superior performance, rapidly converging to the lowest latency values and maintaining them throughout the iterations. WOA, while not as effective as PSO, provides a consistent and stable solution after an initial period of exploration. GA appears to be less suited for latency minimization in this scenario, as indicated by its higher RTT throughout the optimization process. The overall analysis suggests that PSO is the most suitable algorithm for latency-critical applications in 5G networks, given its quick convergence and consistent minimization of RTT.

4.1.3. Metaheuristic Optimization for mMTC

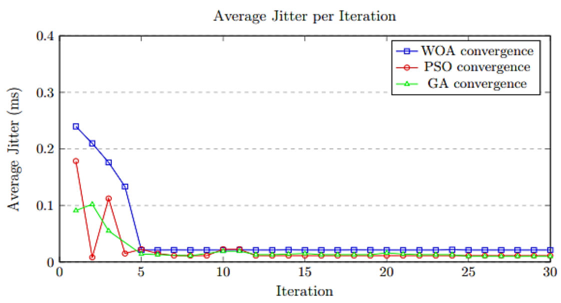

Figure 8: Metaheuristic Applications for Jitter Optimization

The graph in *figure 8* provides a visual representation of the optimization performance of three metaheuristic algorithms WOA, PSO, and GA over thirty iterations to reduce jitter in a 5G network slicing context. Jitter, the variability in packet transmission times, is a critical performance metric, particularly in scenarios involving massive Machine Type Communications (mMTC), where consistent timing is vital. The core findings are summarised in *table 5*.

Table 5. Summary of Metaheuristic Optimization Results for mMTC

Algorithm	Early Iteration Performance	Stability After Convergence	Jitter (ms)
WOA	Starts at 0.2397, quickly reduces	Stabilizes at 0.0212 by 5th iteration	~ 0.0212
PSO	Starts at 0.1785, impressive reduc-tion to 0.00825 by 2nd iteration	Exhibits fluctua-tions but main-tains low jitter	~ 0.00825 at best, ~ 0.01123 average
GA	Gradual decline from 0.091, significant reduction to 0.014 by 5th iteration	Stable, minimal fluctuations, maintains low jitter	~ 0.010

Key Insights:

- GA demonstrated superior performance in reducing and maintaining low jitter levels, suggesting its evolutionary mechanisms are well-suited for mMTC jitter optimization, achieving the lowest average jitter of ~ .010 ms by the 30th iteration.

- WOA showed rapid initial improvement, quickly stabilizing at a low jitter level of 0.0212 ms, indicating its efficacy in rapidly identifying and exploiting effective solutions for jitter reduction.

- PSO displayed the ability to achieve the lowest jitter value (~ 0.00825 ms) at its peak performance, despite experiencing some variability in jitter levels, highlighting its dynamic optimization capabilities.

- Jitter Minimization for mMTC: In the context of mMTC, jitter minimization is essential for maintaining communication reliability among a vast number of IoT devices. GA outperformed other algorithms by converging to an average jitter of 0.010 ms, which is a dramatic decrease from the preoptimization average of 0.1575 ms. This represents a reduction of approximately 93.7% , suggesting that the GA is exceptionally well-suited for optimizing network configurations in mMTC scenarios. PSO also achieved a low jitter value of 0.01123 ms , contributing to the overall reduction in delay variation.

The GA emerges as the most effective algorithm for jitter reduction, showing both a strong ability to decrease jitter values and maintain low levels throughout the optimization process. WOA and PSO also demonstrate commendable performance, with WOA showing early and stable convergence and PSO achieving the lowest jitter values at certain points, albeit with less stability. The graph indicates that the choice of algorithm for jitter optimization in 5G network slicing may depend on the specific network characteristics and the balance required between rapid convergence and long-term stability.

