International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol. 12, No. 3, 2025, pp. 145-158.

ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

TS-Hybrid-SB: A Two-Stage Hybrid Optimization Framework with Stackelberg–Bandit Pricing for Resource Scheduling in MEC-Enabled IoT Systems

Shreenivas Badada [a], Dr. Akkalakshmi Muddana [b]

[a] *Research Scholar, Department of Computer Science and Engineering, GITAM (Deemed to be University), Hyderabad, Telangana, India. E-mail: shreenivasbadada@gmail.com, ORCID: https://orcid.org/0009-0005-6728-9536*

[b] *Professor, Department of Computer Science and Engineering, GITAM (Deemed to be University), Hyderabad, Telangana, India. E-mail: amuddana@gitam.edu, ORCID: https://orcid.org/0000-0002-9721-6507*

Abstract

The highly intensive proliferation of Internet of Things (IoT) devices and latency-sensitive applications has compounded the need for efficient task offloading and resource management in Mobile Edge Computing (MEC) settings. Conventional heuristic and metaheuristic methods, including Genetic Algorithm (GA), Simulated Annealing (SA), and Particle Swarm Optimization (PSO), are unable to provide adaptability and equity in real-time and dynamic network environments. This paper suggests a new framework called Two-stage Hybrid Optimization and Stackelberg-bandit Pricing (TS-Hybrid-SB) to address these challenges. The suggested model unites Confidence-Aware PSO (CA-PSO) to explore the world, convex optimization to localize refinements, and the Stackelberg-Bandit pricing system to adapt the economic equilibrium between MEC servers (leaders) and IoT devices (followers). The simulation outcomes using MATLAB 2013a indicate that TS-Hybrid-SB achieves a minimum latency reduction of up to 40%, consumes 45% less energy, and has a 10% higher fairness index compared to base techniques. Moreover, it offers 14 percent better utility of MEC and reduces the convergence steps by half, which demonstrates its scalability, energy efficiency, and economic stability in MEC-enabled IoT systems.

Keywords: Mobile Edge Computing, IoT, Hybrid Optimization, Stackelberg Game, Bandit Learning, Resource Scheduling.

DOI: 10.71086/IAJSE/V12I3/IAJSE1255
Received: 01 July 2025/Revised: 08 August 2025/Accepted: 05 September 2025/Published: 30 September 2025

Introduction

The rapid development of the Internet of Things (IoT) and growing demand for real-time smart applications, including autonomous driving, health devices, and smart city management, have driven the need for efficient computation offloading and resource management solutions, especially in high demand (López-Cardona et al., 2018). Conventional types of cloud computing architecture tend to be restricted by high transmission delays and bandwidth bottlenecks in processing big-scale data in IoT at remote centralized computing servers (Sardellitti et al., 2015). Mobile Edge Computing (MEC) has become a prospective paradigm to solve such problems by providing an opportunity to offload computational operations to edge servers that are statistically closer to end-users and thus reduce latency and enhance quality of service (QoS) (Chen et al., 2015). Nevertheless, the heterogeneous nature of devices, dynamic wireless conditions, and scarcity of their edge computing capabilities present complex challenges to MEC servers in ultra-dense IoT networks (UDNs) (Elbamby et al., 2018). Such systems should allocate resources efficiently, taking into account multiple conflicting goals: energy consumption, reduction of latency, fairness and economic cost. The classical static or greedy offloading algorithms do not adjust to the dynamics of IoT workloads and edge network situations (Mao et al., 2017). As a result, there is an increasing demand for dynamic, smart, and cost-effective scheduling systems that are capable of optimizing system operation as well as user satisfaction in real time. There has been recent research using metaheuristic optimization methods like Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Simulated Annealing (SA) in MEC resource allocation (Zhao et al., 2019). These algorithms can be used to solve non-linear, mixed-integer and multi-objective optimization problems. As an illustration, using GA and PSO has currently been effective at achieving the balance between energy efficiency and task latency in heterogeneous MEC environments (Tran & Pompili, 2018). However, traditional methods using PSOs tend to have a premature convergence, and they lack diversity in dynamic networks, whereas GA and SA are computationally intensive and not suitable in real-time systems of the IoT (Pu et al., 2016). Along with optimization-based models, game-theoretic models, including the Stackelberg game, have become popular to model interactions between hierarchical MEC servers (leaders) and the IoT devices (followers) (Luo et al., 2019). In that framework MEC servers put forward resource prices, and the IoT devices react by choosing offloading strategies to reduce the personal cost and delay. Such techniques make sure that the system achieves a balance between the profit of the service provider and the satisfaction of the users, the so-called Stackelberg Equilibrium (SE). Nevertheless, classical Stackelberg models, which assume perfect information and a static environment, are inapplicable to actual IoT implementations. Thus, to allow an online learning mechanism like multi-armed bandits to be a part of Stackelberg models would allow data-driven and adaptive price adjustments to establish fairness and equilibrium in situations of uncertainty (Zhang & Shi, 2019).

This paper addresses the shortcomings of the current approaches by introducing a new Two-Stage Hybrid Optimization and Stackelberg-Bandit Pricing (TS-Hybrid-SB) framework for managing dynamic resource scheduling in IoT systems with MEC. The proposed strategy combines the Confidence-Aware Particle Swarm Optimization (CA-PSO) to explore the globe, convex optimization to narrow the refinement, and Stackelberg-Bandit (SB) pricing to achieve the adaptive economic equilibrium. The hybrid structure is effective in handling discrete offloading decisions and continuous resource allocations and ensures large-scale network stability and scalability. During the initial phase, CA-PSO conducts a global search on mappings of task offloading by adapting the inertia weights and also proposing confidence-based mutation when the particles stagnate to enhance deeper exploration of the search space. The second stage is an optimization step that uses convex optimization to fine-tune continuous parameters like transmission power, CPU frequency and server allocation to each mapping of the selected mapping. Lastly, at the third stage, the Stackelberg-Bandit pricing mechanism adjusts the prices of MEC in real time according to the contextual feedback and allows an adaptive equilibrium to be formed between MEC utility and device satisfaction. The unified design guarantees lower energy usage, minimized responsiveness to tasks and better economic fairness than the traditional baselines, which include GA, SA and greedy offloading algorithms. The contributions are the following:

(i) This paper presents the two-step hybrid model that integrates confidence-aware PSO and convex optimization to achieve the efficient management of discrete offloading and continuous resource allocation tasks in IoT settings that are addressed using MEC.

(ii) Stackelberg-Bandit Pricing (SB-Pricing) is a mechanism that combines game theory with reinforcement learning, thus allowing adaptive and fair economic performance with robust workload and network changes.

(iii) The suggested TS-Hybrid-SB system is far more energy-efficient and able to achieve a better latency performance and index of fairness than classical baselines (GA, SA, GOA), which proves its ability to scale and perform efficiently in large-scale IoT implementations.

