IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 August 2022, accepted 16 August 2022, date of publication 18 August 2022, date of current version 26 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3200035

RESEARCH ARTICLE

A Hybrid Latency- and Power-Aware Approach for Beyond Fifth-Generation Internet-of-Things Edge Systems

AJAY KAUSHIK[1] AND HAMED S. AL-RAWESHIDY[2], (Senior Member, IEEE)
[1]Department of Computer Science and Engineering, SRM University, Sonepat, Haryana 131029, India
[2]Department of Electronic and Electrical Engineering, Brunel University London, Uxbridge, London UB8 3PH, U.K.

Corresponding author: Hamed S. Al-Raweshidy (hamed.al-raweshidy@brunel.ac.uk)

ABSTRACT Fifth-generation (5G) empowered internet of things (IoT) edge networks suffer from latency in delay-sensitive applications. To fulfil the low latency requirements of beyond fifth-generation (B5G)-IoT applications and provide quality of service (QoS) to IoT-edge communication, it is important to minimize server delay. Furthermore, 5G-IoT systems consume more power than their predecessors, which is a concern given the growing size of future IoT networks. This research presents a hybrid latency and power-aware approach for B5G-IoT networks (HLPA B5G-IoT) that minimizes latency with minimum overhead on battery-constrained IoT nodes while simultaneously providing a power-efficient solution for B5G-IoT-edge networks. HLPA B5G-IoT has a novel algorithm classifier tool (ACT) for selecting appropriate optimization algorithms based on the characteristics and requirements of B5G-IoT systems. The ACT matrix not only parametrically compares HLPA B5G-IoT with existing approaches but also identifies crucial parameters that enable algorithm selection for load balancing and energy efficiency. In this paper, metaheuristic algorithms, i.e., biogeography-based optimization (BBO) and grey wolf optimization (GWO), are tailored to meet the requirements of load balancing and power efficiency in IoT-edge systems. The proposed load-balancing algorithm reduces latency and improves overall network performance by 33.33%, 27.45%, 23.52%, 21.56%, 13.72%, 11.76%, and 7.84% compared with simulated annealing (SA), genetic algorithm (GA), particle swarm optimization (PSO), bacteria foraging algorithm (BFA), ant colony optimization (ACO), bat algorithm (BA), and genetic SA PSO (GSP), respectively. The power-efficiency algorithm consumes 46.6%, 40%, 32.2%, 27.7%, 15.5%, 11.1%, and 6.6% less energy compared with SA, GA, PSO, BFA, ACO, BA, and GSP, respectively.

INDEX TERMS Beyond fifth-generation, edge computing, Internet of Things, latency, load balancing, power consumption, workload allocation.

I. INTRODUCTION

With the arrival of beyond fifth-generation (B5G) technology, the internet of things (IoT) is expected to generate a large volume of data through diverse applications, e.g., smart everything, augmented reality, and self-driving cars. These applications are complex and time sensitive; they require real-time data processing to produce optimal results [1], [2]. B5G networks are expected to have extremely low latency, live response, high processing capabilities, and low power consumption. Currently, there is a need for technological advancement toward B5G networks due to a mismatch between the requirements of large IoT-enabled applications and fifth-generation (5G) networks [1], [3], [4].

Cloud computing has become increasingly popular in recent years as a method of improving computing capabilities, but, for time-sensitive IoT applications, it has a number of problems: high energy consumption and poor quality of service (QoS) provisioning [5], [6]. Data transmission to a remote cloud can take a long time, as cloud-computing

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Shu.

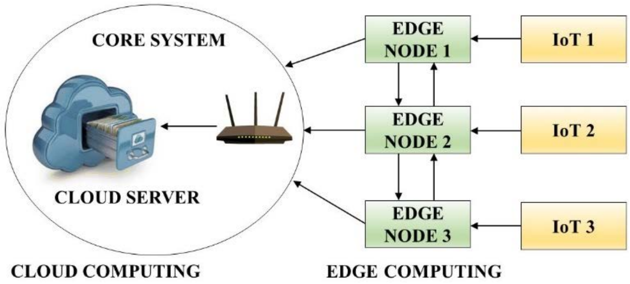


FIGURE 1. An IoT-edge system.

servers are often located in remote data sources. The problem is exacerbated by the increasing number of mobile devices and network sizes, which results in a high network load and thus unacceptable network delays. Furthermore, sending large amounts of data to the cloud necessitates increased bandwidth and consumes a significant amount of network energy. Edge computing is an extension of cloud computing for QoS provisioning, which, in IoT-edge systems, uses terminal edge devices with storage and computing capabilities to offload computation requests to nearby edge servers and thereby reduces network latency [7]. Edge servers have limited computation resources compared with cloud servers, and, therefore, they cannot handle a large number of computation requests. Furthermore, the network energy consumption of edge computing may surpass that of cloud computing for highly loaded systems and multiple requests. As a result, minimizing latency and power consumption in IoT-edge systems is critical to ensure QoS for the end user. The architecture of an IoT-edge system is depicted in Fig. 1, where independent IoT networks interact with the cloud server through respective edge nodes (ENs). In this way, an IoT network can fetch relevant information efficiently and quickly from the cloud using the corresponding EN.

A. MOTIVATION AND CONTRIBUTION

This subsection explains the motivation and contribution of the proposed approach in a profound manner. Workload allocation, load balancing, and energy efficiency are challenging tasks for B5G-IoT networks due to the heterogeneous and dynamic traffic characteristics of IoT-edge servers. A B5G-IoT network is made up of large number of tiny IoT nodes and must offload data from nodes and IoT devices evenly among ENs to optimize network bandwidth, network latency, and network processing time. This is important because inefficient task-offloading schemes increase latency and power consumption in the IoT-edge network [8]. Furthermore, existing task-offloading algorithms seldom prioritize power-consumption reduction in IoT-edge networks, which results in poor network performance [9], [10], [11], [12]. The paper contributions are as follows:

- Load balancing and minimizing power require distinct features and parameter settings, and, therefore, they cannot be optimized using a single approach. Some evolutionary algorithms are suitable for load management in B5G-IoT systems, whereas some swarm-based approaches are suited for minimizing power dissipation. As a contribution, this research presents a hybrid technique to fulfil the aforementioned performance parameters. An algorithm classifier tool (ACT) is developed to facilitate the selection of appropriate metaheuristic algorithms based on their characteristics and the requirements of B5G-IoT systems. The coordination of IoT system requirements and algorithm features is essential for optimal performance.
- Minimizing latency and power consumption in an IoT-edge network is a major challenge due to the power constraints of IoT nodes. Many existing workload-allocation algorithms use comparison methods [13], Markov decision processes [14], and greedy search methods, which are computationally complex and thus cause battery and memory drainage. This research presents two novel parameters to optimize the load distribution and power dissipation of B5G-IoT systems, i.e., active load index (ALI) and effective power coefficient (EPC), respectively. ALI not only minimizes latency but also improves the edge-response time. To further manage the load distribution, this article modifies the traditional mutation step of biogeography-based optimization (BBO). The dynamic mutation of BBO is a simple and effective approach for large B5G-IoT networks.
- A majority of existing methods primarily focus on task offloading, with power consumption being a secondary consideration [9], [10], [11], [12]. Excessive power consumption in IoT-edge systems may result in inactivity and disconnection. This research proposes EPC, which optimizes the data-transmission process between ENs and end users to minimize power consumption in an IoT-edge system.
- B5G-IoT networks require light, efficient, and scalable task-allocation techniques due to their increasing size and limited resources. This paper presents a hybrid latency- and power-aware approach for B5G-IoT networks (HLPA B5G-IoT) using metaheuristic algorithms, i.e., BBO [15] and grey wolf optimization (GWO) [16]. BBO and GWO have better convergence rate than prior metaheuristic algorithms, e.g., genetic algorithm (GA), ant colony optimization (ACO), and particle swarm optimization (PSO). HLPA B5G-IoT is lightweight, and simple, making it ideal for resource-constrained IoT nodes. A hybrid metaheuristic framework that uses different algorithms based on diverse system requirements has not been implemented in the literature to improve the performance of B5G-IoT-edge systems.

This paper is structured as follows: Sections II and III explain a literature review and problem formulation, respectively; system, delay, and energy models are discussed in Section IV; the proposed algorithm is explained in Section V; results and conclusion are presented in Sections VI and VII, respectively.

II. RELATED WORK

Edge computing is a trending research area in IoT. Many offloading and delay-minimizing algorithms have recently been proposed for mobile edge computing (MEC) [17] and fog computing [18]. Reference [19] proposed a mobile offloading strategy for improving wireless-transmission access efficiency in MEC systems. Reference [20] used the Markov model to develop a delay-minimizing method for MEC systems. To minimize overall service delay, [21] proposed a delay-reduction strategy for fog computing systems. Reference [22] focused on the trade-off between reliability and task latency in offloading services to MEC systems. Reference [23] focused on improving network performance, which was measured using network utility, network workload, and service delay. Reference [24] focused on resource allocation to improve the performance of real-time fog computing, the goal of which was to identify a balance between a high task-completion rate and a high throughput. Reference [25] presented a cache-aware edge-offloading scheme for IoT-edge cloud systems. Their work added a sharable cache to the edge server by allowing data to be shared across multiple computing processes to reduce job delays. Reference [26] proposed a dynamic microservice mechanism to reduce edge-server delay and complexity.

Metaheuristic approaches have been employed to handle other technical optimization problems. References [27], [28], [29], [30], [31] used a combination of bio-inspired approaches to solve the reactive power-source planning problem. Reference [32] presented a hybrid metaheuristic approach to reduce the cost of microgrid systems. References [33], [34] proposed a modified version of the original whale optimization approach and the Harris hawk optimizer to solve complex optimization problems. Reference [35] used neural networks for dynamic clustering in IoT. Reference [36] presented multiagent system clustering for efficient resource assignment in massive IoT. Reference [35] and [36] employed backpropagation neural networks and convolutional neural networks, respectively, for IoT performance optimization. Reference [37] used distributed artificial intelligence for active resource allocation in IoT, the results of which suggest that high-performance resource management is achieved by merging cognitive radio with wireless sensor networks (WSNs). References [38] and [39] used simulated annealing (SA) to improve the performance of MEC and photovoltaic systems, respectively. Reference [40] used GA to present a data replica-placement approach that can be used for scientific purposes. However, GA has a poor local search capability, resulting in poor performance at later stages. Reference [41] presented a cloud-based workflow-scheduling approach to reduce the application cost using PSO. Reference [42] proposed a hybrid bacteria foraging algorithm (BFA) for optimal task scheduling, the objective of which was to reduce network makespan and energy consumption. ACO was proposed by [43] for task offloading in fog computing; their research model focused on fog-node service rate as a performance metric for IoT networks. Reference [44] proposed a network model that caters for inadequate computing capabilities and limited battery power of user equipment; they employed Dinkelbach's algorithm and a bat algorithm (BA) for performance optimization. Reference [45] improved the performance of smart mobile devices using genetic SA PSO (GSP). The authors focused on task-execution time and data-transmission time. Reference [46] provided resource allocation and task offloading services in an IoT system using a multi-unmanned aerial vehicle (UAV) approach. They employed UAVs as aerial base stations for the edge network. Reference [47] presented a smart packet transmission strategy for reward clipping that assures high reliability and excellent packet delivery. To handle intelligent transmission scheduling in cognitive IoT systems, they employed a combined approach of generative adversarial network and deep distribution Q network. Reference [48] presented a policy gradient-based actor-critic learning approach for minimizing power, resolving the resource block and providing a solution for ultra-reliable and low-latency communication scheduling by optimizing the policy gradient for optimal rate allocation. Mixed-integer linear programming was used by [49] to reduce latency in IoT-edge applications; the authors also minimized the dimensioning cost of fog nodes. Reference [50] presented a service composition method using a device–edge–cloud combination. In their research, task allocation was based on service-request priority and branch-preference estimation. Reference [51] proposed a three-layer offloading scheme. Their architecture minimized the quantity of data transferred to the cloud and reduced latency by processing data locally on the device layer.