Table 6. Post-Optimization Performance Metrics with KPI Improvement

Metric	WOA	PSO	GA
Average Throughput (Mbps)	18.88	21	23.1
Throughput Improvement (%)	+0.69	+12	+23.2
Average Latency (ms)	0.138	0.127	0.37
Latency Reduction (%)	-83.17	-84.5	-54.9
Average Jitter (ms)	0.0212	0.01123	0.01
Jitter Reduction (%)	-86.5	-92.8	-93.7

4.1.4. Post-Optimization Performance Metrics: The data presented in *table 4* indicates a substantial enhancement in network performance, attributable to the sophisticated optimization algorithms applied. These improvements are especially pronounced when compared to the pre-optimization benchmarks, highlighting the efficacy of the metaheuristic techniques in fine-tuning the network's operational parameters. The successful optimization of these KPIs emphasizes the potential of metaheuristic algorithms in addressing the complex and dynamic challenges inherent in 5G network slicing. By leveraging WOA, PSO, and GA, the testbed has achieved a level of performance that is demonstrably superior to the initial baseline, thereby reinforcing the merit of these algorithms in real-world 5G deployment scenarios.

4.2. MATLAB 5G Model Optimization Results

4.2.1. Pre-optimization Performance Benchmarking

To assess the effectiveness of metaheuristic optimization algorithms in enhancing the performance of 5G NR systems, it is essential to establish a comprehensive preoptimization performance benchmark. This benchmark serves as a baseline to compare the post-optimization results, allowing for a quantifiable evaluation of the improvements gained through the optimization process.

The initial performance metrics are collected under a standardized set of conditions within the 5G NR simulation environment. The simulation is configured to reflect a typical 5G NR setup, adhering to the parameters outlined in the 3GPP Release 15 specifications. The key performance indicators (KPIs) of interest are average throughput and processing time.

4.2.2. Throughput Measurement: Throughput, measured in megabits per second (Mbps), is captured under various signal-to-noise ratio (SNR) conditions to ensure a comprehensive understanding of the system's data handling capabilities. The average throughput is computed over a series of transmission frames to mitigate any anomalies or transient behaviours within the simulation.

4.2.3. Latency Measurement: Latency, reported in seconds, encapsulates the total transmission cycle, including data encoding, propagation through the channel, processing at the receiver, and decoding. The measurement of processing time, from the start of data encoding to the completion of decoding, provides insight into the system's efficiency and responsiveness.

4.2.4. Pre-Optimization Performance Metrics: *Table 7* presents the baseline performance levels for the system's throughput and processing latency. These figures are instrumental in gauging the impact of the metaheuristic optimization algorithms subsequently applied to the 5G NR simulation setup.

Table 7. Pre-Optimization Performance Metrics

Metric	**Average**
Average Throughput (Mbps)	0.82
Processing Time (seconds)	18.75

This benchmarking approach lays the groundwork for an assessment of the optimization techniques applied to the 5G NR testbed, ensuring that any performance gains can be accurately attributed to the optimization interventions.

4.2.5. Metaheuristic Optimization for eMBB

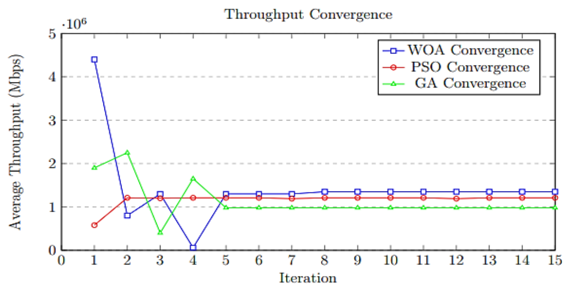

Figure 9: Metaheuristic Applications for Throughput Optimization

The convergence graph in *figure 9* illustrates the optimization performance of three metaheuristic algorithms WOA, PSO, and GA over fifteen iterations, to maximize the average throughput in a 5G NR PDSCH simulation environment. The graph is a representation of how each algorithm progresses towards finding a parameter configuration that yields the highest throughput, measured in megabits per second (Mbps). The core findings are summarized in *table 8.*