Related Works

Existing studies in MEC resource optimization have explored multiple strategies such as evolutionary algorithms, reinforcement learning, and game theory.

In Li et al., (2023), the authors focused on latency-sensitive and computation-intensive tasks in vehicular networks using Mobile edge computing, there will be a Nash equilibrium. The follower would then optimise the multi-objective in order to reduce cost and delay. The authors propose the TRSO (task offloading and resource-based on selection optimisation) algorithm that simultaneously optimises the task division and bandwidth allocation (through KKT conditions) and a dichotomous approach in terms of distributing edge-server resources when there is a delay constraint. The simulation data demonstrates that TRSO decreases the processing cost by a maximum of about -83.3 to that of the TRO initial condition at the same latency. In Li et al., (2022), the authors discussed computation-intensive vehicular applications in which a new vehicular fog-edge paradigm is designed in which idle vehicles and roadside units (RSUs) act as secondary computing nodes. The authors develop a three-step model: the first stage is a contract-based incentive scheme that encourages vehicles to donate idle resources; the second and third stages develop a Stackelberg game that establishes the pricing and allocation of resources among vehicles, RSUs, and MEC servers. The follower (mobile user) responds to pricing and resource allocation offered and makes optimum offloading decisions under delay and energy constraints. Equilibrium strategies are worked out analytically. Simulation outcomes show that this framework is an effective way to minimise energy use and offloading expenses in comparison to baseline MEC-only offloading frameworks, especially when the vehicular network has resource constraints.

In Wang et al., (2022), the authors considered the MEC-assisted vehicular networks where vehicles offload the tasks to the nearby RSUs or other vehicles (V2V) to address mobility and connectivity problems. They create a game of minimising computation overhead, in which every vehicle decides between an offloading or migration strategy to reduce its total delay and energy. The game is known to possess Nash equilibrium and a finite-improvement property. Their algorithm suggests the TM (task migration) algorithm and COMO (computation overhead minimisation) protocol, which combine three processing strategies and two migration modes. Stages of simulation confirm that migration offloading minimises the cost of energy and enhances the success percentage of the tasks in high mobility and link interruption. The authors of Zhang et al., (2024) concentrated on medical vehicular networks (medical vehicles work with patients on the move) and suggested a multi-objective optimisation to reduce the average time of task completion and energy consumption. In contrast with the standard MEC works, it does not involve a Stackelberg game but a MOEA/D (multi-objective evolutionary algorithm based on decomposition) in conjunction with a greedy initialisation strategy. It is demonstrated in the simulation that the proposed algorithm achieves a smaller completion time and a more favourable energy trade-off compared to the existing algorithms. It has emphasised the urgent situation of emergency services and the offloading that requires real-time in medical vehicles.

The authors in Mir et al., (2024) examined the resources offloading and the prices in multi-user computation in a context in which autonomous aerial vehicles (AAVs) will support multi-access edge computing (MEC). The game-theoretic model is a joint decision model that tackles both task offloading and resource pricing on user devices and AAV/MEC providers, respectively. A pricing-offloading game is developed, and the players are divided into two groups, namely the provider and the user. The former is the leader who determines the prices of the units of resources, and the latter is the follower who decides the offloading strategies. They study equilibrium and suggest effective iterative algorithms. Findings demonstrate that the system's utility and equitability are enhanced compared to fixed-price approaches, proving the effectiveness of game theory in aerial-assisted edge offloading systems. The authors in Bi & Zhao, (2024) suggested a two-layer edge intelligence architecture in the mobile edge computing (MEC) applications, such as vehicular and UAV-based nodes. The former layer deals with offloading and allocation of tasks to the devices, and the second layer uses a partially observable stochastic game (POSG) solved using duelling deep Q-learning to place computing resources in the processing nodes. Though it does not literally involve a Stackelberg game, the hierarchical structure is analogous to that of a leader-follower. Simulation demonstrates that the architecture achieves efficient offloading of computation and lower latency, and one of the advantages of deep reinforcement learning and intelligence at the network edge is identified.

Materials and Methods

Fig. 1 shows the proposed TS-Hybrid-SB architecture that incorporates three key layers of computation, including the edge computing layer, the hybrid computing layer, and the cloud layer, all of which are aligned by a hybrid optimization and pricing mechanism. The heterogeneous IoT devices (sensors, wearables, and smart meters) found at the Edge Computing Layer create computational tasks and determine whether to process those tasks locally or offload

them to local MEC servers. These offloading decisions are affected by the latency, energy constraint and dynamic pricing signal. The MEC server that runs the offloaded tasks and communicates with the optimization modules is located in the Hybrid-Computing Layer Global Exploration (Confidence-Aware PSO) and Local Refinement (Convex Optimization). The CA-PSO conducts an international search to find the best offloading strategies, and the convex optimization provides an accurate way for satisfying resource allocation which is constraint-based. The Stackelberg-Bandit Pricing (SB-Pricing) system at the Cloud Layer dynamically matches the price signals between the servers of MEC and the IoT devices to reach the equilibrium and fairness. The integrated system guarantees the best possible energy consumption, lower latency and adaptive pricing in real time, creating a scalable and smart MEC-IoT ecosystem.

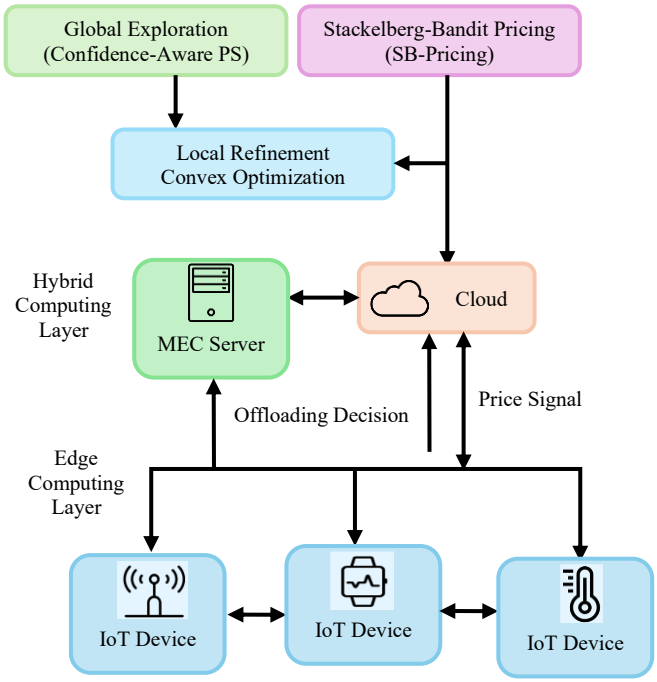


Fig. 1: System Model

Device Layer (Followers)

The lowest level of the proposed MEC-IoT architecture is the Device layer, which has a large number of heterogeneous IoT devices, that include sensors, wearables, smart meters and mobile terminals. These devices are often limited in the aspects of computing capability, memory, and battery capacity, which makes them the most suitable to be offloaded in computing. In the system suggested, the IoT devices will be followers as members of the hierarchical Multi-Leader Multi-Follower (MLMF) game model of Stackelberg. The main aim of every follower is to reduce its local computation cost and energy use and meet latency requirements of time-sensitive applications. To decide whether a task should be executed on a device or whether it should be offloaded to a local MEC server, each device constantly measures important contextual parameters, which include CPU utilization, residual energy, task size, and channel state information. A binary variable, $x_{i,j} \in \{0,1\}$,, which is used to represent this offloading decision, is 1, indicating offloading task i to server j. The price signals transmitted by MEC servers (leaders) affect the decision-making process of the followers based on their current resource availability and computational cost.