Many researchers throughout the world have recently focused on 5G and B5G-IoT communication. A network slicing strategy for 5G enabled IoT services was presented in [52]. Their approach used dynamic 5G slice allocation to allow the execution of concurrent IoT applications. Reference [53] presented a detailed survey of physical layer security frameworks in 5G IoT. The authors examined various security attacks and threats, as well as associated challenges. Reference [54] covered the new features introduced in releases 16 and 17 of 5G standard. The authors discussed both present and upcoming 5G features including fast data rate and reliability. Reference [55] merged edge computing with multi-tier integrated blockchain to ensure enhanced security for B5G-IoT networks. Reference [56] focused on ensuring QoS for B5G-IoT network and presented an approach to specify node-specific QoS requirements at each individual node in B5G-IoT. Reference [57] emphasized on minimizing the latency in B5G-IoT networks based on the analog-to-digital compression radio-over-fiber approach. Reference [58] proposed an algorithm to maximize the number of served devices in B5G-IoT using nonorthogonal multiple access (NOMA). Reference [59] combines the features of artificial intelligence and blockchain for improving the performance of B5G-IoT edge system. They used a collaborative approach to address computational, storage, intelligence and connectivity

TABLE 1. Parametric comparison of state-of-the-art algorithms with the proposed approach.

Parameters / Algorithms	Optimizing Latency	Optimizing Power Consumption	Selection tool for appropriate metaheuristic approach	Minimal complexity	Hybrid Approach based on the requirements of B5G-IoT
Zhu *et. al.* [57]	✓	×	×	✓	×
Mlika & Cherkaoui [58]	×	✓	×	✓	×
Al Ridhawi *et. al.* [59]	✓	×	×	×	×
Ahmed *et. al.* [60]	✓	×	×	×	×
Mohammadkarimi *et. al.* [61]	×	×	×	✓	×
HLPA B5G-IoT	✓	✓	✓	✓	✓

issues in IoT nodes. Reference [60] maximized the usage of underutilized edge resources to minimize the turnaround time for applications in 5G IoT. They employed dynamic resource allocation to minimize service response time. Very recently, [61] proposed a new multiple access approach for massive machine-type communications in B5G-IoT. Their approach was designed to support a large number of IoT devices with sporadic traffic.

A. RESEARCH GAPS AND PROFOUND CONTRIBUTIONS

A majority of existing algorithms [52], [53], [54], [55], [56], [57], [58], [59], [60], [61] overlook the varying characteristics of B5G-IoT-edge systems. Furthermore, [52], [53], [54], [55] and [57], [58], [59], [60], [61] have ignored the heterogeneous nature and ever-increasing size of B5G-IoT networks. Load balancing and power minimization need distinct characteristics and parameter settings, and so cannot be optimized using a single approach. Power consumption is a critical concern in B5G-IoT networks due to the large number of battery-constrained IoT nodes, which demand light, efficient, fast, and scalable task-allocation strategies. Existing techniques in [52], [53], [54], [55], [56], [57], [58], [59], [60], [61] do not account for the aforementioned research gaps.

This paper develops a novel ACT that selects metaheuristic approaches based on the requirements of a IoT-edge network. A hybrid metaheuristic framework with the novel parameters ALI and EPC is proposed to minimize latency and power consumption in B5G-IoT-edge networks. GWO and BBO are used for optimization, which are efficient, light, and simple approaches that suit the needs of B5G-IoT networks. The evolutionary method BBO is selected for load balancing because its unique habitat structure ensures that user requests can be allocated to ENs. Unlike swarm-based algorithms such as PSO and BA, BBO has a mutation step that may be modified to improve server latency for delay-sensitive IoT applications. A parametric comparison of state-of-the-art art algorithms with proposed approach is presented in Table 1 that highlights the existing research gaps.

III. PROBLEM FORMULATION

End users use edge computing to run a variety of application programs. Each application program must be uploaded to an EN for processing, resulting in a large number of user requests. This process causes long end-user queues and high EN energy consumption, which necessitates the need for workload allocation and energy conservation.

As a hypothesis, B5G-IoT network consist of many small battery-operated nodes with low computational and memory capabilities. It is essential that the workload allocation and power-efficiency algorithms are lightweight and do not add unnecessary overhead to IoT nodes. Letting η_i denote the edge server response time to i^{th} user request and U_n be the number of incoming end users, the hypothesis can be expressed as

$$\eta_i \alpha U_n, \quad 1 < i < U_n \tag{1}$$

Equation (1) shows that the processing time for an incoming user request for the edge server increases as the number of incoming requests grows, which thereby delays the edge-server response in real-time applications. According to [62] and [63], there is a direct link between CPU utilization and a server's overall power consumption: The power consumed by a server grows linearly as CPU usage increases, which can be expressed as

$$P_S = S_i + (S_u - S_i) \times C_t \tag{2}$$

where P_S denotes the total power consumed by server, S_i is the power dissipated by a server in idle state, S_u denotes server power dissipation when it is fully utilized and C_t denotes the CPU utilization.

A. COMPUTATIONAL CAPABILITY

Computational capability is the maximum rate at which the server can process a task, e.g., $P_i^c \sim f_i$, where P_i^c denotes the computation capacity of the i_{th} EN and f_i is the CPU-cycle frequency. In this paper, we assume that $P_i^c \leq f_i$. Balancing the EN computational load during real-time incoming requests and minimizing EN power consumption can respectively be expressed as

$$Minimize\ X = IQR(L_i) \tag{3}$$

$$Minimize\ Y = \sum_{i=1}^{n} P_i \tag{4}$$

where L_i is the load assigned to the i^{th}EN, P_i is the power consumption in the i_{th} EN, and n is the total number of ENs in an IoT-edge system. Equation (3) minimizes the interquartile range (IQR) for the load assigned to each EN. IQR is analogous to standard deviation but provides more accurate data. Equation (4) minimizes the total power dissipation in an IoT-edge system.

IV. MODEL DESCRIPTION

A. SYSTEM MODEL AND TRAFFIC MODE

As a hypothesis, the network model in this research comprises of multiple IoT regions and ENs, as shown in Fig. 1. The computing requests for an IoT region are sent to the EN assigned to that region. The EN decides whether to handle the workload of an incoming request locally, move it to another EN, or send it to the cloud. Taking M IoT regions, we can consider unlimited computation resource for a cloud while the resource assigned to an incoming request by the edge server is limited. This research uses single carrier frequency division multiple access (SC-FDMA) and orthogonal frequency-division multiple access (OFDMA) for the uplink and downlink, respectively. SC-FDMA is a modified form of orthogonal frequency-division multiplexing and is preferred for uplink due to its low peak-to-average power ratio. OFDMA is used for downlink in long-term evaluation (LTE) and 5G applications due to its large bandwidth, low collision rate, and improved network performance [64], [65], [66]. The bandwidth for OFDMA in this paper is 10 MHz [66].

B. TYPES OF USER

IoT nodes are connected to a cellular network as end users in the proposed research. This paper considers a multiuser IoT-edge system in which end users utilize edge computing to execute various application programs. Each application must be uploaded to an edge node for processing. The proposed IoT network consists of a large number of battery-powered IoT nodes with limited computational and memory capacity.

C. DELAY MODEL

In the coming sections, $X_i(t)$ denotes the total number of computation jobs generated by an IoT region and processed by an EN, $E[X_i(t)]$ denotes the long term job generation rate, and V_r denotes the expected size of jobs in the i^{th} region given by $V_r = E[\frac{Xw_i(t)}{X_i(t)}]$, where $Xw_i(t)$ is the total aggregated workload.

There are three transmission paths in this paper: EN–cloud transmission path, EN–end-user transmission path, and the transmission path between neighboring ENs. Letting $w_{i,j}$ denote the transmission-route bandwidth from the i_{th} EN to the j_{th} EN, then the transmission delay for the incoming request can be expressed as

$$T^r_{delay(i,j)} = comm\,(i,j) + \frac{V_r}{w_{i,j}} \tag{5}$$

where $comm\,(i,j)$ is the network delay caused by factors such as congestion [67].

D. ENERGY CONSUMPTION MODEL

We use an energy model similar to that of [67], as discussed below.

1) COMPUTATION ENERGY CONSUMPTION IN AN EDGE NODE

Computation energy consumption ($Cw_i^F(t)$) for the i_{th} EN at time t is defined as a function of workload allocation. $Cw_i^F(t)$ can be expressed as

$$Cw_i^F(t) = u_f(Xw_i(t))^2 + v_f(Xw_i(t))^2 + w_f \tag{6}$$

where $u_f > 0$ and $v_f, w_f >= 0$ are factor parameters and $Xw_i(t)$ denotes the aggregated workload for time t [67].

2) TRANSMISSION ENERGY CONSUMPTION

The transmission energy consumption in path from i to j can be expressed as

$$Cw^{comm}_{i,j}(t) = u(i,j)^{¥w}(i,j)^{(t)} \tag{7}$$

where $u\,(i,j) > 0$ denotes the transmission power; $¥w(i,j)^{(t)}$ denotes the transmission workload for time t [67].

3) TOTAL ENERGY CONSUMPTION

In an IoT-edge system, the total aggregated energy consumption Cw is calculated as the sum of energy consumption in all computing nodes and all communication paths, which are calculated in (6) and (7), respectively. Accordingly, the total aggregated energy consumption in an IoT-edge system is calculated by summing (6) and (7), i.e.,

$$Cw = E[\sum_{i \in T} Cw_i^F(t) + \sum_{i \in T} \sum_{j \in T \cup G} Cw^{comm}_{i,j}(t)] \tag{8}$$

where T is the EN space as well as the IoT region space and G is the cloud space, which includes one cloud [67].

V. HYBRID LATENCY- AND POWER-AWARE MODEL

Edge computing enables end users in IoT-edge systems to access resources faster but at the expense of increased network computation latency and power consumption. Users must wait in line for lengthy periods of time, which makes it difficult to handle a large number of incoming requests. This paper proposes a hybrid approach for minimizing delay and power dissipation in B5G-IoT-edge systems. The proposed approach begins with ACT to select an appropriate methodology, after which it moves on to frameworks that achieve low latency and optimal energy dissipation in a B5G-IoT-edge system.