Table 8. Summary of Throughput Optimization Performance

Algorithm	Early Iteration Performance	Stability After Convergence	Average Throughput
WOA	Initially high variation, stabilizes $\sim 1.35 \times 10^6$ from 8^{th} iteration	High variation initially, stable after convergence	$\sim 1.35 \times 10^6$
PSO	Rapid increase, quick convergence to $\sim 1.21 \times 10^6$ by 2^{nd} iteration	High stability, minor fluctuations	$\sim 1.21 \times 10^6$
GA	Fluctuations till 4^{th} iteration, steady afterwards	Consistent post-5^{th} iteration, but lower throughput jitter	$\sim 0.98 \times 10^6$

Key Insights:

- WOA demonstrated a strong initial performance with the highest starting throughput, but it exhibited significant fluctuations before stabilizing at a high throughput of about $\sim 1.35 \times 10^6$Mbps, suggesting an aggressive exploration of solutions before converging to a local optimum.

- PSO showed a quick convergence to an optimal throughput level of about $\sim 1.21 \times 10^6$Mbps, maintaining high stability with only minor deviations, reflecting its effective balance between exploration and exploitation, making it well-suited for scenarios requiring rapid optimization.

- GA started with a moderately high throughput but did not reach the levels of WOA or PSO after convergence.

- *Throughput Enhancement:* Post-optimization, a significant improvement in throughput was observed across all algorithms. The WOA demonstrated a substantial increase, with average throughput stabilizing at approximately $\sim 1.35 \times 10^6$ Mbps, a marked improvement from the pre-optimization value of 966,912 Mbps. The PSO also showed an enhanced performance, with a consistent throughput of around $\sim 1.21 \times 10^6$ Mbps post-optimization. The GA, while showing a more moderate improvement, still managed to increase the throughput to $\sim 0.98 \times 10^6$ Mbps.

achieving about $\sim 0.98 \times 10^6$Mbps, indicating the need for further parameter tuning or a more extensive exploration phase to enhance its performance.

WOA might be preferable when the initial solution space is large and diverse, while PSO could be more suitable in scenarios where quick convergence to a high-quality solution is desired. GA may be beneficial where a more extensive search of the solution space is necessary, possibly over a larger number of iterations.

For throughput optimization in a 5G NR PDSCH simulation context, WOA and PSO demonstrate strong performance, with PSO showing slightly lower throughput but higher stability after convergence. GA's performance suggests that it may require further parameter tuning or a greater number of iterations to compete with WOA and PSO in this specific context.

4.2.6. Metaheuristic Optimization for URLLC

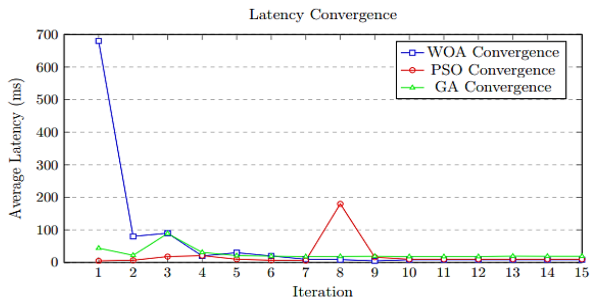

Figure 10: Metaheuristic Applications for Latency Optimization

The graph in *figure 10* depicts the performance of three metaheuristic algorithms WOA, PSO, and GA over fifteen iterations, to minimize latency in a 5G NR PDSCH simulation setup. The graph is a measure of the algorithms' success in finding parameter configurations that yield the lowest average latency, measured in milliseconds (ms). The core findings are summarised in *table 9*.

Table 9. Summary of Metaheuristic Optimization Results for URLLC

Algorithm	Early Iteration Performance	Stability After Convergence	Average Latency
WOA	High initial latency at 680 ms, dropping to 8.5 ms	Highly stable post-initial improvement, maintaining ~ 8.5ms	~ 8.5
PSO	Starts at a low 5 ms, fluctuates but stabilizes at 10 ms	Shows variability, including a spike, but maintains a low latency 10ms	~ 10
GA	Begins at moderate 44 ms , gradually decreases to ~ 19 ms	Demonstrates stable convergence with minor fluctuations 19ms	~ 19

Key Insights:

- WOA exhibited a significant recovery capability, rapidly reducing latency from a high initial value to stabilize at a low latency of approximately 8.5ms , showcasing its potential for scenarios demanding quick latency reduction.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 3 | Pages 849-862 | e-ISSN: 2347-470X