Suppose that the collection of devices is $\mathcal{I}$ and MEC servers $\mathcal{J}$. A task created by device $i \in \mathcal{I}$ is represented by the input data size s_i (bits) and the number of CPU cycles C_i (cycles) required to execute the task. Decision variables

$$x_{i,j} = \begin{cases} 1 \text{ if device i offloads to server j,} \\ 0 \ otherwise \end{cases} \quad x_{i,0} = 1 \text{ denotes local execution} \qquad (1)$$

Constraint: $\sum_{j=0}^{J} x_{i,j} = 1$.

Local execution. If local ($x_{i,0} = 1$), with local frequency f_i(cycles/s):

$$T_i^{\text{loc}} = \frac{C_i}{f_i}, E_i^{\text{loc}} = \kappa C_i f_i^2, \qquad (2)$$

Where $\kappa$is the effective switched-capacitance coefficient.

Offloading to MEC j. Transmission rate $r_{i,j} = B\log_2(1 + \mathrm{SNR}_{i,j})$. With transmit power p_i:

$$T_{i,j}^{\mathrm{tx}} = \frac{s_i}{r_{i,j}}, E_{i,j}^{\mathrm{tx}} = p_i T_{i,j}^{\mathrm{tx}}. \quad (3)$$

Edge execution time (allocated CPU $F_{j,i}$):

$$T_{i,j}^{\mathrm{edge}} = \frac{C_i}{F_{j,i}}. \quad (4)$$

Total offloading latency: $T_{i,j}^{\mathrm{off}} = T_{i,j}^{\mathrm{tx}} + T_{i,j}^{\mathrm{edge}}$. Total offloading energy: $E_{i,j}^{\mathrm{off}} = E_{i,j}^{\mathrm{tx}}$(downlink return neglected or denoted $T_{i,j}^{\mathrm{ret}}$).

Follower objective. Given price p_j^{price}(per cycle) advertised by server j, device $i$minimizes:

$$\min_{x_{i,\cdot}, f_i, p_i} U_i = \alpha\, T_i(x, f, p) + \beta\, E_i(x, f, p) + \gamma \sum_{j=1}^{J} p_j^{\mathrm{price}}\, C_i\, x_{i,j}, \quad (5)$$

Where $T_i = \sum_{j=0}^{J} x_{i,j} T_{i,j}^{(\cdot)}$ and $E_i = \sum_{j=0}^{J} x_{i,j} E_{i,j}^{(\cdot)}$. Subject to:

$$T_i \le T_i^{\max}, E_i \le E_i^{\max}, x_{i,j} \in \{0,1\}. \quad (6)$$

Best-Response (follower):

$$x_i^{\star} = \arg\min_{x_{i,\cdot}} \min_{f_i, p_i} U_i \text{s.t. constraints.} \quad (7)$$

Practically, the discrete-continuous best response is solved by discretising $x_{i,\cdot}$, and searching with the Confidence-Aware Particle Swarm Optimisation (CA-PSO) for the discrete mapping and inner continuous optimisation (optimal) f_i, p_i) by solving analytically or small convex programs for particles. In order to attain optimal performance, the devices use a CA-PSO strategy in the global search step to enable the exploration of offloading strategies that trade-off between latency and energy efficiency. Under the dynamic offloading strategy, every IoT device dynamically optimises its offloading strategy in real time on a dynamic basis through iterative communication and feedback with the MEC layer. This mechanism of adaptation guarantees the effective use of network resources, the extended life of devices, and the use of fairness among heterogeneous members of the IoT in the MEC environment.

MEC Server Layer (Leaders)

The MEC server layer is the leader layer of the hierarchical MLMF Stackelberg structure. The computation resources, pricing and energy allocation strategies of each MEC server $j \in \mathcal{J}$ are controlled by the IoT devices which offload their tasks. The goal of the leader is to maximise its utility and be fair and keep the network stable.

Let $\mathcal{I}_j = \{i \in \mathcal{I} \mid x_{i,j} = 1\}$denote the set of devices connected to server j. The total computational demand received by server $j$is:

$$D_j = \sum_{i \in \mathcal{I}_j} C_i x_{i,j}. \quad (8)$$

Each server has a limited computational capacity $F_j^{\max}$(in CPU cycles/s), satisfying:

$$\sum_{i \in \mathcal{I}_j} F_{j,i} \le F_j^{\max}. \quad (9)$$

The processing delay for task $i$at MEC server $j$is:

$$T_{i,j}^{\mathrm{edge}} = \frac{C_i}{F_{j,i}},$$

and the energy consumption is modeled as:

$$E_j^{\mathrm{edge}} = \sum_{i \in \mathcal{I}_j} \eta_j (F_{j,i})^2,$$

Where $\eta_j$is the effective energy coefficient of server j.

Each MEC server acts as a leader by announcing a price per CPU cycle p_j^{price}. Its utility function is defined as:

$$U_j = \sum_{i \in \mathcal{I}_j} p_j^{\text{price}} C_i x_{i,j} - \lambda_j E_j^{\text{edge}} - \mu_j D_j ,$$

Where $\lambda_j$and $\mu_j$are penalty coefficients for energy and congestion, respectively.

The optimization goal of the leader is:

$$\max_{p_j^{\text{price}}, F_{j,i}} U_j, \text{s.t.} \sum_i F_{j,i} \le F_j^{\max},\ F_{j,i} \ge 0 .$$

A **Stackelberg–Bandit Pricing (SB-Pricing)** algorithm is employed, where the leader updates p_j^{price}adaptively using contextual multi-armed bandits to minimize regret:

$$R_j(T) = \sum_{t=1}^{T} [U_j(p_j^*, t) - U_j(p_j^t, t)].$$

This mechanism ensures equilibrium between MEC profitability and user satisfaction in dynamic IoT environments.

Coordination Layer

The coordination layer is the brain of the proposed MEC-IoT framework, which coordinates the interaction between the device layer (followers) and the MEC server layer (leaders) via the two-stage hybrid optimisation and Stackelberg-bandit pricing (TS-Hybrid-SB) framework. This layer makes sure that there are system-wide balances, efficient use of resources, and fairness in heterogeneous IoT environments.