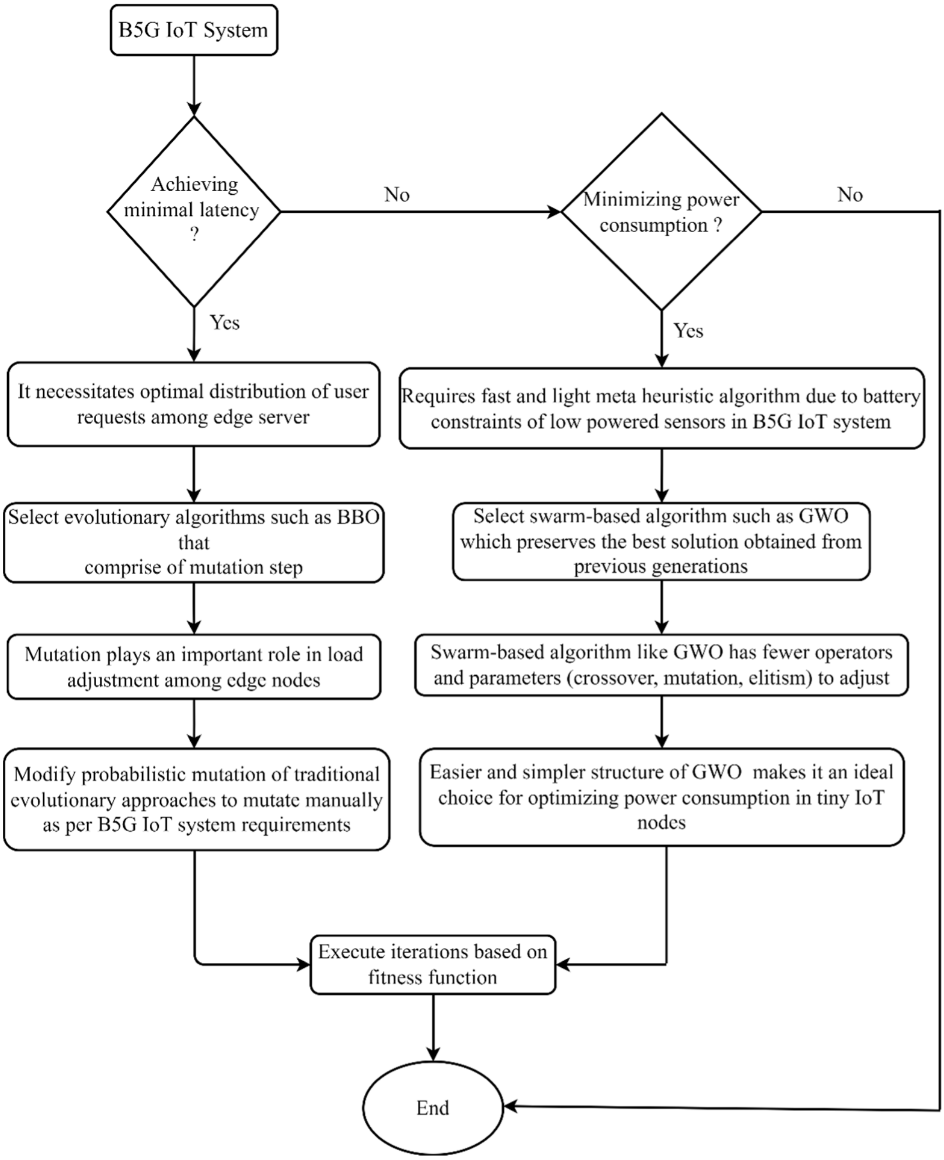


FIGURE 2. ACT flowchart.

A. ALGORITHM CLASSIFIER TOOL TO MINIMIZE LATENCY AND POWER CONSUMPTION

Selecting an appropriate algorithm is crucial for minimizing latency and power consumption in B5G-IoT-edge systems. This research investigates the properties and requirements of the aforementioned performance metrics and proposes an algorithm selector. Load balancing necessitates the optimal distribution of user requests among edge servers, and swarm-based algorithms such as PSO and GWO may not produce good results because they do not have a mutation step in their optimization process. Alternatively, evolutionary algorithms such as GA and BBO include a mutation process, which might be useful in load adjustment.

In traditional evolutionary approaches, the candidate solution mutates based on a random probability, which is ineffective for load management because there is no control over which genes are mutated in a candidate solution. Accordingly, this research presents a modified mutation mechanism in which each gene (mapped to an edge server in a B5G-IoT-edge system) may be manually mutated to balance the load and minimize latency. This approach is implemented in Section V.B. (6).

Minimizing power requires fast and light metaheuristic algorithms due to the battery constraints of low-powered sensors in B5G-IoT systems. The ever-increasing size of future IoT necessitates limiting energy dissipation in nodes with minimal overhead. Evolutionary algorithms such as GA and BBO may be inefficient for this purpose, as they are heavy, complex, and have more parameters to adjust, which results in significant overhead. Swarm-based algorithms, such as GWO, have fewer operators and parameters to adjust since they do not involve crossover, mutation, and elitism, and

TABLE 2. ACT matrix.

B5G-IoT-Edge System Parameters	Requirements	Evolutionary Algorithm (BBO)	Swarm-Based Approach (GWO)	Remedy
Load balancing	Mutation operator	✓	×	Not applicable
	Manual mutation	×	×	Traditional mutation step is modified to meet the requirements of B5G-IoT-edge systems
Suitable for load balancing?		✓	×	
Low power consumption	Lightweight	×	✓	Not applicable
	Less operators	×	✓	Not applicable
	Less parameters	×	✓	Not applicable
	Preserves best solution from previous iterations	×	✓	Not applicable
Suitable for low power consumption?		×	✓	Not applicable

they can be used to optimize power consumption in tiny IoT nodes. Furthermore, unlike evolutionary algorithms, swarm-based approaches preserve the best solution obtained from previous generations. The ACT flowchart is shown in Fig. 2. Table 2 shows a matrix of B5G-IoT system requirements and metaheuristic algorithm attributes. The matrix can be used as a part of the ACT to determine the optimal algorithm based on system requirements.

From the ACT, it is evident that evolutionary algorithms and swarm-based approaches are well-suited to balancing load and minimizing power consumption in a B5G-IoT-edge system, respectively.

B. MINIMAL-LATENCY SIXTH-GENERATION INTERNET-OF-THINGS SYSTEM USING ACTIVE LOAD INDEX

A B5G-IoT system receives a large number of requests from IoT nodes, which are handled by edge servers. This paper uses a novel parameter, i.e., ALI, to balance the load among ENs.

1) ACTIVE LOAD INDEX VECTOR

For the i^{th} EN (E_{n_i}), ALI can be expressed as

$$ALI_i = \frac{(E_{energy}(i) - (E_{diss}(i))}{(E_{diss}(i))} \tag{9}$$

where $E_{energy}(i)$ is the initial energy of E_{n_i} and $E_{diss}(i)$ is the energy dissipated by E_{n_i}. In essence, ALI_i is the approximate lifetime of E_{n_i} in terms of rounds. The proposed algorithm computes the ALI for each EN in an IoT-edge system and stores it in a vector, as shown in Table 3. Equation (9) is used to calculate the ALI value of E_{n_i}. The ALI-vector size is equal to the total number of ENs (n) in the network. For implementation purposes, the ALI values are rounded off and stored as integers in the ALI vector. These ALI values are important in load balancing and are utilized to calculate the fitness function in Section V.B. (3).

After formulating the ALI vector, the problem is mapped to BBO and followed by implementation.

TABLE 3. ALI vector.

ALI vector	ALI_i	ALI_{i+1}	ALI_{i+2}	.	.	ALI_n

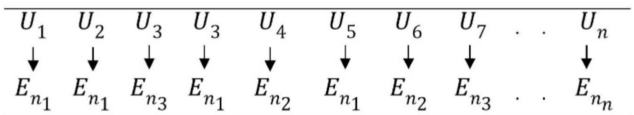


FIGURE 3. BBO habitat representing the assignment of end users to ENs.

2) MAPPING OF WORKLOAD-ALLOCATION PROBLEM

A candidate solution is represented by a habitat in BBO. The habitat suitability index (HSI) represents the fitness of a candidate solution [15]. The HSI values for each candidate solution are calculated and ranked accordingly. Letting the edge server consist of n number of ENs to handle the incoming requests in an IoT-edge system, BBO is initialized by randomly assigning end users ($U_1, U_2 \ldots . . U_n$) to respective ENs ($E_{n_1}, E_{n_2}, E_{n_3} \ldots . . E_{n_n}$) in a habitat, as shown in Fig. 3.

3) FITNESS FUNCTION

The fitness function (F_1) is used to determine the fitness for each habitat in BBO. It minimizes the IQR for the ALI of all ENs, which results in an equal EN lifespan, and it can be expressed as

$$F_1 = Minimize\,[IQR\,(ALI)] \tag{10}$$

By providing an equal lifespan, the workload is evenly distributed among the ENs in an IoT-edge system.

4) MIGRATION IN BIOGEOGRAPHY-BASED OPTIMIZATION

Migration maintains population diversity; it is performed according to immigration rate (λ) and emigration rate (μ), which can be expressed as

$$\lambda_i = I \times \left(1 - \frac{k_i}{H_t}\right) (11) \tag{11}$$

$$\mu_i = E \times \left(\frac{k_i}{H_t}\right) \tag{12}$$

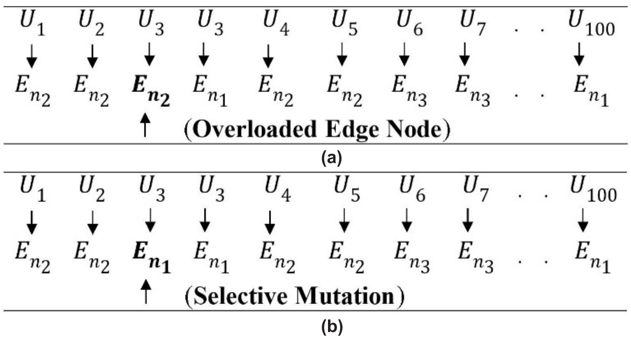


FIGURE 4. (a). Identifying overloaded ENs. (b). Selective mutation process in BBO.

where I and E are the maximum values of λ and μ, respectively, k_i is the rank of habitat (h_i) and H_t denotes the total number of habitats [15].

5) MUTATION IN BIOGEOGRAPHY-BASED OPTIMIZATION

Mutation in BBO randomly mutates the habitat according to mutation probability m_i, which can be expressed as

$$m_i = m_{max} \times \left(1 - Pr_i / p_{max}\right) \tag{13}$$

where m_{max} is the maximum probability of mutation, Pr_i is the probability of a solution existing and p_{max} is the maximum probability of Pr_i [15].