- PSO presented an impressive start at 5ms and, despite fluctuations, managed to achieve a consistently low latency of around 10ms by the end, indicating its dynamic optimization effectiveness.
- GA, with a more-steady approach, displayed gradual improvements but settled at a higher latency of about 19 ms, suggesting a need for further optimization or exploration to enhance its latency reduction capabilities.
- *Latency Reduction:* In terms of latency, the optimization algorithms effectively reduced the total transmission cycle time. The WOA achieved a post-optimization latency convergence to around 8.5ms, a significant decrease from the preoptimization processing time of 24.9051 seconds. The PSO algorithm, despite some fluctuations, successfully reduced latency to a steady 10 ms in the latter iterations. The GA displayed a consistent decline in latency, ultimately converging to approximately 19 ms, indicating a substantial reduction from the pre-optimization benchmark.

WOA might be preferred when initial solutions are not close to the optimal and a rapid convergence is required. PSO could be chosen for scenarios where an initial good performance is likely, and the risk of occasional performance drops is acceptable. GA may be beneficial where a consistent and gradual improvement is preferred over a set number of iterations.

For latency optimization in a 5G NR PDSCH simulation context, WOA demonstrates a remarkable ability to recover from initially high latency and converge to a low-latency solution. PSO offers a strong start and quick convergence but may require careful monitoring to manage fluctuations. GA provides consistent performance, although not reaching the low latency levels of WOA or PSO, and may benefit from extended exploration or different parameter settings.

Post-Optimization Performance Metrics: *Table* 10 shows the performance levels post the optimization process, showcasing the substantial gains achieved through the application of the metaheuristic algorithms. The WOA emerged as the leading algorithm, offering the highest throughput and the lowest latency. PSO also displayed robust performance, particularly in reducing latency. GA, while not outperforming WOA or PSO, still demonstrated noticeable improvements in both throughput and latency.

Table 10. Post-Optimization Performance Metrics with KPI Improvement

Metric	WOA	PSO	GA
Average Throughput (Mbps)	1,350,000	1,210,000	~ $0.98\text{x}10^6$
Throughput Improvement (%)	+39.66	+25.14	+1.35
Average Latency (ms)	8.5	10	19
Latency Reduction (%)	-65.87	-59.86	-23.76

The comparative analysis clearly illustrates that the application of metaheuristic optimization algorithms has successfully enhanced the performance of the 5G NR system. The improvements in throughput and latency are not only statistically significant but also likely to translate into tangible benefits in a real-world deployment, such as higher data rates and more responsive network services. These results validate the potential of utilizing advanced optimization techniques in the ongoing development and refinement of 5G networks.

4.3. Algorithm-Specific Performance Characteristics

The performance differences observed among the Whale Optimization Algorithm (WOA), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA) can be attributed to their intrinsic operational mechanisms, which impact their efficiency in optimizing network parameters under varying network conditions and requirements.

4.3.1 Whale Optimization Algorithm (WOA)

WOA simulates the hunting strategy of humpback whales, which involves encircling and spirally approaching prey. This method is mirrored in the algorithm's search pattern, where potential solutions spiral towards the best solution, enabling rapid aggregation of good solutions. This behavior explains WOA's rapid initial improvements in throughput optimization for eMBB. In the dynamic 5G environment, WOA's strategy can be effective for quickly identifying promising areas of the search space. However, this same characteristic may lead to premature convergence, as seen in the suboptimal final throughput compared to GA and PSO. The algorithm's strong initial convergence was effective in the mMTC scenario, where rapid stabilization at low jitter levels (0.0212 ms) was crucial.

4.3.2. Particle Swarm Optimization (PSO)

PSO uses a combination of individual knowledge (cognitive component) and social sharing of information (social component). Each particle adjusts its path not only based on its own best-known position but also influenced by the swarm's overall best findings, allowing for dynamic adjustments based on shared learnings. This dual-input system provided the robustness needed in URLLC scenarios, where PSO achieved the lowest and most stable RTT (0.127 ms). The adaptability of PSO also makes it well-suited for URLLC where quick adaptation to new conditions is paramount. The continual adjustment to both individual and collective learning helps maintain minimal RTT, optimizing for the critical latency requirements of URLLC applications.