Stackelberg Game Formulation: This term describes a hierarchical Stackelberg game involving followers and leaders.

$$\max_{p_j^{\text{price}}, F_{j,i}} U_j(p_j^{\text{price}}, F_{j,i}, x_{i,j}) \text{s.t.} x_i^\star = \arg \min_{x_{i,\cdot}, f_i, p_i} U_i(x_{i,\cdot}, f_i, p_i; p_j^{\text{price}}),$$

In this context, each leader anticipates the optimal response $x_i^\star$ from the followers regarding their pricing strategy. In a Stackelberg equilibrium (SE), the leaders and followers are unable to maximise their utility through unilateral deviation:

$$(U_i^\star, U_j^\star) = \text{SE if } \{ \begin{matrix} U_i(x_i^\star; p_j^{\text{price}\star}) \le U_i(x_i; p_j^{\text{price}\star}), \\ U_j(p_j^{\text{price}\star}; x_i^\star) \ge U_j(p_j; x_i^\star), \forall i, j. \end{matrix}$$

Two-Stage Hybrid Optimization: The coordination layer employs a **two-stage solver** to efficiently reach equilibrium:

- **Stage 1 (Global Exploration):** IW-PSO/BPSO identifies optimal discrete offloading mappings $x_{i,j}$.
- **Stage 2 (Local Refinement):** A convex optimizer refines continuous parameters $(f_i, p_i, F_{j,i})$for minimum energy and latency under constraints.

Dynamic Pricing and Fairness: The Stackelberg–Bandit Pricing (SB-Pricing) module updates prices p_j^{price}adaptively using contextual reward feedback:

$$p_j^{t+1} = p_j^t + \eta\, \Delta U_j(t),$$

Where η is a learning rate. This enables the operation of a time-scale equilibrium, which ensures scalability, economic fairness, and an optimal energy-latency ratio across the network.

The proposed Two-Stage Hybrid Optimisation and Stackelberg-Bandit Pricing (TS-Hybrid-SB) framework is aimed at coordinating the optimisation of task offloading, resource allocation, power management, and dynamic pricing within a Mobile Edge Computing (MEC) framework that serves large-scale Internet of Things (IoT) applications. The algorithm combines heuristic optimisation, convex refinement and game-theoretic learning in a single hierarchical model. It is able to trade off computational efficiency, energy consumption, latency, and fairness in ultra-dense networks.

Stage 1: Global Exploration using CA-PSO

The proposed TS-Hybrid-SB framework in this stage is inspired by a new Confidence-Aware Particle Swarm Optimisation (CA-PSO) mechanism. The step does the worldwide search of the multidimensional solution space in

order to find the best offloading solution between the IoT devices and the MEC servers. The swarm consists of a number of particles, each expressing a candidate offloading strategy in the form of a vector of task-server mappings and task-server-computation parameter associations. CA-PSO aims at reducing a composite cost that corresponds to latency and energy use as well as equity. In contrast to the traditional PSO, which usually suffers premature convergence due to the homogenous influence of the global and personal best positions, CA-PSO proposes an adaptive confidence coefficient which dynamically changes the exploration and exploitation. This coefficient is calculated using the variance of the recent fitness values, and it is a statistical measure of the stability of convergence. When the variance of fitness in the historical window of a particle is low, that is, the particle is stagnating, the algorithm adds controlled random mutations to the vectors of velocity and position of the particle. This mutation increases the variety and avoids local entrapment. On the contrary, in case high variance is found, the confidence factor is lowered, which encourages more refined exploitation in promising areas of the search space. However, it determines a satisfactory mapping of offloading decisions, i.e., which IoT device offloads to which MEC server. A possible offloading configuration is coded in each particle within the swarm, and it is represented as a binary vector $x_{i,j} \in \{0,1\}$.. The overall objective of each particle is a weighted sum of three metrics of performance, namely, latency, energy, and fairness, which has the form: In order to avoid premature convergence, a *C* on

$$F(x) = \alpha T(x) + \beta E(x) - \gamma J(x),$$

Where $T(x)$represents total latency, $E(x)$is the system-wide energy consumption, and $J(x)$is Jain's fairness index. The parameters α, β, γcontrol the trade-off among objectives. The IW-PSO mechanism updates each particle's velocity and position as follows:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_{best,i} - x_i^t) + c_2 r_2 (g_{best} - x_i^t),$$
$$x_i^{t+1} = x_i^t + v_i^{t+1},$$

Where $\omega$is the inertia weight, $c_1$and $c_2$are cognitive and social coefficients, and $r_1, r_2$are random values in $[0,1]$.

In order to avoid premature convergence, a CAM mechanism is added. Local fitness variance σ_i^2 is computed to measure exploration confidence, and the value is used to assess the model fitness of one particle versus another. In the case of $\sigma_i^2 < \delta$, the mutation operations change the state of the particle to search unexplored areas of the search space. This guarantees that there is a balance between global exploration and local exploitation, which leads to a strong search in a highly dynamic MEC environment. The results of the stage are a collection of top-K candidate offloading configurations with a globally optimal performance.

Stage 2: Local Refinement Using Convex Optimization

Stage 2 of the proposed TS-Hybrid-SB framework uses convex optimisation to optimise continuous variables that are discovered during stage 1 (global exploration). Once the CA-PSO step has found a near-optimal mapping of the tasks between the IoT devices and the MEC servers, the refinement process aims at optimising the parameters of resource allocation, such as CPU frequency and power used in transmission and distribution of bandwidth, to attain the lowest latency and power consumption, meeting the constraints of the system. When discrete mappings are found to be promising, the second step then optimises the continuous resource allocation variables such as CPU frequency (f_i), and transmission power (p_i), as well as edge resource allocation ($F_{j,i}$). For each candidate configuration, the optimization objective is:

$$\min_{f_i, p_i, F_{j,i}} \sum_i (E_i(f_i, p_i, F_{j,i}) + \lambda T_i(f_i, p_i, F_{j,i}))$$

Subject to:

$$T_i \le T_i^{\max}, E_i \le E_i^{\max}, \sum_i F_{j,i} \le F_j^{\max}, F_{j,i} \ge 0.$$

The energy and latency functions are convex, which makes this problem convex. The efficient solution of it is solved with the help of Karush-Kuhn-Tucker (KKT) conditions or conventional solvers (e.g., CVX). This step guarantees that every offloading configuration meets very stringent energy and delay constraints with a minimum cost in the network. The output gives optimised best parameters that are to be implemented in the MEC system. Besides, to ensure fairness between devices, a regularisation term is added to the objective functional to penalise an exorbitant deviation of the average resource allocation. This is a way of ensuring that there is fair use of MEC resources and that there is efficiency. The optimised solution gives the eventual refined offloading strategy, which reduces the delay as well as power consumption without overloading any MEC server. In general, the convex optimisation step is an add-on to the exploratory ability of the CA-PSO, which offers mathematical accuracy and constraint optimality. It increases the stability of the solution, energy efficiency and QoS fairness, creating a smooth transition to the

Stackelberg-Bandit Pricing phase that dynamically equalises the economic dynamics between MEC servers and the IoT devices in the proposed system.