6) MODIFIED MUTATION IN BIOGEOGRAPHY-BASED OPTIMIZATION

This research presents modified (selective) mutation, which plays a pivotal role in balancing EN load. From the habitat representation of end users and ENs in Fig. 4(a), it is evident that E_{n_2} is overloaded, as it is assigned the maximum number of end users during the workload-allocation process. As shown in Fig. 4(a)–(b), selective mutation identifies the least loaded ENs (E_{n_1} is least loaded in this case, as it is assigned two end users), removes the end user assigned to the overloaded ENs (E_{n_2} in this case), and reassigns it to the least loaded ENs (E_{n_1}). The selective mutation not only reduces the load of overloaded ENs but also balances the overall network load.

The proposed algorithm minimizes latency in a B5G-IoT-edge system by ensuring equal workload distribution among all ENs. Algorithm (1) shows the pseudocode for workload allocation and low latency based on BBO.

Algorithm 1 Workload-Allocation Approach for Load Balancing in B5G-IoT-Edge Systems Using BBO

```
1.  Use ACT to select appropriate algorithm for load balance.
2.  Begin with population size Ps_i and maximum iterations
      I_max.
3.  While iteration < I_max, for each iteration do
4.  |   for each habitat h_i do
5.  |   |   Workload assignment process
6.  |   |   Clone ENs and end users to h_i.
7.  |   |   Load balancing process
8.  |   |   To balance the load, compute the fitness of each
    |   |     habitat using (10).
9.  |   |   Perform migration using (11) and (12).
10. |   |   Perform modified mutation as per Fig. 4 (a–b).
11. |   |   Chose the habitat with the best fitness.
    |   end
    end
12. Optimum workload allocation and balanced IoT-edge
    system is obtained after BBO convergence.
```

C. MINIMIZING POWER CONSUMPTION IN SIXTH-GENERATION INTERNET-OF-THINGS SYSTEMS USING EFFECTIVE POWER COEFFICIENT

1) EFFECTIVE POWER COEFFICIENT

In large IoT WSNs, data forwarding via lengthy transmission routes consume a significant amount of energy. This paper presents EPC to minimize power consumption in a B5G-IoT system, which is calculated as a combination of transmission energy (ρ) and effective residual energy (Ψ), the former of which can be expressed as

$$\rho = \frac{1}{n} \sum\nolimits_{i=1to}^{n} (\Phi + \kappa) \tag{14}$$

where κ denotes the energy dissipated in data transmission between any two consecutive ENs and Φ denotes the energy consumed in data transmission from an EN to its respective end user in an IoT-edge system which in turn depends on the distance between an EN and its respective end user as expressed below.

$$\Phi = average\ of\ distance\ between \begin{bmatrix} \left(U_1, E_{n_1}\right)\left(U_2, E_{n_2}\right)\left(U_3, E_{n_1}\right)\left(U_4, E_{n_2}\right)\ldots\ldots \\ \ldots\left(U_n, E_{n_n}\right) \end{bmatrix} \tag{15}$$

In a B5G-IoT WSN, power is dissipated after each round. It is important to select a network scenario where total network residual power is high. Ψ is the sum of network residual energy and node residual energy. Accordingly, EPC can be expressed as

$$EPC = \rho + \frac{1}{\Psi} \tag{16}$$

2) MAPPING INTERNET-OF-THINGS EDGE SYSTEMS TO GREY WOLF OPTIMIZATION

GWO optimization is based on wolf hunting, with top three fittest wolves denoted by α, β, and δ, and the rest denoted by ω [16]. GWO converges to the optimum solution by an exploration–exploitation process. In exploration, the wolves search for the best solution, and, in exploitation, they converge to the optimum solution [16].

The power-efficiency algorithm involves mapping the problem to GWO, followed by implementation. Each wolf (w_i) is denoted by a vector of size n, as shown in Table 4.

A wolf (w_i) contains many genes (g), which, in the vector, are represented as 1 for ENs and 0 for end users, and many

TABLE 4. A WOLF in GWO.

Wolf W_i	1	0	0	0	0	1	0	.	.	n

wolves are present in the initial GWO population:

$$w_i(g) = \begin{cases} 1, & if\ g\ is\ an\ edgenode \\ 0, & if\ g\ is\ an\ enduser \end{cases} \tag{17}$$

The fitness function for power efficiency (F_2) minimizes EPC, which can be expressed as

$$F_2 = Minimize\,(EPC) \tag{18}$$

3) ENCIRCLING THE PREY (EXPLORATION)

After calculating the fitness, the wolf approaches and encircles its prey (optimum solution) to cease its movement [16]. Mathematically, we represent this as

$$\left|\vec{C}.\vec{X}_p(t) - \vec{X}(t)\right| \tag{19}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}.\vec{D} \tag{20}$$

where t is the current iteration and $\vec{X}_p$ and $\vec{X}$ are the prey's position vector and wolf's position vector, respectively [16]. Vectors $\vec{A}$ and $\vec{C}$ are calculated as

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \tag{21}$$

$$\vec{C} = 2\vec{r}_2 \tag{22}$$

Random vectors $\vec{r}_1$ and $\vec{r}_2$ are enclosed in interval [0,1] [16].

4) HUNTING (EXPLOITATION)

In GWO, the α, β, and δ wolves are closest to the prey's position (optimum solution). Hence, the updated positions of α, β, and δ are saved as

$$\vec{D}_\alpha = \left|\vec{C}_1.\vec{X}_\alpha - \vec{X}_{w_i}\right| \tag{23}$$

$$\vec{\vec{D}}_\beta = \left|\vec{C}_2.\vec{X}_\beta - \vec{X}_{w_i}\right|, \tag{24}$$

$$\vec{D}_\delta = \left|\vec{C}_3.\vec{X}_\delta - \vec{X}_{w_i}\right|, \tag{25}$$

where $\vec{\vec{X}}_{w_i}$ is the position of wolf w_i, $\vec{X}_\alpha$ is the position of α wolf, and $\vec{D}_\alpha$ denotes the updated position of α. Similarly, $\vec{X}_\beta$ is the position of β wolf and $\vec{D}_\beta$ denotes the updated position of β. $\vec{X}_\delta$ is the position of δ wolf, and $\vec{D}_\delta$ is the updated position of δ. $\vec{C}_1$, $\vec{C}_2$, and $\vec{C}_3$ are calculated as per (22) [16]. For the present iteration, the wolf positions ($\vec{X}_1$, $\vec{X}_2$, and $\vec{X}_3$) are calculated as

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \tag{26}$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \tag{27}$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \tag{28}$$

Algorithm 2 Power Efficiency in B5G-IoT-Edge Systems Using GWO

1. Use **ACT** to select appropriate algorithm for power efficiency.
2. Begin with population size Ps_i and maximum iterations I_{max}.
3. GWO mapping
  a) While $iteration < I_{max}$ **for each** iteration **do**
    i) **For each** wolf w_i **do**
    ii) Clone ENs and end users to w_i.
  b) Power-efficiency process
    i) To optimize the transmission distance, compute the fitness of each wolf using (18).
    ii) Update leader wolves $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$.
    iii) Update the wolves' position using (29).
    end
    end
  c) A power-efficient IoT-edge system is obtained after GWO convergence.

where $\vec{A}_1, \vec{A}_2,$ and $\vec{A}_3$ are calculated according to (21) [16]. Finally, the wolf updates its position based on the best solutions for α, β, and δ [16]. Therefore, the optimum solution can be expressed as

$$\vec{X}(t+1) = \left(\vec{X}_1 + \vec{X}_2 + \vec{X}_3\right)/3 \tag{29}$$

In HLPA B5G-IoT, EPC is optimized using GWO, which minimizes κ and Φ, and maximizes Ψ. As a result, EPC optimization reduces the energy consumption in battery-constrained IoT nodes with minimum overhead while ensuring QoS for end users. Furthermore, the algorithm classifier tool carefully selects the metaheuristic approach GWO for optimizing EPC due to its simple structure and implementation. Since GWO does not involve crossover, mutation, or elitism, it has fewer operators and parameters to adjust; hence, it is used to optimize power consumption in tiny IoT nodes with minimum overhead. Algorithm (2) shows the pseudocode for the power-efficiency algorithm, and the detailed architecture of HLPA B5G-IoT is shown in Fig. 5.

VI. EXPERIMENTAL RESULTS

The performance of HLPA B5G-IoT was evaluated and tested for load balancing and power efficiency with seven state-of-the-art algorithms: SA [39], GA [40], PSO [41], BFA [42], ACO [43], BA [44], and GSP [45]. The results for load balancing and power efficiency in an IoT-edge system are shown in Section VI.A and VI.B, respectively. Algorithm complexity is explained in section VI.C. Three IoT regions with ENs were considered, as shown in Table 5. All of the experimental results are the average of values obtained from 30 independent runs of the proposed algorithm. The initial computation power of each EN was set to 2.0 GHz in IoT regions 1-3. The boundary limits of the parameters are given in Table 6.

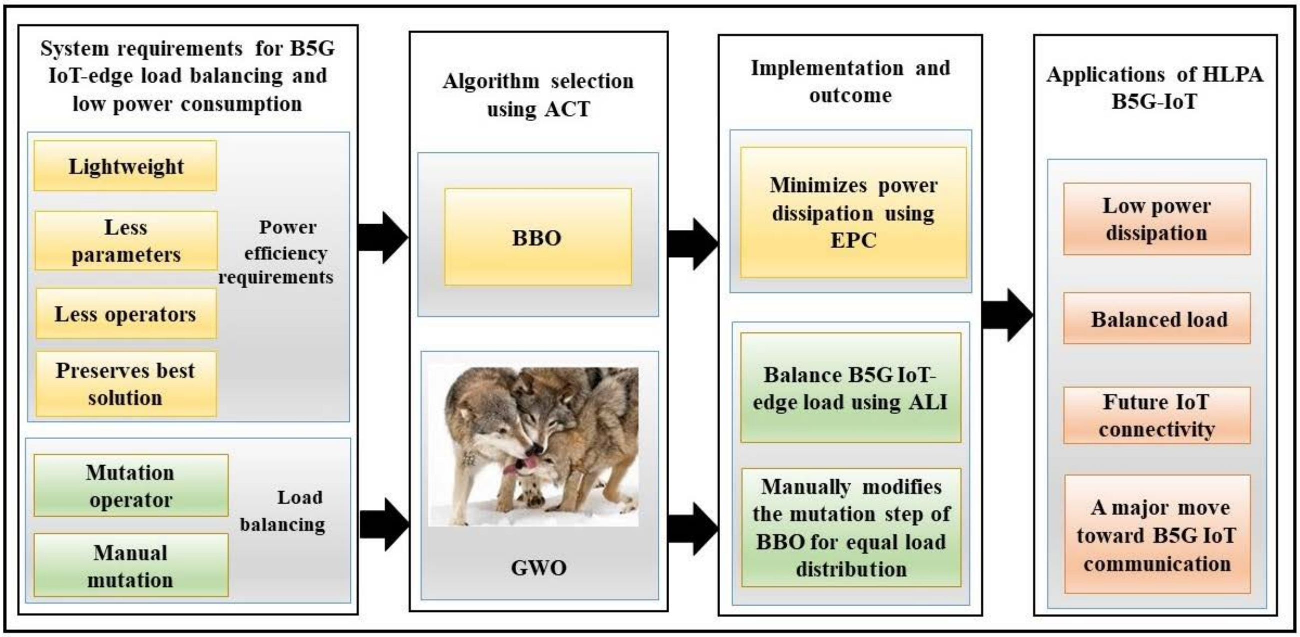


FIGURE 5. HLPA B5G-IoT architecture.