4.3.3. Genetic Algorithm (GA)

GA operates through mechanisms analogous to natural selection, including selection, crossover, and mutation. This evolutionary approach allows for a diverse exploration of the solution space, which can evolve over generations to adapt to complex optimization landscapes. GA's methodical search process makes it particularly effective in mMTC scenarios where a broad exploration of network configurations is necessary to optimize for a vast array of device types and communication patterns. GA achieved the lowest average jitter (0.010 ms), demonstrating its capability to refine and stabilize network performance over time through generational data processing and robust exploration-exploitation balance.

4.4. Practical Implications for 5G Deployment

- *Network Customization:* The distinct behaviours of these algorithms suggest a strategic approach to selecting optimization methods based on specific network needs. For instance, networks that require rapid setup and immediate responsiveness might Favor PSO or WOA, while those that deal with a highly heterogeneous environment with less stringent time constraints might benefit from the exhaustive search capabilities of GA.

- *Algorithmic Efficiency:* Understanding each algorithm's strengths and limitations helps in tuning them more effectively for specific network roles. For example, enhancing WOA's exploration capabilities or refining GA's mutation rates could tailor these algorithms more precisely for their deployment environments.

- *Hybrid Optimization Models:* Given the strengths and weaknesses observed, a hybrid model employing PSO for initial rapid convergence followed by GA for sustained long-term optimization could be particularly effective, leveraging the rapid discovery of PSO and the deep exploratory capabilities of GA. The dynamic nature of 5G network demands, especially in dealing with varying traffic loads, device mobility, and diverse service requirements (e.g., eMBB vs. URLLC vs. mMTC), suggests that a flexible optimization approach, possibly even a hybrid model combining strengths of different algorithms, could be necessary.

5. CONCLUSION

This study presented a novel approach to optimizing resource allocation within 5G network slices, catering to the demands of diverse 5G scenarios such as eMBB, URLLC, and mMTC. By employing metaheuristic optimization algorithms like the WOA, PSO, and GA, we effectively tuned network parameters, resulting in significant performance enhancements. The utilization of a 5G network testbed and a MATLAB 5G NR PDSCH simulation model facilitated a thorough performance analysis, leading to notable improvements in throughput and latency. Despite facing challenges related to real-world network complexities and the need for algorithm parameter tuning, the applied metaheuristic algorithms demonstrated a promising capacity for rapid convergence to optimal solutions and performance stability. The application of GA achieved a throughput increase of approximately 23.2% and a Jitter reduction of up to 93.7%, with PSO showing latency reductions of up to 85.4% emphasizing the potential of these algorithms to significantly boost network efficiency and reliability. This research marks a substantial advancement in 5G network optimization, showcasing the potential of metaheuristic algorithms to surmount significant telecommunications challenges and heralding a new era of more efficient and adaptable 5G infrastructures.

6. FUTURE WORK

The exploration of metaheuristic optimization algorithms within 5G network contexts has unveiled significant potential for performance enhancement across key metrics. Future research should broaden the scope of algorithm application, delve into hybrid algorithm strategies, and tackle the challenge of real-world deployment to ensure scalability and adaptability to diverse network conditions. Moreover, integrating 5G with burgeoning technologies like edge computing and Artificial Intelligence (AI) could unlock new optimization avenues, enhancing the interplay between 5G and these technologies. Additionally, addressing energy efficiency within 5G optimizations will align with global sustainability goals, making 5G networks not only more effective but also more environmentally friendly.

7. ACKNOWLEDGMENTS

This research was made possible through funding provided by the African Union Commission through the Pan African University Institute of Basic Sciences, Technology and Innovation (PAUISTI), which has played a crucial role in the successful completion of this work. The views and opinions expressed herein are those of the Author(s) and do not necessarily reflect the views of the African Union Commission.

