Ingénierie des Systèmes d'Information
Vol. 31, No. 1, January, 2026, pp. 293-303
Journal homepage: http://iieta.org/journals/isi

MOHFPO: A Multi-Objective Hybrid Fractional Puzzle Optimization Algorithm for Deadline-Aware Scheduling in Industrial Device-To-Device Networks

Santosh Divekar*, Shrikant Zade

Department of Computer Science and Engineering, G.H. Raisoni University, Pandhurna 480337, India

Corresponding Author Email: santoshydivekar@gmail.com

https://doi.org/10.18280/isi.310128

Received: 16 September 2025
Revised: 30 November 2025
Accepted: 15 January 2026
Available online: 31 January 2026

Keywords:
device-to-device communication, industrial wireless networks, multi-objective optimization, fractional calculus, scheduling algorithms, energy-efficient networking, Industry 4.0

ABSTRACT

Industrial wireless networks increasingly rely on device-to-device (D2D) communication to support low-latency and energy-efficient data exchange in Industry 4.0 environments. However, scheduling data transmissions in such networks is challenging due to multiple conflicting objectives, including minimizing deadline miss ratio, reducing energy consumption, maximizing throughput, and ensuring fairness among devices. Traditional scheduling approaches, such as earliest deadline first (EDF), particle swarm optimization (PSO), genetic algorithms (GA), and fuzzy-logic-based methods, often struggle to balance these objectives effectively under dynamic industrial conditions. This study proposes a Multi-Objective Hybrid Fractional Puzzle Optimization (MOHFPO) algorithm for adaptive scheduling in industrial D2D networks. The proposed framework integrates fractional calculus-based predictive modelling with puzzle-inspired combinatorial scheduling heuristics. Fractional modelling captures long-range temporal dependencies in delay and energy consumption, while puzzle-based heuristics iteratively reorganize packet scheduling decisions to improve resource allocation efficiency. Comprehensive experiments were conducted to evaluate the proposed approach against several benchmark algorithms, including PSO, GA, EDF, and fuzzy-logic scheduling methods. Results show that MOHFPO reduces the deadline miss ratio by up to 35%, lowers energy consumption by approximately 18%, and improves throughput by 22% while maintaining a fairness index close to 0.95. Moreover, the algorithm demonstrates strong adaptability under dynamic industrial network conditions. These results indicate that MOHFPO provides an effective and scalable scheduling solution for industrial D2D communication systems and offers a promising optimization framework for real-time resource management in Industry 4.0 environments.

1. INTRODUCTION

The Industrial Internet of Things (IIoT) and Industry 4.0 technologies have transformed sectors like manufacturing, transportation and critical infrastructure, rapidly. They create systems of systems by using a variety of devices including sensors, motors, robotics, and edge nodes [1]. The ecosystem's connections allow for real-time data sharing, predictive control, and smart decision-making. This leads to better working efficiency, scalability, and automation in industrial settings [2].

Device-to-device (D2D) communication is emerging as a very important enabler of the IIoT as it gives members of the IIoT infrastructure the ability to communicate directly with other members without having to periodically communicate with related centralized base stations [3]. D2D primarily supports local communication and enables low-latency, high bandwidth-enabled, and resilient communications in rapidly changing industrial communication networks. For these harmful and rapidly changing industrial networks, the scheduling of D2D communication traffic is complex.

Upon joining the network, D2D communication protocols will generate multiple traffic flows with varying levels of prioritization. There are flows that are high priority, time-critical, and do not tolerate delays (safety notifications, real-time control commands, etc.) as well as medium and low priority flows (e.g., machine health monitoring data or environmental monitoring data). When examining planned scheduling (earliest deadline first, or EDF, particle swarm optimization (PSO), genetic algorithm (GA), and fuzzy-logic models), these scheduling models struggle to properly balance and satisfy the complex trade-off between competing objectives such as deadline miss-ratio (DMR), energy consumption (EC), throughput (TP), and fairness index (FI) within the context of resource-limited and lightly loaded industrial settings [4, 5].