TABLE 5. Parameters.

Parameters	Region 1	Region 2	Region 3
Initial computation power of ENs (GHz)	2	2	2
Energy parameters (u_f, v_f, and w_f)	(0.18×10^{-12}), 0, 0		
Job-generation rate (jobs/ms)	10	15	10
Mean job size (V_r)	0.7	1.4	0.7
Computation resource (P^c)	3.2		

A. LOAD BALANCING AND MINIMIZING LATENCY IN AN INTERNET-OF-THINGS EDGE SYSTEM

Fig. 6 compares the service time (measured in milliseconds (ms)) for multiple incoming IoT operations (measured in megabytes (MB)) from end users, from which it is evident that SA and GA have the highest rate of increase in service time. This is because, when applied to complex optimization, they are often time consuming and, as a result, cannot converge to high-quality solutions in a predefined number of iterations. Alternatively, HLPA B5G-IoT uses the BBO workload-allocation approach to optimally allocate different types of application requests to the corresponding ENs and thereby improve the EN-allocation process and minimizes service latency. In Fig. 6, the x-axis indicates the number of incoming requests with request sizes ranging from 2 to 100 MB. In particular, HLPA B5G-IoT minimizes the service delay

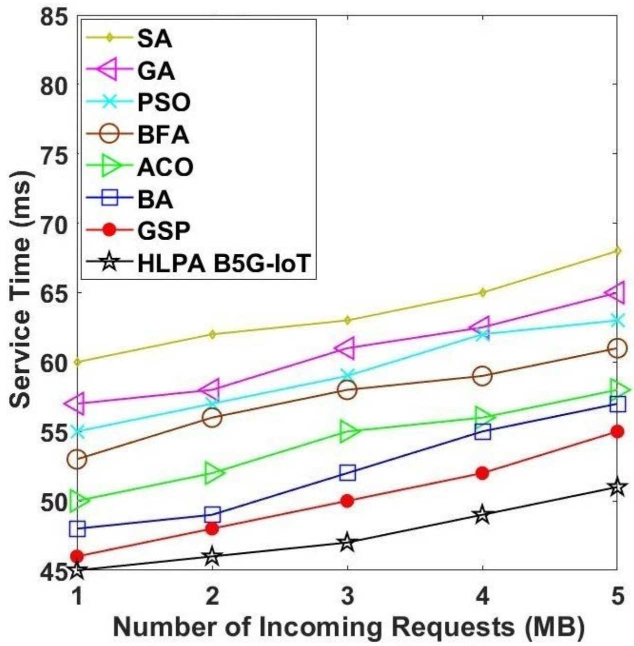


FIGURE 6. Service time (milliseconds) for different incoming application requests.

by 33.33%, 27.45%, 23.52%, 21.56%, 13.72%, 11.76%, and 7.84% compared with SA [39], GA [40], PSO [41], BFA [42], ACO [43], BA [44], and GSP [45], respectively.

Fig. 7 compares HLPA B5G-IoT with SA [39], GA [40], PSO [41], BFA [42], ACO [43], BA [44], and GSP [45] in terms of average latency with varying incoming request size. In general, as the incoming request size increases, so does the average latency. This is because more data must be processed

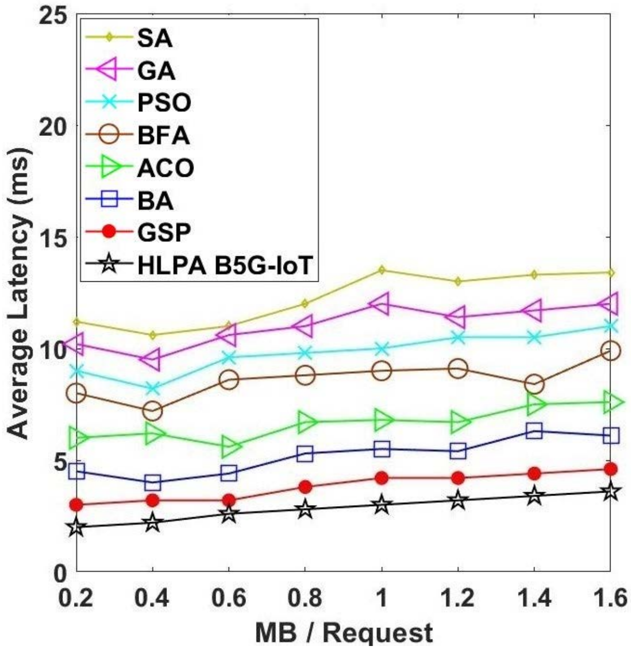


FIGURE 7. Average latency vs. incoming request size.

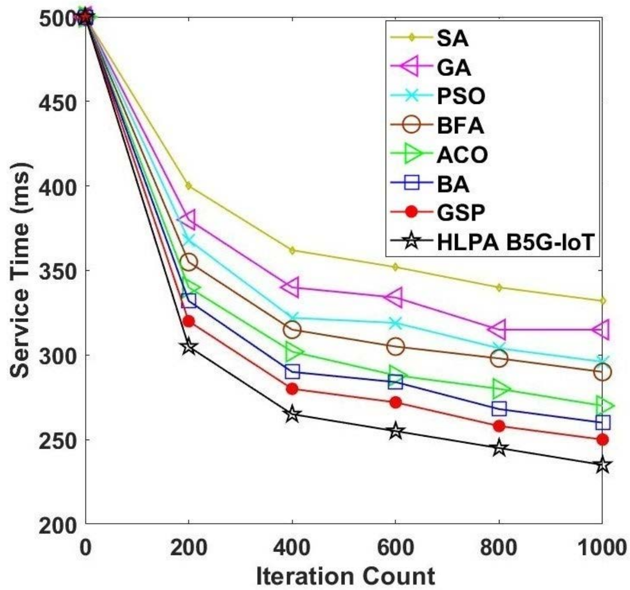


FIGURE 8. Comparison of HLPA B5G-IoT with existing approaches in terms of variation in service time and convergence.

TABLE 6. Boundary limits.

Parameters	Minimum	Maximum
BBO iterations	0	1,500
GWO iterations	0	1,500
Job-generation rate (jobs/ms)	0	15
Mean job size (V_r)	0.7	1.4
Cycles/bite (¥)	0	2,100

TABLE 7. Statistical analysis of HLPA B5G-IoT.

Algorithm	Best	Worst	Mean	Standard Deviation
SA	1.5578×10^8	1.5634×10^8	1.5602×10^8	1.9578×10^5
GA	1.5546×10^8	1.5622×10^8	1.5580×10^8	1.3875×10^5
PSO	1.5502×10^8	1.5592×10^8	1.5540×10^8	2.9875×10^5
BFA	1.5492×10^8	1.5546×10^8	1.5514×10^8	2.1248×10^5
ACO	1.5246×10^8	1.5298×10^8	1.5269×10^8	1.1548×10^5
BA	1.5203×10^8	1.5244×10^8	1.5219×10^8	7.5485×10^4
GSP	1.5055×10^8	1.5122×10^8	1.5083×10^8	7.2547×10^4
Proposed	1.5002×10^8	1.5042×10^8	1.5018×10^8	8.5819×10^4

when the incoming request size grows. For this paper, the incoming-request scale is between 0.2 and 1.6 MB/request. From Fig. 7, it is evident that SA has the maximum average latency, as it fails to converge to an optimal solution. The proposed algorithm minimizes average latency by distributing the load across multiple ENs and thereby reduces the queuing delay for incoming user requests. Indeed, HLPA B5G-IoT operates with significant low latency compared with existing approaches.

Fig. 8 compares the proposed algorithm with SA [39], GA [40], PSO [41], BFA [42], ACO [43], BA [44], and GSP [45] in terms of convergence, and variation in service time, from which it is evident that HLPA B5G-IoT has a shorter service time and converges faster than the other algorithms. In particular, the proposed algorithm converges 41.27%, 34.04%, 25.95%, 23.40%, 14.89%, 10.63%, and 6.38% faster than SA [39], GA [40], PSO [41], BFA [42], ACO [43], BA [44], and GSP [45], respectively.

Fig. 9 shows the bar graph for load distribution among nodes in pure cloud computing (PCC), pure edge computing (PEC) and HLPA B5G-IoT for low, moderate, and high incoming-request volumes. PEC is a network scenario that operates without power efficiency and load balancing algorithms, whereas in PCC, all jobs are sent to the cloud server. As seen in Fig. 9, the load in PCC and PEC is not evenly distributed, resulting in long queuing delays and latency. Alternatively, the proposed approach ensures a balanced load distribution among ENs in the IoT-edge server, resulting in an even bar graph.

Fig. 10 compares HLPA B5G-IoT with SA [39], GA [40], PSO [41], BFA [42], ACO [43], BA [44], and GSP [45] in terms of average latency and incoming requests. In general, average latency increases as the rate of incoming end-user requests grows. This is because an edge server will require more time to process growing number of requests,