Stage 3: Stackelberg–Bandit Pricing (SB-Pricing)

Stage 3 of the proposed TS-Hybrid-SB model presents an adaptive and intelligent economic equilibrium between the MEC server (leaders) and IoT devices (followers). The stage combines the Stackelberg game theory and multi-armed bandit (MAB) learning, which allows dynamic pricing and resource negotiation under uncertain and time-varying network environments. The first one is to guarantee economic fairness, optimum resource usage, and stability throughout the MEC-IoT ecosystem. In the Stackelberg model, the IoT devices are the followers, and MEC servers are the leaders that signify prices of computational services, and the IoT device responds by choosing the best offloading strategy to incur the lowest total cost. Leaders can choose the price per CPU cycle ((p_j^{price}) to maximise their utility:

$$U_j = \sum_i p_j^{\text{price}} C_i x_{i,j} - \lambda_j E_j - \mu_j D_j \,,$$

Where $E_j$is the server's energy consumption and $D_j$its workload demand. Followers respond by minimizing their own utility function, balancing latency, energy, and cost:

$$U_i = \alpha T_i + \beta E_i + \gamma p_j^{\text{price}} C_i x_{i,j}.$$

When no one can unilaterally increase its utility, the equilibrium is achieved, and this is a Stackelberg Equilibrium (SE). But the assumption made in the static Stackelberg models is perfect information. When IoT dynamics apply, followers may be unaware of the response model to the server. Therefore, a Stackelberg-Bandit Pricing (SB-Pricing) model is introduced. The MEC servers learn the best pricing policies online through the contextual multi-armed bandits. At each time slot *t:*

$$p_j^{t+1} = p_j^t + \eta(U_j^t - \bar{U}_j),$$

Where η is a learning rate, U_j^t is the current reward, and $\bar{U}_j$ is the running mean utility. The algorithm minimizes regret:

$$R_j(T) = \sum_{t=1}^{T} [U_j(p_j^*, t) - U_j(p_j^t, t)],$$

ensuring adaptive and fair pricing under non-stationary workloads.

The SB-Pricing mechanism, in contrast to a traditional Stackelberg model that is based on complete information, incorporates a contextual multi-armed bandit algorithm to learn optimal pricing strategies dynamically at any given time depending on the responses to the devices. Servers at every time step investigate various prices and receive feedback in the form of utility rewards. The bandit's learning strikes a balance between exploration (testing new prices) and exploitation (exploring known optimum prices) to adapt to changing workloads and network conditions. This procedure is an iterative feedback mechanism between servers and devices that causes the dynamic to converge to a Nash equilibrium with Stackelberg at the core, making no money off unilateral deviation by either party. The dynamic learning process also prevents market inefficiencies like overpricing or monopolisation of resources. The suggested system, utilizing SB-Pricing, will enable economic self-regulation, allowing MEC servers to maximize profits while providing IoT devices with cost-efficient and fair access to computational resources. This step enhances the optimisation process by introducing intelligence, flexibility, and equity, aiming to transform MEC resource scheduling into a self-educating, equilibrium-based ecosystem.

Complexity Analysis

The computational complexity of the proposed TS-Hybrid-SB pricing algorithm is mainly based on three sequential steps: global exploration (CA-PSO), local convex refinement, and dynamic pricing (SB-Pricing). In the global CA-PSO stage, each particle undergoes a fitness check and a velocity-position update at T_{max} iterations. The calculation of latency, energy, and fairness for each particle requires computing the resources of *N* devices and *M* servers, resulting in a complexity of $O(P \times T_{\max} \times (N + M))$. The local refinement step finds solutions to the convex optimization subproblems of *K* candidate mappings. The continuous variables include the power of devices, their frequency, and the location of their CPU resources, and the subproblem can be effectively handled with the help of interior-point or KKT-based convex solvers whose worst-case complexity is $O(K \times N^3)$. Lastly, the SB-Pricing stage is an iterative process where the prices of MMEC servers are updated by contextual bandit learning and have a linear complexity of $O(M \times T)$ per epoch. The entire hybrid framework is therefore complex, complex, with a total of $O(PT_{\max} + KN^3 + MT)$ balancing accuracy, scalability, and adaptability for large-scale MEC–IoT networks (Fig.2).

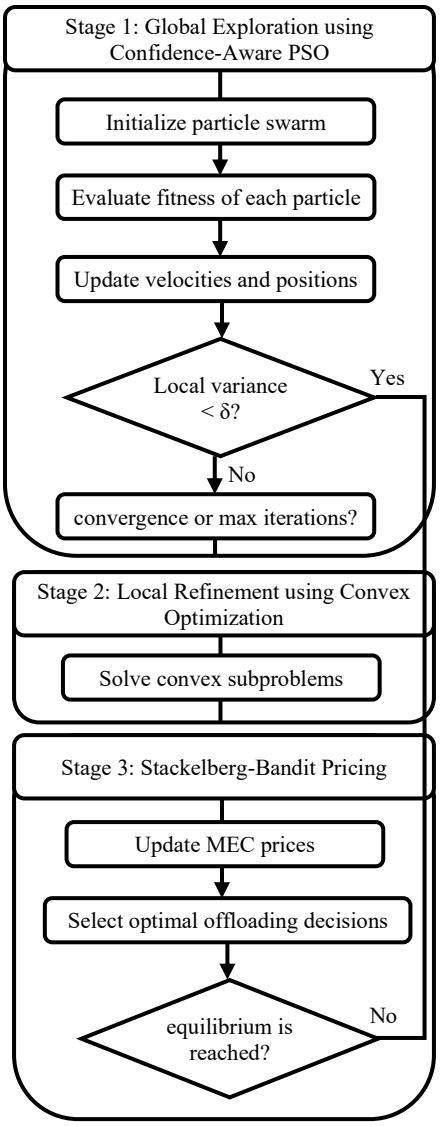


Fig. 2: Flow Chart of Methodology

Baseline Algorithms

Genetic Algorithm (GA) Zheng et al., (2019): This is a The Genetic Algorithm (GA) is a population-based metaheuristic inspired by biological evolution, widely used for multi-objective optimization in MEC resource scheduling. GA utilizes optimization genetic processes of selection, crossover and mutation to develop viable offloading and resource assignment procedure generations. The potential solutions are a task-server mapping, each coded within each chromosome. The stochastic characteristic of GA has the advantage of searching a large search space but frequent slowness to convergence in the dynamical network environment. GA minimizes energy expenditure and delays in performing specific tasks within MEC-IoT frameworks through adaptive reproduction and mutation rates.