Supplementary Materials*:* The Python and MATLAB scripts and raw data outputs are available online at: **https://github.com/mogfrey/5GMETAHEURISTIC.git**

REFERENCES

[1] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," Ieee Access, vol. 6, pp. 55765–55779, 2018.

[2] L. A. Freitas et al., "Slicing and Allocation of Transformable Resources for the Deployment of Multiple Virtualized Infrastructure Managers (VIMs)," in 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), 2018, pp. 424–432. doi: 10.1109/NETSOFT.2018.8459990.

[3] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," IEEE Communications Magazine, vol. 55, no. 8, pp. 138–145, 2017.

[4] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in engineering software, vol. 95, pp. 51–67, 2016.

[5] H. Qu, B. Zhang, Z. Duan, and Y. Zhang, "Network Slice Resource Mapping Method Based on Discrete Binary Particle Swarm Optimization Algorithm," in 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS), 2020, pp. 412–416.

[6] X. Qi, S. Khattak, A. Zaib, and I. Khan, "Energy efficient resource allocation for 5G heterogeneous networks using genetic algorithm," IEEE Access, vol. 9, pp. 160510–160520, 2021.

[7] D. Boughaci, "Solving optimization problems in the fifth generation of cellular networks by using meta-heuristics approaches," Procedia Comput Sci, vol. 182, pp. 56–62, 2021.

[8] R. Gomes, D. Vieira, and M. F. de Castro, "Application of Meta-Heuristics in 5G Network Slicing: A Systematic Review of the Literature," Sensors, vol. 22, no. 18, p. 6724, 2022.

[9] H. Ganame, L. Yingzhuang, H. Ghazzai, and D. Kamissoko, "5G base station deployment perspectives in millimeter wave frequencies using meta-heuristic algorithms," Electronics (Basel), vol. 8, no. 11, p. 1318, 2019.

[10] R. Arshad, M. Farooq-i-Azam, R. Muzzammel, A. Ghani, and C. H. See, "Energy efficiency and throughput optimization in 5g heterogeneous networks," Electronics (Basel), vol. 12, no. 9, p. 2031, 2023.

[11] M. Turhan, G. Scopelliti, C. Baumann, E. Truyen, J. T. Muehlberg, and M. Petik, "The Trust Model for Multi-tenant 5G Telecom Systems Running Virtualized Multi-Component Services," 2021.

[12] R. Botez, A.-G. Pasca, and V. Dobrota, "Kubernetes-Based Network Functions Orchestration for 5G Core Networks with Open-Source MANO," in 2022 International Symposium on Electronics and Telecommunications (ISETC), 2022, pp. 1–4.

[13] C. Novaes, C. Nahum, I. Trindade, D. Cederholm, G. Patra, and A. Klautau, "Virtualized c-ran orchestration with docker, Kubernetes, and openairinterface," arXiv preprint arXiv:2001.08992, 2020.

[14] S. Barrachina-Muñoz, M. Payaró, and J. Mangues-Bafalluy, "Cloud-native 5G experimental platform with over-the-air transmissions and end-to-end monitoring," in 2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), 2022, pp. 692–697.

[15] J. Sancho, "Design and Testing of a Transmitter-Channel-Receiver Model Using Matlab 5G Toolset," 2019.

[16] M. Polese, M. Giordani, and M. Zorzi, "3GPP NR: the standard for 5G cellular networks," 5G Italy White eBook: from Research to Market, 2018.

[17] A. Spelman and others, "System modeling for 5G integrated digital transmitters," 2021.

[18] F. Schaich, T. Wild, and R. Ahmed, "Subcarrier spacing-how to make use of this degree of freedom," in 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring), 2016, pp. 1–6.

[19] X. Lin, Z. Lin, S. E. Löwenmark, J. Rune, R. Karlsson, and others, "Doppler shift estimation in 5G new radio non-terrestrial networks," in 2021 IEEE Global Communications Conference (GLOBECOM), 2021, pp. 1–6.

[20] T. Alam, S. Qamar, A. Dixit, and M. Benaida, "Genetic algorithm: Reviews, implementations, and applications," arXiv preprint arXiv:2007.12673, 2020.