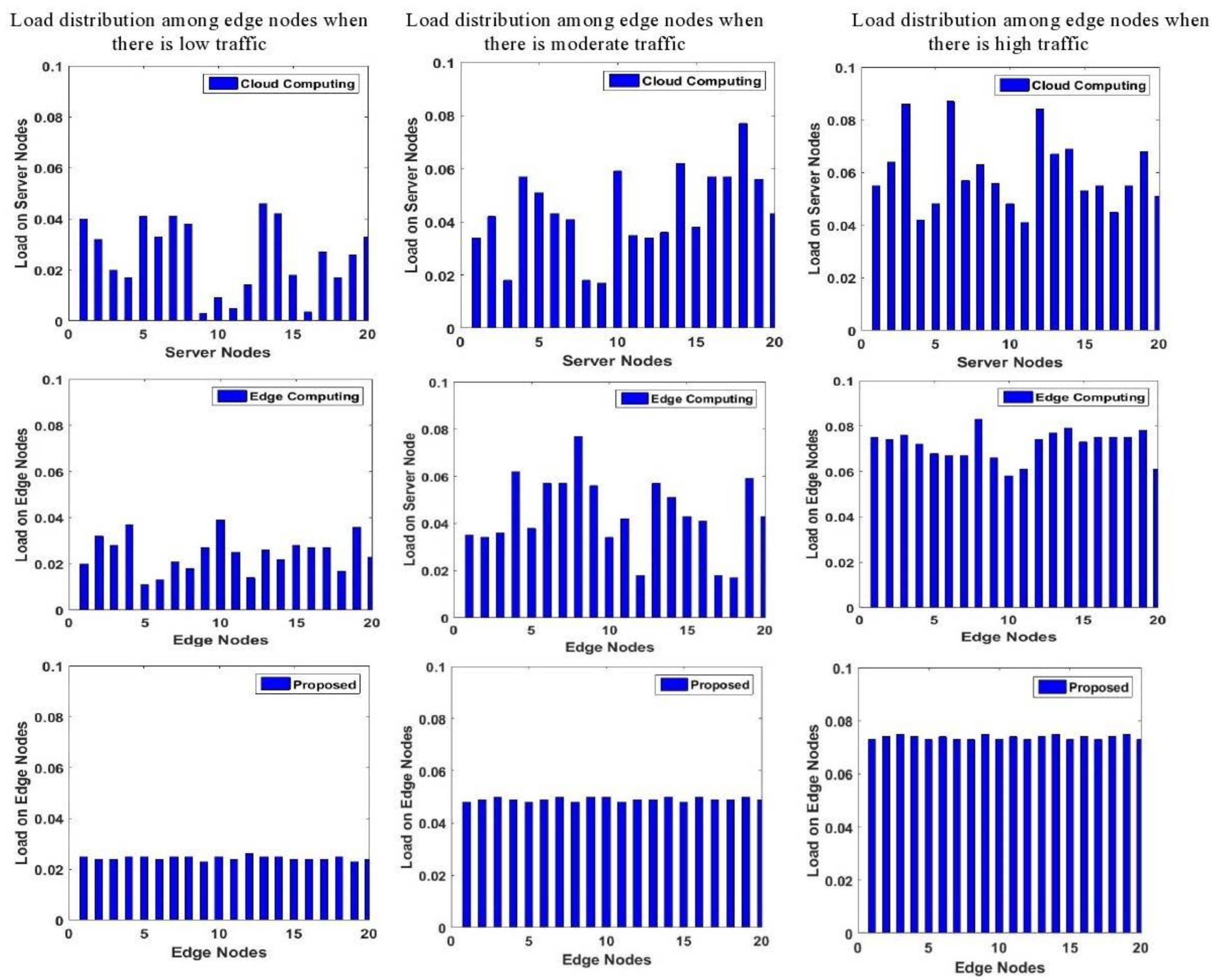


FIGURE 9. **A comparison of load distribution among server odes in pure cloud computing, pure edge computing, and the proposed algorithm.**

which, in turn, increases the server latency. As seen in Fig. 10, SA and GA have the maximum latency due to their weak convergence, and, therefore, they cannot manage a burst of incoming user requests, which results in long queuing delays. The proposed algorithm balances the EN load and optimizes workload allocation using BBO. This reduces service latency in an IoT-edge system. For statistical analysis, the proposed HLPA B5G-IoT algorithm was run 30 times. Table 7 compares the results with state-of-the-art algorithms.

B. POWER EFFICIENCY IN SIXTH-GENERATION INTERNET-OF-THINGS EDGE SYSTEM

Power consumption is a serious issue in B5G-IoT networks due to their ever-increasing size. Both industrial and domestic IoT networks are equipped with battery-operated sensor nodes that are constrained by limited power. In this section, HLPA B5G-IoT performance is tested for power efficiency in the network for a variety of incoming user requests. The parameters used for implementation are listed in Table 5. To ensure fair analysis of the impact of request size on power consumption and end-to-end latency in IoT-edge systems, incoming user requests were varied in size from 0.2 to 1.6 Mb/request.

Fig. 11 shows the overall power consumption (measured in Joule (J)) of a B5G-IoT-edge system with various edge-server computation capacities. Edge-server scaling ratios were set to 0.6, 0.8, 1, 1.2, and 1.4. As seen in Fig. 11, overall power consumption reduces with an increase in the processing capacity of edge servers. For the proposed algorithm, the power consumption is smaller than that of existing approaches for all types of edge servers. This is because GWO efficiently optimizes the EPC in HLPA B5G-IoT, which minimizes power dissipation. In particular, HLPA B5G-IoT reduces power consumption in the system by 46.6%, 40%, 32.2%, 27.7%, 15.5%, 11.1%, and 6.6% compared with SA [39], GA [40], PSO [41], BFA [42], ACO [43], BA [44], and GSP [45], respectively.

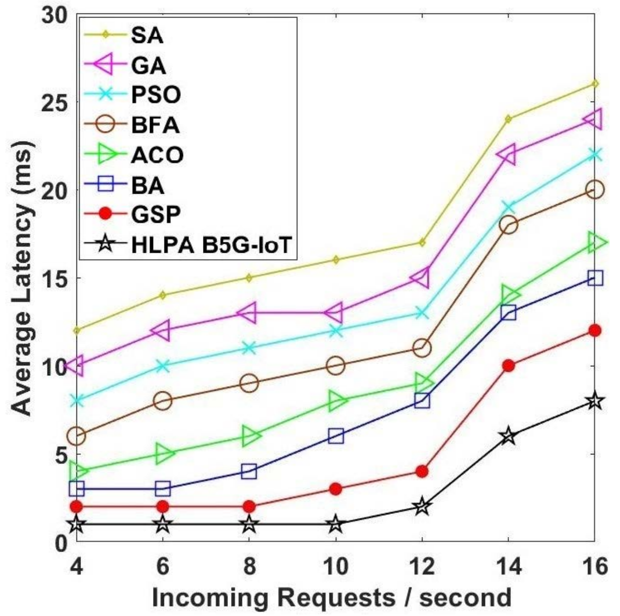


FIGURE 10. Average delay vs. number of incoming requests.

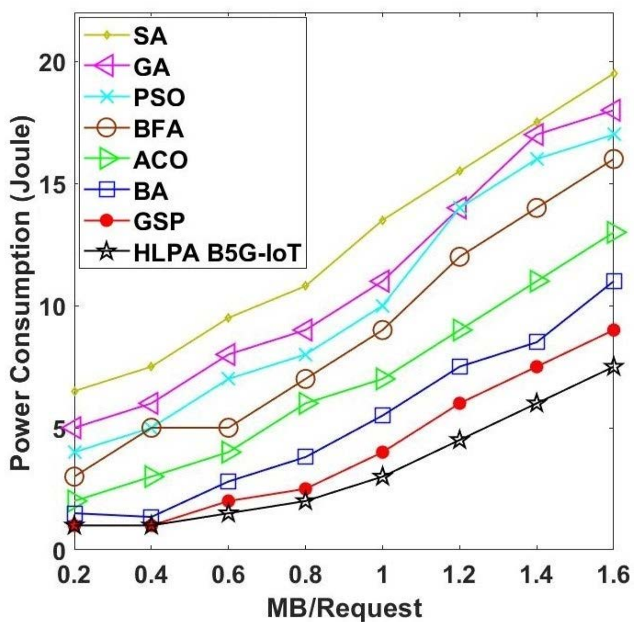


FIGURE 12. Power consumption of the network vs. incoming request size.

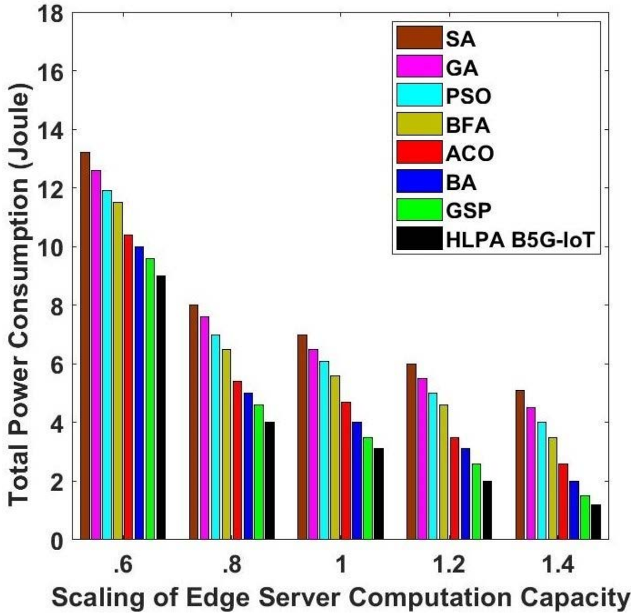


FIGURE 11. Power consumption in an IoT-edge system vs. computation capacity.

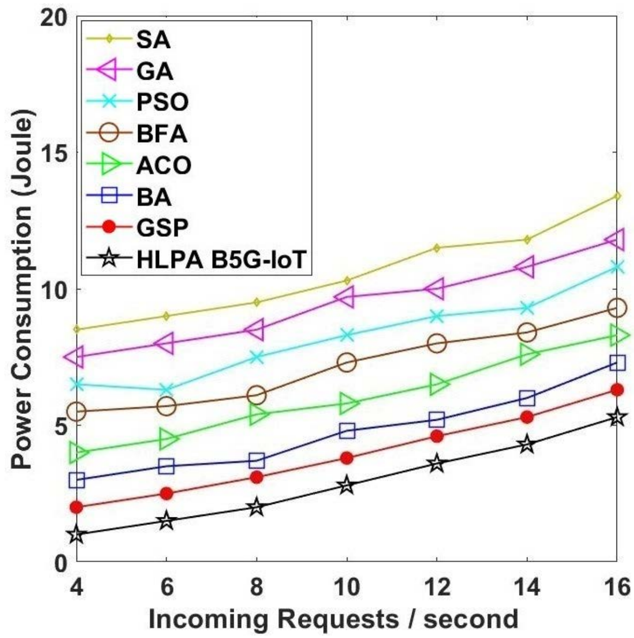


FIGURE 13. Power consumption of the network vs. number of incoming requests.

Fig. 12 compares HLPA B5G-IoT with SA [39], GA [40], PSO [41], BFA [42], ACO [43], BA [44], and GSP [45] in terms of power consumption and request size, from which it is evident that the power consumption of HLPA B5G-IoT increases with an increase in the size of user requests. This is because, with growing user-request size, more servers are required, which increases the energy consumption. The power dissipation in SA and GA grows rapidly with a rise in user-request size. This is because the search scope for SA and GA is local at each iteration. Alternatively, HLPA B5G-IoT performs optimally for all incoming request sizes. HLPA B5G-IoT can adapt its search capabilities to the current situation and iteratively evolve globally. It can also dynamically switch between local ENs and the cloud server according to the size of incoming requests. Moreover, the switching process is optimized during dynamic workload allocation in IoT-edge systems, which thereby reduces power consumption.

Fig. 13 compares the power consumption of HLPA B5G-IoT with SA [39], GA [40], PSO [41], BFA [42], ACO [43], BA [44], and GSP [45], from which it is evident that the energy-consumption trend is identical to that in Fig. 12.