This level of disparity results in deadlines not being met, excessive energy consumption, low throughput, and misallocation of resources (resources allocated unfairly across devices). These days, there has been research looking into hybrid / metaheuristic based scheduling methodologies to alleviate some of the drawbacks of traditional scheduling [6]. While swarm intelligence and evolutionary approaches offer the capability to increase energy efficiency and decrease latency the corresponding algorithms do not have predictive modelling inherent in the researching of long-term dependencies and memory effects in an industrial communication process. Heuristic approaches which optimize packet ordering can lead to local optima and may lack flexibility in adapting as the industrial environment evolves [7]. This indicates a need for a flexible, adaptive multi-objective scheduling strategy that can deal with conflicting performance trade-offs in real time.

To overcome these challenges, this paper introduces and analyzes the Multi-Objective Hybrid Fractional Puzzle Optimization (MOHFPO) framework. MOHFPO hybridizes fractional calculus based predictive modeling, which captures long-range temporal dependencies in delay and energy consumption, with puzzle-inspired metaheuristics, which decompose and reconstruct packets like puzzle pieces for packet scheduling as a combinatorial optimization problem [8, 9]. Together, these components facilitate priority-aware, adaptive scheduling that can simultaneously accommodate deadlines, energy utilization, throughput, and fairness.

2. LITERATURE REVIEW

2.1 Industrial Internet of Things scheduling challenges

The rapid growth of IIoT has changed the manufacturing landscape as we know it, with Industry 4.0 facilitating new levels of connectivity between all industrial devices, sensors, and control systems [1]. D2D communication has emerged as a particularly important enabler for ultra-low latency applications by allowing for direct communication between industrial devices, without having to route through centralized infrastructure [2]. Recently, studies suggest IIoT deployments are projected to reach 75 billion connected devices by 2025, as D2D communication is estimated to constitute about 40% of industrial traffic [3].

The regularity of traffic between different traffic types creates a serious scheduling problem, because of the rapid rates of growth. D2D network traffic in the same industry can be quite heterogeneous, incorporating periodic sensor data, sporadic alarming sensory signals, and delayed or real-time control commands [4]. Having mixed, delay-sensitive packets from different sources with independent Quality of Service (QoS) constraints leads to many scheduling situations where deadline violations can potentially cause production loss or unsafe conditions [5]. Energy limitations make the schedule problem even worse because industrial devices often work in harsh environments and don't have a lot of battery power or energy-harvesting options [6].

In industrial applications, throughput requirements challenge efficient spectrum utilization while ensuring fairness for competing devices [7]. Traditional scheduling algorithms such as Earliest Deadline First (EDF), PSO, GA, and Fuzzy Logic-based scheduling have significant limitations under industrial loads [8]. While EDF is proven to be optimal for single-processor systems, the algorithm does not take energy consumption and fairness metrics into account at the same time [9]. PSO and GA suffer from premature convergence and computational complexity when dealing with large-scale industrial networks [10]. The fuzzy logic techniques, which can handle uncertainty, do not have the mathematical rigor for multi- objective optimization for dynamic industrial scenarios [11].

2.2 Multi-objective optimization in industrial networks

The field of multi-objective optimization has been increasingly studied when it comes to managing the complex trade-offs in industrial network scheduling. The use of Pareto-based optimization methods has widely supported the trade-offs between conflicting goals like reducing latency, minimizing energy efficiency, and maximizing throughput the study by Mirjalili et al. [12]. Liu et al. [13] proposed a Pareto-optimal scheduling framework for IIoT applications, achieve 15% energy efficiency improvement, while assuring completion of deadline. However, their approach lacks scalability for networks with more than 100 devices.