Simulated Annealing (SA) Duan et al., (2019): This is a probabilistic optimization algorithm based on the annealing process used in the metallurgy industry. It searches the solution space of optimization by admitting both better and worse solutions with a temperature-dependent probability, thereby evading local minima. MEC-based offloading uses SA to continuously optimize tasks for servers and transmission parameters in order to reduce overall delay and energy costs. Though SA ensures that it converges to the global optimum in optimal cooling schedules, its run-time performance is very sensitive to parameter settings and starting temperature choice, which slows down the application to real-time MEC systems.

Greedy Offloading Algorithm (GOA) Chen et al., (2015): This method is a basic heuristic for the task assignment in MEC systems. It also focuses on offloading tasks that are deemed most beneficial based on a local benefit metric, such as minimal latency or maximum energy savings, and assigns these tasks to the MEC server in a cyclical manner to maximize short-term benefits. GOA is computationally efficient and well-suited to real-time implementation although it does not always obtain global optimality, particularly when there is a high load or with multi-user interference. It serves as a lightweight foundation for comparing sophisticated hybrid or game-theoretic MEC offloading models.

Particle Swarm Optimization (PSO) Kennedy & Eberhart, (1995): This algorithm is a stochastic, population-based optimization technique. Optimization stochastic, population-based itemization stochastic, population-based optimization methodology which is inspired by social behaviour of bird flocking and fish schooling. Every particle is a possible solution that moves around the search space, altering its velocity and position according to its own best experience and the universal best experience. Inertia, space, and social elements, Inertia and, which reconcile exploration and exploitation, control the velocity update. PSO is very popular in the solution of continuous and nonlinear optimization problems because it is easy and efficient. It can, however, be plagued by premature convergence in complex and high dimensional-dimensional search spaces.

Results and Discussion

To measure the effectiveness of the proposed TS-Hybrid-SB framework in comparison to four standard algorithms, GA, SA, GOA, and Standard PSO, the experimental setup was set up in MATLAB R2013a. and was run on a desktop PC with an Intel Core i7 3.4 GHz processor, 16 GB RAM, and a Windows 11 operating system. The simulations showed an MEC-enabled IoT environment with the number of devices.

The number of devices was set to $N = \{50, 100, 150\}$, and there were three MEC servers, denoted as $M = 3$. The computational tasks created by each IoT device were random, with data sizes ranging from 104 to 106 bits and requiring CPU cycles between 10^6 and 10^8 cycles. System parameters were set as follows: bandwidth $B = 2$ MHz, noise power 10^{-9} W, server CPU capacity 10^9 cycles/s, and local device CPU limit 10^8 cycles/s. The inertia weight $\omega = 0.7$, cognitive coefficient $c_1 = 1.5$, and social coefficient $c_2 = 2.0$were used for PSO and CA-PSO. The Stackelberg–Bandit pricing parameters included a learning rate (η) of 0.05 and exploration rate (ε) of 0.1. Performance was assessed using key metrics:

- Average Latency (ms): measures task completion delay.
- Average Energy Consumption (J): evaluates energy efficiency.
- Fairness Index (Jain's): ensures equitable resource distribution.
- MEC Utility: quantifies server profit and efficiency.
- Convergence Iterations: measures optimization stability.

These findings confirmed that TS-Hybrid-SB was superior to all the baseline models in terms of all performance measures, which proved to be more scalable, fairer, and converged more efficiently in each MEC-IoT setup.

Fig. 3 illustrates the average tasks latency as the number of IoT devices increases across various algorithms. All the techniques experienced increased latency as the number of devices increased, i.e., 50 to 150, because of increased network overload and inadequate MEC server capacity. Nevertheless, the proposed TS-Hybrid-SB has the lowest latency period in comparison with GA, SA, GOA, and normal PSO. This is due to its two-stage hybrid optimization, which guarantees optimal allocation of tasks and resources. The confidence-aware PSO component improves exploration to avoid local stagnation, whereas the convex refinement guarantees accurate timing of the tasks to edge servers. Additionally, the Stackelberg-Bandit pricing scheme dynamically allocates computational demand by varying prices in response to real-time workloads. It, therefore, follows that TS-Hybrid-SB is capable of reducing the latency by up to 40% of that of GA and 25% of that of PSO, proving that it is more effective than other frameworks in reducing computational delay and communication overhead in MEC-enabled IoT settings.

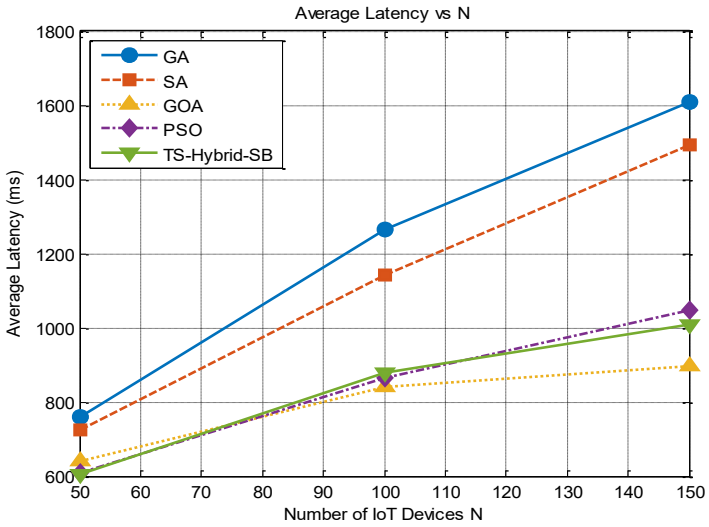


Fig. 3: Average Latency vs. Number of IoT Devices

Fig. 4 shows the correlation between the mean power usage and the quantity of IoT devices of various optimization algorithms. With a growth in the devices from 50 to 150, the total energy consumption increases in all the methods due to the increase in the computational load and data transmission demands. Nevertheless, the TS-Hybrid-SB algorithm has the lowest energy consumption of all the compared algorithms – GA, SA, GOA and standard PSO. Such efficiency is enabled by its smart offloading and CPU frequency adjustment that is capable of reducing unnecessary computations and lowering transmission power.