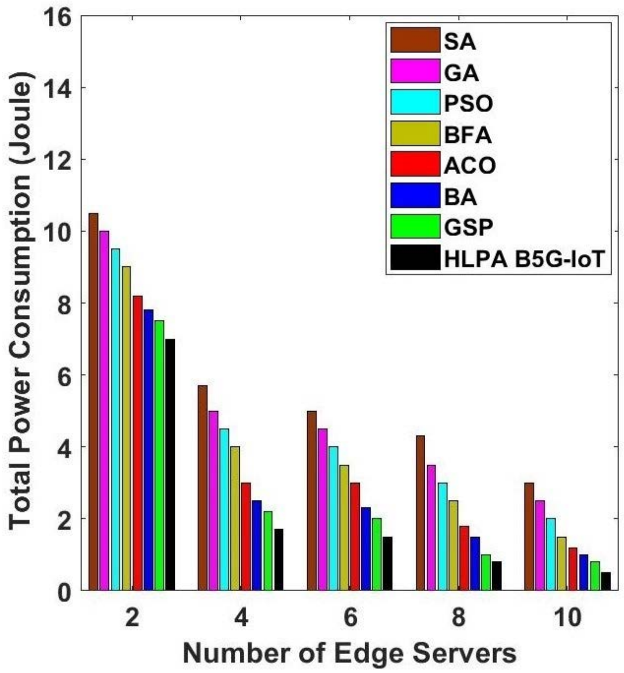


FIGURE 14. Power consumption in an IoT-edge system vs. number of edge servers.

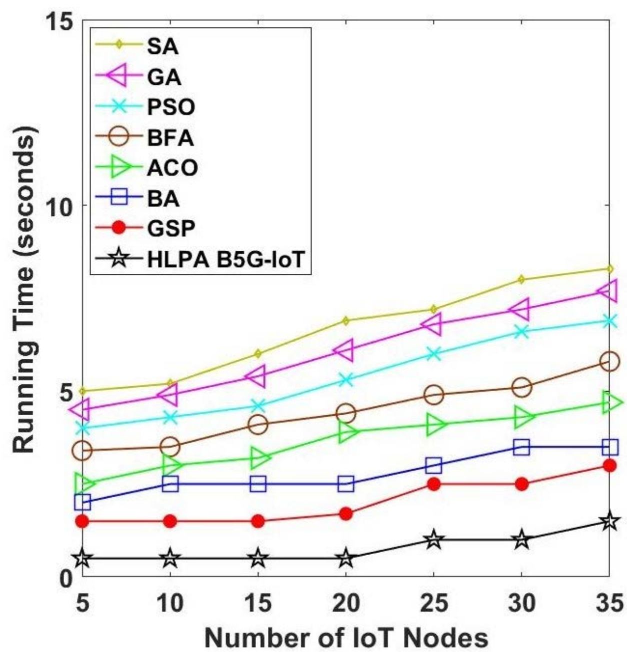


FIGURE 15. Algorithm complexity comparison.

In general, the amount of power consumed increases because, as the number of user requests grows, so does the number of required servers. SA is slow and has poor performance, and ACO fails to produce a high-quality solution. The proposed approach employs GWO to minimize energy consumption in transmission channels from the edge to the cloud in IoT-edge systems, and, therefore, the total network power consumption decreases.

Fig. 14 presents the system power consumption vs. number of edge servers. The number of edge servers for this experiment was set to 2, 4, 6, 8 and 10. Generally, the system power consumption deceases with an increase in the number of edge servers. On the other hand, when the number of edge servers is small, offloading techniques deploy more resources to the cloud to fulfil deadline constrains, resulting in excessive power consumption. HLPA B5G-IoT consumes less power than its predecessors. In particular, it minimizes power consumption by 50%, 42.85%, 35.71%, 28.57%, 17.14%, 11.42%, and 7.14% compared with SA [39], GA [40], PSO [41], BFA [42], ACO [43], BA [44], and GSP [45], respectively.

C. COMPLEXITY ANALYSIS OF HLPA B5G-IoT

In the initialization phase HLPA B5G-IoT has complexity $O(f_p \times z_p)$, where f_p denotes the total number of agents in the population and z_p is the dimension of the problem. The fitness function calculation and updating of agents' positions also require complexity $O(f_p \times z_p)$. Hence, the complexity for each generation is $O(f_p \times z_p)$. The overall complexity for maximum number of iterations is $O\left(f_p \times z_p \times MaxIter\right)$, where $MaxIter$ represents the maximum number of iterations. The computation time of GWO is lower than that of many other metaheuristic approaches such as GA, PSO and harmony search algorithm [68]. Fig. 15 compares HLPA B5G-IoT with state-of-the-art algorithms such as SA [39], GA [40], PSO [41], BFA [42], ACO [43], BA [44], and GSP [45] for algorithm complexity. It is seen from Fig. 15 that the proposed approach takes the least time to complete the operations of an IoT-edge system.

VII. CONCLUSION

Although 5G networks are IoT compliant, B5G networks ought to be IoT and "internet of everything" compliant. Therefore, this paper proposed a hybrid latency- and power-aware method for B5G-IoT networks that minimizes latency with minimum overhead on battery-constrained IoT nodes and offers a power-efficient solution for IoT-edge networks. In addition, an ACT was presented to enable the selection of ideal metaheuristic algorithms based on the characteristics and requirements of B5G-IoT systems. Coordination of algorithm attributes and system requirements is necessary for performance improvement. A B5G-IoT server is expected to handle many delay-sensitive applications that require latency-free processing. Excessive power consumption in edge servers (both near and far edge) degrades network performance. To address this, HLPA B5G-IoT carefully modifies BBO and employs GWO to fulfil the requirements of load balancing and power efficiency in IoT-edge systems. The load-balancing algorithm reduces the network latency by 33.33%, 27.45%, 23.52%, 21.56%, 13.72%, 11.76%, and 7.84% compared with SA, GA, PSO, BFA, ACO, BA, and GSP, respectively. The power-efficiency algorithm reduces the power consumption by 46.6%, 40%, 32.2%, 27.7%, 15.5%, 11.1%, and 6.6% compared with SA, GA, PSO, BFA, ACO, BA, and GSP in IoT-edge systems, respectively. HLPA B5G-IoT paves the way for future IoT connectivity, as it reduces power consumption and latency in IoT-edge systems. 5G plays an important role in contemporary IoT, but, for B5G-IoT networks, the increased usage of automated IoT systems and data-centric services will require an upgrade in existing communication standards. HLPA B5G-IoT is a coherent and creative approach to improve the performance of IoT-edge systems by optimizing crucial IoT parameters and variables, and, therefore, it is a major move forward in B5G-IoT communication.

HLPA B5G-IoT minimizes latency in IoT networks, which is useful for latency-sensitive applications such as industrial process monitoring, driverless vehicles, virtual reality, security and surveillance. The proposed framework is also useful in applications requiring real-time communication such as healthcare, telecommunications, and high-speed trains. The proposed algorithm reduces power consumption in IoT nodes. Low-power IoT nodes offer a wide range of applications including home automation, smart warehouses, wearable monitoring devices, transportation and logistics. Future research may include more QoS factors such as data rate, priority and improved spectral efficiency.

REFERENCES

[1] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. M. Leung, "Enabling massive IoT toward 6G: A comprehensive survey," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 11891–11915, Aug. 2021.

[2] J. R. Bhat and S. A. Alqahtani, "6G ecosystem: Current status and future perspective," *IEEE Access*, vol. 9, pp. 43134–43167, 2021.

[3] H. Viswanathan and P. E. Mogensen, "Communications in the 6G era," *IEEE Access*, vol. 8, pp. 57063–57074, 2020.

[4] I. F. Akyildiz, A. Kak, and S. Nie, "6G and beyond: The future of wireless communications systems," *IEEE Access*, vol. 8, pp. 133995–134030, 2020.

[5] H. M. Al-Kadhim and H. S. Al-Raweshidy, "Energy efficient data compression in cloud based IoT," *IEEE Sensors J.*, vol. 21, no. 10, pp. 12212–12219, May 2021.

[6] Z. Ning, X. Kong, F. Xia, W. Hou, and X. Wang, "Green and sustainable cloud of things: Enabling collaborative edge computing," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 72–78, Jan. 2019.

[7] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.

[8] L. Li, X. Zhang, K. Liu, F. Jiang, and J. Peng, "An energy-aware task offloading mechanism in multiuser mobile-edge cloud computing," *Mobile Inf. Syst.*, vol. 2018, p. 12, Apr. 2018.

[9] X. Xu, "Dynamic resource allocation for load balancing in fog environment," *Wireless Commun. Mobile Comput.*, vol. 2018, p. 15, Apr. 2018.

[10] Q. Fan and N. Ansari, "Towards workload balancing in fog computing empowered IoT," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 253–262, Jan./Mar. 2018.

[11] X. Chen, Q. Shi, L. Yang, and J. Xu, "ThriftyEdge: Resource-efficient edge computing for intelligent IoT applications," *IEEE Netw.*, vol. 32, no. 1, pp. 61–65, Jan. 2018.

[12] T. Suganuma, T. Oide, S. Kitagami, K. Sugawara, and N. Shiratori, "Multiagent-based flexible edge computing architecture for IoT," *IEEE Netw.*, vol. 32, no. 1, pp. 16–23, Jan. 2018.

[13] J. Zhou, D. Tian, Z. Sheng, X. Duan, and X. Shen, "Distributed task offloading optimization with queueing dynamics in multiagent mobile-edge computing networks," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12311–12328, Aug. 2021.

[14] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via Markov decision process in mobile-edge computing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2483–2493, Feb. 2021.

[15] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008.

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[17] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[18] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018.

[19] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[20] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 1451–1455.

[21] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the Internet of Things," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Honolulu, HI, USA, Jun. 2017, pp. 17–24.

[22] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[23] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.

[24] L. Li, Q. Guan, L. Jin, and M. Guo, "Resource allocation and task offloading for heterogeneous real-time tasks with uncertain duration time in a fog queueing system," *IEEE Access*, vol. 7, pp. 9912–9925, 2019.

[25] H. Wei, H. Luo, Y. Sun, and M. S. Obaidat, "Cache-aware computation offloading in IoT systems," *IEEE Syst. J.*, vol. 14, no. 1, pp. 61–72, Mar. 2020.

[26] A. Samanta and J. Tang, "Dyme: Dynamic microservice scheduling in edge computing enabled IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6164–6174, Jul. 2020.

[27] B. Bhattacharyya and S. Raj, "Swarm intelligence based algorithms for reactive power planning with Flexible AC transmission system devices," *Int. J. Elect. Power Energy Syst.*, vol. 78, pp. 158–164, Jun. 2016.

[28] B. Bhattacharyya and S. Raj, "PSO based bio inspired algorithms for reactive power planning," *Int. J. Electr. Power Energy Syst.*, vol. 74, pp. 396–402, Jan. 2016.

[29] S. Raj and B. Bhattacharyya, "Optimal placement of TCSC and SVC for reactive power planning using Whale optimization algorithm," *Swarm Evol. Comput.*, vol. 40, pp. 131–143, Jun. 2018.