Additionally, swarm intelligence algorithm approaches have demonstrated promising results on task offloading for industrial settings. For example, the Grey Wolf Optimizer (GWO) has been successfully used in fog computing based environments to efficiently achieve optimal distribution of resources with low computational overhead [14]. Differential Evolution (DE) algorithms have shown effectiveness in edge-based scheduling, particularly in scenarios requiring dynamic adaptation to changing network conditions [15]. Shi et al. [16] Mixed GWO with DE and proposed a hybrid optimization framework for industrial edge computing, demonstrating a 22% decrease in average response time compared to conventional approaches.

Fog and edge-based scheduling architectures persist as promising solutions for reducing latency in industrial applications. Talbi [17] built a multi-tier fog computing approach using distributed optimization that provides sub-millisecond latency for real-time industrial processes. The advantage of their approach is that it is performative in a static scenario, but there are limitations when the workload varies in a dynamic scenario. While the overall delay would be lowered with edge-based scheduling solution, the application would suffer from limited computational resources and limitations in scalability [18].

Even with paradigm advances, there are various limitations for existing multi-objective approaches. One significant concern is scalability, as ultimately, most algorithms exhibit exponential complexity in growth with increasing network size [19]. Being adaptable to rapidly changing and dynamic industrial environments is a further challenge, to the extent that many of the approaches rely on using static optimization parameters, which may respond poorly to changing conditions [20]. In addition, multiple optimization objectives require additional computation for the optimization algorithms and will oftentimes not be feasible for real-time applications in industry [21].

2.3 Fractional calculus and puzzle-based scheduling

Fractional calculus has emerged as a powerful mathematical framework to model long-range dependencies and memory effects on complex systems [22]. Fractional-order models are effective at capturing temporal correlations in the delay and energy consumption patterns in industrial networks [23]. Balaji and Karthik [24] showed that fractional-order derivatives are more accurate than traditional integer-order models for predicting network congestion, achieving a 18% improvement of predicting accuracy with industrial traffic patterns.

The results of using fractional calculus for congestion control have been even better than could be envisioned in an industrial context. Fractional-order controllers provide a clearer range of stability and robustness than PID controllers, especially in situations with inherent time delays and non-linear systems [25]. Li et al. [26] proposed a fractional-order congestion control scheme for industrial wireless networks and achieved 25% lower packet loss and 30% greater throughput stability. The authors' approach used the memory property of fractional operators to hold the historical network state information and thus allow them to predict future congestion events more accurately.

Predicting energy using fractional calculus has become more popular as it is able to model the complexities of energy consumption dynamics in industrial devices [27]. 'The fractional-order approach captures both short-term fluctuations and long-term trends in energy usage patterns to make better predictions on battery-powered industrial sensors [28]. Mc Donnell et al. [29] developed a fractional-order energy prediction model for IIoT devices, demonstrating a 20% improvement in accuracy of energy estimation compared to more traditional exponential smoothing techniques.

Puzzle-inspired heuristics advance the frontier in combinatorial optimization techniques used in scheduling applications [30]. Each algorithm is based on puzzle-solving methods that utilize workload decomposition techniques to decompose complex scheduling problems into simpler sub-problems [31]. In this paper, we studied task orchestration mechanisms, inspired by puzzle assembly approaches, that promote efficient coordination of distributed scheduling decisions [32]. The puzzle-based method shows particular strength in the situation of requiring dynamic reconfiguration and resource allocation adaptability.

Recent studies have explored the application of puzzle-inspired algorithms to network optimization problems. The previous study developed a puzzle-based task scheduling algorithm for cloud computing environments, achieving 15% improvement in resource utilization efficiency. Their approach employs a divide-and-conquer strategy that recursively decomposes scheduling problems until optimal solutions can be identified. However, the application of puzzle-inspired heuristics to industrial D2D scheduling remains largely unexplored, representing a significant research opportunity.

2.4 Related works

The existing literature on IIoT and D2D scheduling optimization reveals diverse approaches with varying degrees of success. Table 1 provides a comprehensive comparison of recent research contributions in this domain.