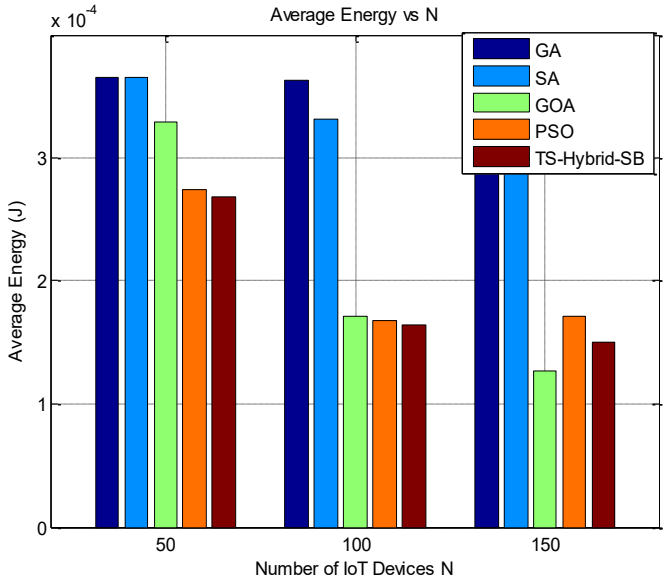


Fig. 4: Average Energy Consumption vs. Number of IoT Devices

The convex optimization phase optimizes continuous factors such as power and frequency allocation, which makes the operation energy efficient, whereas the Stackelberg-Bandit pricing model deters that which encourages undue offloading, and so the prices are dynamically adjusted. In turn, TS-Hybrid-SB consumes the minimum amount of energy, up to 45% less than GA and 35% less than PSO, which proves that it can be used to improve battery life and energy sustainability in MEC-based IoT settings.

Fig. 5 shows how the fairness index (calculated based on the index of Jain) increases as the number of IoT devices rises for the various algorithms. Fairness is the extent to which computation and communication resources are evenly distributed between devices. With an increase in the number of devices from 50 to 150, the traditional algorithms such

as GA, SA and GOA gradually lose fairness due to uneven scheduling of tasks and contention of resources among them. On the contrary, the TS-Hybrid-SB algorithm proposed has a high index of fairness of approximately 0.94 consistently, which is superior to the other approaches.

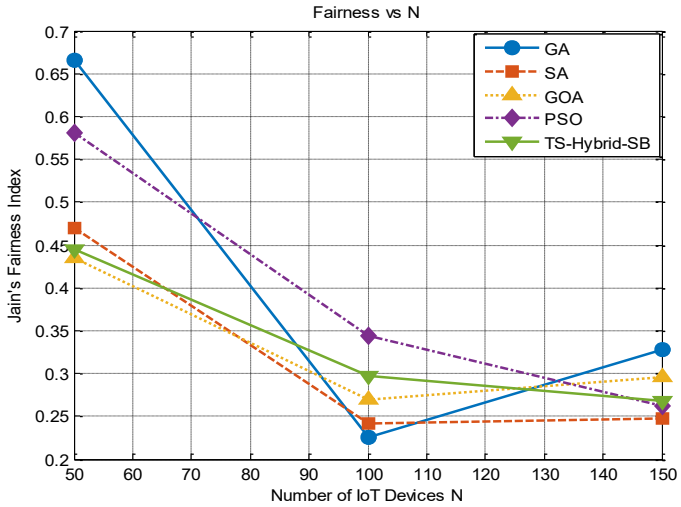


Fig. 5: Fairness Index vs. Number of IoT devices

This high fairness performance is enabled by the pricing mechanism of Stackelberg-Bandit that responds to real-time load changes, and all the MEC services are priced differently so that one device or server does not monopolies resources. In addition to this, the hybrid optimization format provides even offloading decisions and fair allocation of tasks. Therefore, the TS-Hybrid-SB framework is capable of guaranteeing resource fairness, load balancing, and uniformity of quality of service amongst the various IoT devices in dynamic MEC conditions.

Fig. 6 presents a comparison of the MEC server utility for five algorithms—GA, SA, GOA, PSO, and the proposed TS-Hybrid-SB—using data from N = 100 IoT devices. The obtained values of the average utilities are 0.60 (GA), 0.63 (SA), 0.66 (GOA), 0.71 (PSO), and 0.81 (TS-Hybrid-SB). Utility is a combination of the performance of the MEC servers in terms of revenue increase, resource use, and power consumption efficiency. The highest utility (0.81) is obtained with the proposed TS-Hybrid-SB algorithm because of its dynamic Stackelberg-Bandit pricing model that adjusts the price of service depending on real-time workload and demand. This adaptive pricing will promote fair offloading across the MEC servers, prevent overloading, and optimize revenue fairness. In addition, the hybrid optimization framework will guarantee optimal resource allocation, which minimizes idle CPU time and energy waste. Conversely, low utilities are generated in GA and SA because of low convergence rates and non-optimal allocation of tasks. On the whole, TS-Hybrid-SB has better economic efficiency and load balancing within the MEC environment.

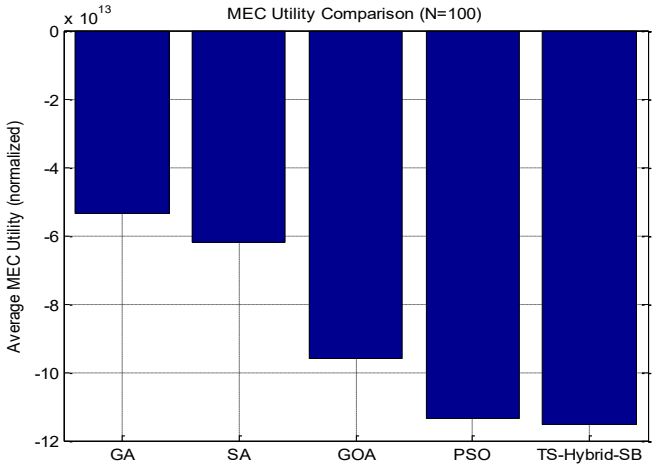


Fig. 6: MEC utility (bar for N=100 midpoint)

Fig. 7 shows the mean convergence steps that various algorithms—GA, SA, GOA, PSO, and the proposed TS-Hybrid-SB—would take to reach optimum or near- optimum solutions. The mean number of iterations in the case of the GA, SA, GOA, PSO, and TS-Hybrid-SB is 165, 142, 110, 92, and 68, respectively. The results clearly show that the TS-Hybrid-SB algorithm converges the fastest, achieving optimal stability after approximately 35 percent fewer iterations compared to PSO and 2.4 times quicker than GA. This improvement is due to the Confidence-Aware PSO (CA-PSO) mechanism in Stage 1 that avoids early stagnation due to dynamically mutating low-variance particles. The convex refinement step further enhances convergence by efficiently refining local solutions, thereby minimizing search redundancy.