[30] S. Shekarappa, S. S. Mahapatra, and S. Raj, "Voltage constrained reactive power planning problem for reactive loading variation using hybrid Harris hawk particle swarm optimizer," *Electr. Power Compon. Syst.*, vol. 49, nos. 4–5, pp. 421–435, Sep. 2021.

[31] R. Babu, S. Raj, B. Dey, and B. Bhattacharyya, "Optimal reactive power planning using oppositional grey wolf optimization by considering bus vulnerability analysis," *Energy Convers. Econ.*, vol. 3, no. 1, pp. 38–49, Feb. 2022.

[32] B. Dey, S. Raj, S. Mahapatra, and F. P. G. Márquez, "Optimal scheduling of distributed energy resources in microgrid systems based on electricity market pricing strategies by a novel hybrid optimization technique," *Int. J. Electr. Power Energy Syst.*, vol. 134, Jan. 2022, Art. no. 107419.

[33] S. Mahapatra, B. Dey, and S. Raj, "A novel ameliorated Harris hawk optimizer for solving complex engineering optimization problems," *Int. J. Intell. Syst.*, vol. 36, no. 12, pp. 7641–7681, Aug. 2021.

[34] M. S. Shaikh, C. Hua, S. Raj, S. Kumar, M. Hassan, M. M. Ansari, and M. A. Jatoi, "Optimal parameter estimation of 1-phase and 3-phase transmission line for various bundle conductor's using modified whale optimization algorithm," *Int. J. Electr. Power Energy Syst.*, vol. 138, Jun. 2022, Art. no. 107893.

[35] L. Manman, Q. Xin, P. Goswami, A. Mukherjee, and L. Yang, "Energy-efficient dynamic clustering for IoT applications: A neural network approach," in *Proc. IEEE 8th Int. Conf. Commun. Netw. (ComNet)*, Hammamet, Tunisia, Oct. 2020, pp. 1–7.

[36] A. Mukherjee, P. Goswami, M. A. Khan, L. Manman, L. Yang, and P. Pillai, "Energy-efficient resource allocation strategy in massive IoT for industrial 6G applications," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5194–5201, Apr. 2021.

[37] L. Manman, P. Goswami, P. Mukherjee, A. Mukherjee, L. Yang, U. Ghosh, V. G. Menon, Y. Qi, and L. Nkenyereye, "Distributed artificial intelligence empowered sustainable cognitive radio sensor networks: A smart city on-demand perspective," *Sustain. Cities Soc.*, vol. 75, Dec. 2021, Art. no. 103265.

[38] X. Niu, S. Shoa, C. Xiu, J. Zhou, S. Guo, X. Chen, and F. Qi, "Workload allocation mechanism for minimum service delay in edge computing-based power Internet of Things," *IEEE Access*, vol. 7, pp. 83771–83784, 2019.

[39] S. Lyden and M. E. Haque, "A simulated annealing global maximum power point tracking approach for PV modules under partial shading conditions," *IEEE Trans. Power Electron.*, vol. 31, no. 6, pp. 4171–4181, Jun. 2016.

[40] L. Cui, J. Zhang, L. Yue, Y. Shi, H. Li, and D. Yuan, "A genetic algorithm based data replica placement strategy for scientific applications in clouds," *IEEE Trans. Services Comput.*, vol. 11, no. 4, pp. 727–739, Jul. 2018.

[41] Y. Wang and X. Zuo, "An effective cloud workflow scheduling approach combining PSO and idle time slot-aware rules," *IEEE/CAA J. Automat. Sinica*, vol. 8, no. 5, pp. 1079–1094, May 2021.

[42] S. Srichandan, T. A. Kumar, and S. Bibhudatta, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm," *Future Comput. Inf. J.*, vol. 3, no. 2, pp. 210–230, Dec. 2018.

[43] M. K. Hussein and M. H. Mousa, "Efficient task offloading for IoT-based applications in fog computing using ant colony optimization," *IEEE Access*, vol. 8, pp. 37191–37201, 2020.

[44] E. Xuefei, Z. Ma, and K. Yu, "Energy-efficient computation offloading and resource allocation in SWIPT-based MEC networks," *IEEE Access*, vol. 1, pp. 1, 2020, doi: 10.1109/ACCESS.2020.3047690.

[45] J. Bi, H. Yuan, S. Duanmu, M. Zhou, and A. Abusorrah, "Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3774–3785, Mar. 2021.

[46] A. M. Seid, G. O. Boateng, B. Mareri, G. Sun, and W. Jiang, "Multi-agent DRL for task offloading and resource allocation in multi-UAV enabled IoT edge network," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 4, pp. 4531–4547, Dec. 2021.

[47] A. Salh, L. Audah, M. A. Alhartomi, K. S. Kim, S. H. Alsamhi, F. A. Almalki, Q. Abdullah, A. Saif, and H. Algethami, "Smart packet transmission scheduling in cognitive IoT systems: DDQN based approach," *IEEE Access*, vol. 10, pp. 50023–50036, 2022.

[48] A. Salh, L. Audah, K. S. Kim, S. H. Alsamhi, M. A. Alhartomi, Q. Abdullah, F. A. Almalki, and H. Algethami, "Refiner GAN algorithmically enabled deep-RL for guaranteed traffic packets in real-time URLLC B5G communication systems," *IEEE Access*, vol. 10, pp. 50662–50676, 2022.

[49] I. Martinez, A. Jarray, and A. S. Hafid, "Scalable design and dimensioning of fog-computing infrastructure to support latency-sensitive IoT applications," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5504–5520, Jun. 2020.

[50] J. Tang, T. Lin, D. Wang, and Z. Zhou, "Optimized composition for multiple user service requests based on edge-cloud collaboration," *IEEE Access*, vol. 9, pp. 94862–94878, 2021.

[51] A. Naouri, H. Wu, N. A. Nouri, S. Dhelim, and H. Ning, "A novel framework for mobile-edge computing by optimizing task offloading," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 13065–13076, Aug. 2021.

[52] E. Kapassa, M. Touloupou, P. Stavrianos, and D. Kyriazis, "Dynamic 5G slices for IoT applications with diverse requirements," in *Proc. 5th Int. Conf. Internet Things: Syst., Manage. Secur.*, Oct. 2018, pp. 195–199.

[53] N. Wang, P. Wang, A. Alipour-Fanid, L. Jiao, and K. Zeng, "Physical-layer security of 5G wireless networks for IoT: Challenges and opportunities," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8169–8181, Oct. 2019.

[54] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, "5G evolution: A view on 5G cellular technology beyond 3GPP release 15," *IEEE Access*, vol. 7, pp. 127639–127651, 2019.

[55] S. A. Bhat, I. B. Sofi, and C.-Y. Chi, "Edge computing and its convergence with blockchain in 5G and beyond: Security, challenges, and opportunities," *IEEE Access*, vol. 8, pp. 205340–205373, 2020.

[56] M. Asad, S. Qaisar, and A. Basit, "Client-centric access device selection for heterogeneous QoS requirements in beyond 5G IoT networks," *IEEE Access*, vol. 8, pp. 219820–219836, 2020.

[57] P. Zhu, Y. Yoshida, and K.-I. Kitayama, "Ultra-Low-latency, high-fidelity analog-to-digital-compression radio-over-fiber (ADX-RoF) for MIMO fronthaul in 5G and beyond," *J. Lightw. Technol.*, vol. 39, no. 2, pp. 511–519, Jan. 15, 2021.

[58] Z. Mlika and S. Cherkaoui, "Massive IoT access with NOMA in 5G networks and beyond using online competitiveness and learning," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13624–13639, Sep. 2021.

[59] I. Al Ridhawi, M. Aloqaily, A. Boukerche, and Y. Jararweh, "Enabling intelligent IoCV services at the edge for 5G networks and beyond," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5190–5200, Aug. 2021.

[60] J. Ahmed, M. A. Razzaque, M. M. Rahman, S. A. Alqahtani, and M. M. Hassan, "A Stackelberg game-based dynamic resource allocation in edge federated 5G network," *IEEE Access*, vol. 10, pp. 10460–10471, 2022.

[61] M. Mohammadkarimi, O. A. Dobre, and M. Z. Win, "Massive uncoordinated multiple access for beyond 5G," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 2969–2986, May 2022.

[62] X. Fan, W. D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proc. Int. Symp. Comput. Archit.*, San Diego, CA, USA, vol. 35, no. 2, Jun. 2007, pp. 13–23.

[63] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency Comput., Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012.

[64] A. Ndikumana, N. H. Tran, T. M. Ho, Z. Han, W. Saad, D. Niyato, and C. S. Hong, "Joint communication, computation, caching, and control in big data multi-access edge computing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1359–1374, Jun. 2020.

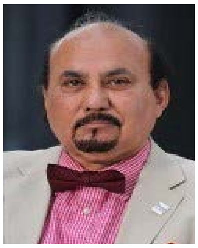
[65] R. Gai, X. Du, S. Ma, N. Chen, and S. Gao, "A summary of 5G applications and prosprcts of 5G in the Internet of Things," in *Proc. IEEE 2nd Int. Conf. Big Data, Artif. Intell. Internet Things Eng. (ICBAIE)*, Nanchang, China, Mar. 2021, pp. 858–863.

[66] M. Feng, M. Krunz, and W. Zhang, "Joint task partitioning and user association for latency minimization in mobile edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 8108–8121, Aug. 2021.

[67] M. Guo, L. Li, and Q. Guan, "Energy-efficient and delay-guaranteed workload allocation in IoT-edge-cloud computing systems," *IEEE Access*, vol. 7, pp. 78685–78697, 2019.

[68] M. S. Shaikh, C. Hua, M. A. Jatoi, M. M. Ansari, and A. A. Qader, "Application of grey wolf optimisation algorithm in parameter calculation of overhead transmission line system," *IET Sci., Meas. Technol.*, vol. 15, no. 2, pp. 218–231, Feb. 2021.

AJAY KAUSHIK received the B.Tech. and Master of Technology degrees from Kurukshetra University, India, and the Ph.D. degree from the Department of Computer Engineering, Delhi Technological University, Delhi, India. He is currently an Associate Professor at SRM University, India. He has published many journals and conference research papers in field of wireless sensor networks. His research interests include the Internet of Things, edge computing, 5G, wireless sensor networks, machine learning, neural networks, and nature-inspired intelligence. He received the Research Excellence Award for outstanding research during his Ph.D. degree. He passed the International English Language Testing System with a Band seven score, in 2021. He is an Associate Member of the British Computer Society.



HAMED S. AL-RAWESHIDY (Senior Member, IEEE) received the B.Eng. and M.Sc. degrees from the University of Technology, Baghdad, in 1977 and 1980, respectively, the Diploma (master's) degree from the University of Glasgow, Glasgow, Scotland, in 1987, and the Ph.D. degree from the University of Strathclyde, Glasgow, in 1991. He has worked with the Space and Astronomy Research Centre, Iraq, USA, Germany, and Kent University. He is currently a Professor in communications engineering. He is also the Director of the Wireless Networks and Communications Centre and the Director of Postgraduate Studies (EEE) at Brunel University London, U.K.