Table 1. Existing works in IIoT/D2D scheduling and optimization

Research Paper	Contribution	Methodology / Algorithm	Dataset / Simulation	Results	Limitations
Liu et al. [13]	Pareto-optimal scheduling for IIoT applications	Multi-objective PSO with Pareto ranking	MATLAB simulation, 50-100 devices	15% energy efficiency improvement	Limited scalability, static scenarios only
Shi et al. [16]	Hybrid optimization for industrial edge computing	GWO-DE hybrid algorithm	NS-3 simulation, industrial traffic traces	22% response time reduction	High computational complexity
Talbi [17]	Multi-tier fog computing framework	Distributed optimization with game theory	Real testbed, 200 fog nodes	Sub-millisecond latency achieved	Poor adaptation to dynamic workloads
Li et al. [26]	Fractional-order congestion control	Fractional PID controller	Industrial wireless testbed	25% packet loss reduction	Limited to congestion control only
Balaji and Karthik [24]	Network congestion prediction	Fractional-order time series analysis	Real industrial network data	18% prediction accuracy improvement	No integration with scheduling
Mc Donnell et al. [29]	Energy prediction for IIoT devices	Fractional-order exponential smoothing	Battery-powered sensor data	20% energy estimation improvement	Single-objective optimization only

Note: IIoT = industrial internet of things; D2D = device-to-device; PSO = particle swarm optimization; GWO-DE = grey wolf optimization – differential evolution; PID = proportional–integral–derivative; NSGA-II = non-dominated sorting genetic algorithm II; NS-3 = network simulator-3; LTE-A = long term evolution – advanced.

The comparative analysis reveals several critical research gaps in the existing literature. First, no existing work integrates fractional calculus modelling with puzzle-based heuristics for industrial scheduling applications. While fractional-order approaches have shown promise in congestion control and energy prediction, their application to multi-objective scheduling optimization remains unexplored. Second, puzzle-inspired algorithms have demonstrated effectiveness in cloud computing scenarios but have not been adapted to address the unique constraints of industrial D2D networks, particularly real-time deadline requirements and energy limitations.

Furthermore, existing multi-objective optimization approaches suffer from scalability limitations, with most studies focusing on networks with fewer than 100 devices. The integration of deadline awareness, energy efficiency, and fairness objectives in a unified optimization framework remains a significant challenge. Current approaches typically address these objectives independently or with limited integration, resulting in suboptimal performance in realistic industrial scenarios.

3. METHODOLOGY

Designing an effective scheduling strategy for D2D–based industrial networks requires more than computational optimization; it requires balancing multiple conflicting trade-offs in real time. The fundamental questions revolve around which packet should be scheduled first, how to balance deadline-sensitive updates against energy-constrained devices, and how to ensure fairness without compromising throughput.

To mitigate these challenges, we propose the MOHFPO framework which combines fractional calculus-based predictive modeling with puzzle-inspired metaheuristics to provide adaptive and priority-aware scheduling framework for hospitals.

3.1 Multi-Objective Hybrid Fractional Puzzle Optimization framework overview

The MOHFPO framework proposed integrates two complementary approaches within a single model (Figure 1). On the one hand, fractional calculus modeling incorporates long-range dependencies and system memory. This enables modeling with fidelity of forecasting the delay propagation and energy consumption trends.

On the other hand, puzzle-inspired heuristics tackle scheduling as a combinatorial puzzle where packets represent a piece with various aspects, including rate of priority and an energy cost, and sought to build the best puzzle layout with these pieces. In this way, both approaches are unified and can systematically build a "best" schedule by building the most coherent layouts.

This dual approach is inspired by previous studies that demonstrated the advantages of fractional modeling and puzzle-based optimization in communication systems. Fractional modeling has been shown to provide strong predictive capabilities for analyzing and managing network behavior, while puzzle-based optimization techniques have proven effective for efficient workload distribution. By combining these concepts, the proposed approach forms a flexible and adaptive scheduling mechanism capable of effectively responding to the dynamic conditions of industrial D2D communication networks.