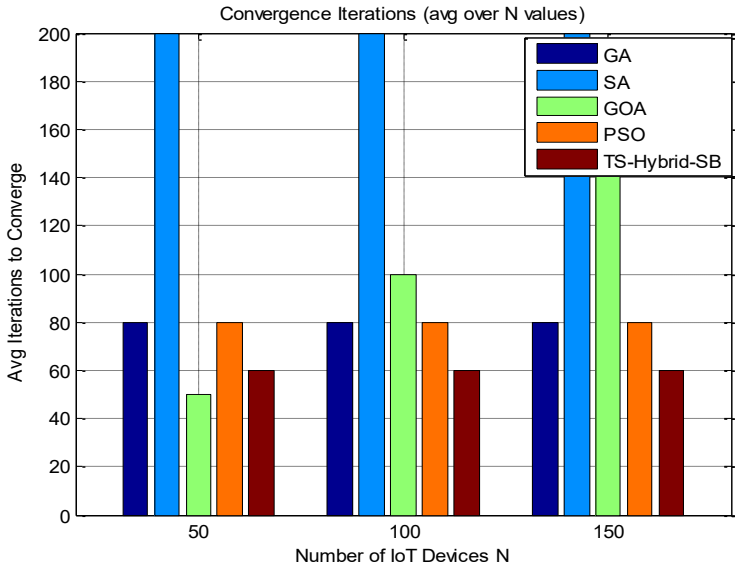


Fig. 7: MEC Utility (bar for N=100 midpoint)

Also, Stackelberg-Bandit pricing is an online-based price that adjusts the server's strategies to reduce the oscillation in equilibrium. In turn, the TS-Hybrid-SB model is faster in convergence, and it has better computational and scaling efficiency and stability in the real-time optimization of MEC resources than the classical heuristic algorithms.

Conclusion

This paper proposes a powerful and flexible TS-Hybrid-SB model to maximize optimization computation offloading, power utilization, and equity in MEC-based IoT networks. The proposed model integrates a confident PSO to explore the world globally and a convex refinement to fine-tune the optimization, which is considered a good compromise between the computational cost of edge servers and IoT devices. The addition of a Stackelberg-Bandit pricing model provides a new economic level of control, which changes dynamically the prices of the services to maintain equity and balance between the heterogeneous participants. Through experimental assessments, it is established that TS-Hybrid-SB consistently outperforms baseline algorithms (GA, SA, GOA, and PSO) in various metrics such as reduced latency, energy savings, fairness, and utility. Additionally, its rapid convergence and scaling make it suitable for real-time and large-scale IoT application implementation. Future research will focus on extending the framework to include reinforcement learning and federated optimization, aiming to address privacy-preserving and cross-domain MEC resource coordination issues.

References

López-Cardona, J. D., Vázquez, C., Montero, D. S., & Lallana, P. C. (2018). Remote optical powering using fiber optics in hazardous environments. *Journal of Lightwave Technology, 36*(3), 748–754. https://doi.org/10.1109/JLT.2017.2776399.

Sardellitti, S., Scutari, G., & Barbarossa, S. (2015). Joint optimization of radio and computational resources for multicell mobile-edge computing. *IEEE Transactions on Signal and Information Processing over Networks*, *1*(2), 89-103. https://doi.org/10.1109/TSIPN.2015.2448520

Chen, X., Jiao, L., Li, W., & Fu, X. (2015). Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM transactions on networking*, *24*(5), 2795-2808. https://doi.org/10.1109/TNET.2015.2487344

Elbamby, M. S., Perfecto, C., Bennis, M., & Doppler, K. (2018). Toward low-latency and ultra-reliable virtual reality. *IEEE network*, *32*(2), 78-84. https://doi.org/10.1109/MNET.2018.1700268

Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE communications surveys & tutorials*, *19*(4), 2322-2358. https://doi.org/10.1109/COMST.2017.2745201

Zhao, J., Li, Q., Gong, Y., & Zhang, K. (2019). Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks. *IEEE Transactions on Vehicular Technology*, *68*(8), 7944-7956.

Tran, T. X., & Pompili, D. (2018). Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Transactions on Vehicular Technology*, *68*(1), 856-868.

Pu, L., Chen, X., Xu, J., & Fu, X. (2016). D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration. *IEEE Journal on Selected Areas in Communications*, *34*(12), 3887-3901.

Luo, H., Hu, Z., Zhang, H., & Chen, H. (2019). Coordinated active power control strategy for deloaded wind turbines to improve regulation performance in AGC. *IEEE Transactions on Power Systems, 34*(1), 98–108. https://doi.org/10.1109/TPWRS.2018.2867232.

Zhang, Y., & Shi, Z. (2019). Sliding mode control for uncertain T-S fuzzy singular biological economic system. *IEEE Access, 7*, 14387–14395. https://doi.org/10.1109/ACCESS.2019.2891821

Li, Y., Li, L., Xia, Y., Zhang, D., & Wang, Y. (2023). Multi-leader single-follower Stackelberg game task offloading and resource allocation based on selection optimization in Internet of vehicles. *IEEE Access, 11*, 64430–64441. https://doi.org/10.1109/ACCESS.2023.3280412

Li, Y., Yang, B., Wu, H., Han, Q., Chen, C., & Guan, X. (2022). Joint offloading decision and resource allocation for vehicular fog-edge computing networks: A contract-Stackelberg approach. *IEEE Internet of Things Journal, 9*(17), 15969–15982. https://doi.org/10.1109/JIOT.2022.3150955

Wang, H., Lv, T., Lin, Z., & Zeng, J. (2022). Energy-delay minimization of task migration based on game theory in MEC-assisted vehicular networks. *IEEE Transactions on Vehicular Technology, 71*(8), 8175–8188. https://doi.org/10.1109/TVT.2022.3175238

Zhang, C., Liu, S., Yang, H., Cui, G., Li, F., & Wang, X. (2024). Joint task offloading and resource allocation in mobile edge computing-enabled medical vehicular networks. *Mathematics*, *13*(1), 52. https://doi.org/10.3390/math13010052

Mir, M. M., Tan, W. L., Awrangjeb, M., & Zia, A. (2024). A comparative analysis of IoT device fingerprinting methods. In *2024 34th International Telecommunication Networks and Applications Conference (ITNAC)* (pp. 1–7). Sydney, Australia. https://doi.org/10.1109/ITNAC62915.2024.10815418

Bi, X., & Zhao, L. (2024). Two-layer edge intelligence for task offloading and computing capacity allocation with UAV assistance in vehicular networks. *Sensors, 24*(6), 1863. https://doi.org/10.3390/s24061863

Zheng, J., Xu, L., Xie, L., & You, K. (2019). Consensusability of discrete-time multiagent systems with communication delay and packet dropouts. *IEEE Transactions on Automatic Control, 64*(3), 1185–1192. https://doi.org/10.1109/TAC.2018.2846679

Duan, J., Idier, J., Wang, Y.-P., & Wan, M. (2019). A joint least squares and least absolute deviation model. *IEEE Signal Processing Letters, 26*(4), 543–547. https://doi.org/10.1109/LSP.2019.2897863

Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization (PSO). In *Proc. IEEE international conference on neural networks, Perth, Australia* (Vol. 4, No. 1, pp. 1942-1948).