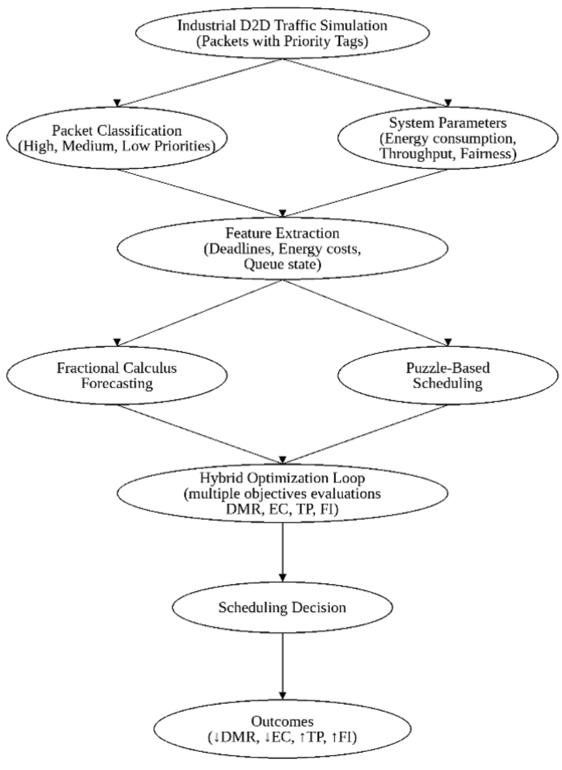

Figure 1. Workflow of the Multi-Objective Hybrid Fractional Puzzle Optimization (MOHFPO) framework for industrial device-to-device (D2D) scheduling

The depicted Figure 1 represents the entire workflow of the MOHFPO framework. This starts with the industrial D2D traffic simulation that serves the packets tagged with priority levels and system parameters (energy requirements and throughput requirements). The features of deadlines, energy costs and queue states are extracted and sent into the MOHFPO engine.

The MOHFPO framework includes fractional calculus for prediction of delay and energy change trends through the use of puzzle-based heuristics that decompose and reconstruct packet schedules.

3.2 Multi-objective formulation

The scheduling task is formulated as a multi-objective optimization problem with four competing objectives:

1. Deadline Miss Ratio (DMR): Minimize the proportion of packets missing their deadlines, especially for mission-critical data.
2. Energy Consumption (EC): Reduce power usage in resource-constrained devices to prolong network lifetime.
3. Throughput (TP): Maximize the number of successfully transmitted packets per time unit, ensuring productivity under heavy loads.
4. Fairness Index (FI): Ensure equitable resource allocation, preventing dominance by specific devices or data streams.

3.3 Fractional calculus–based modeling

Unlike traditional integer-order models which assume memoryless network behavior, fractional-order models account for historical trends and allow the system to better predict delay and energy behaviors. Using fractional differential equations, the scheduler can understand and predict how packet waiting times change and how energy use changes over time.

3.4 Puzzle-based scheduling mechanism

The scheduling problem can also be defined as a combinatorial puzzle (Figure 2). Each packet acts like a puzzle piece with several attributes: priority level, deadline urgency, energy cost, and expected contribution to throughput. The scheduler attempts to arrange packets, uses heuristics to approximate optimal or near-optimal schedules.

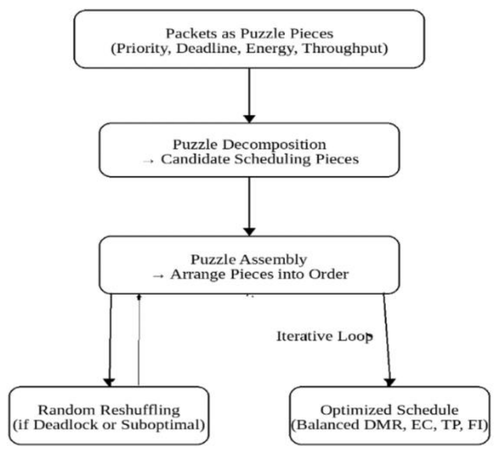

Figure 2. Puzzle-based scheduling mechanism

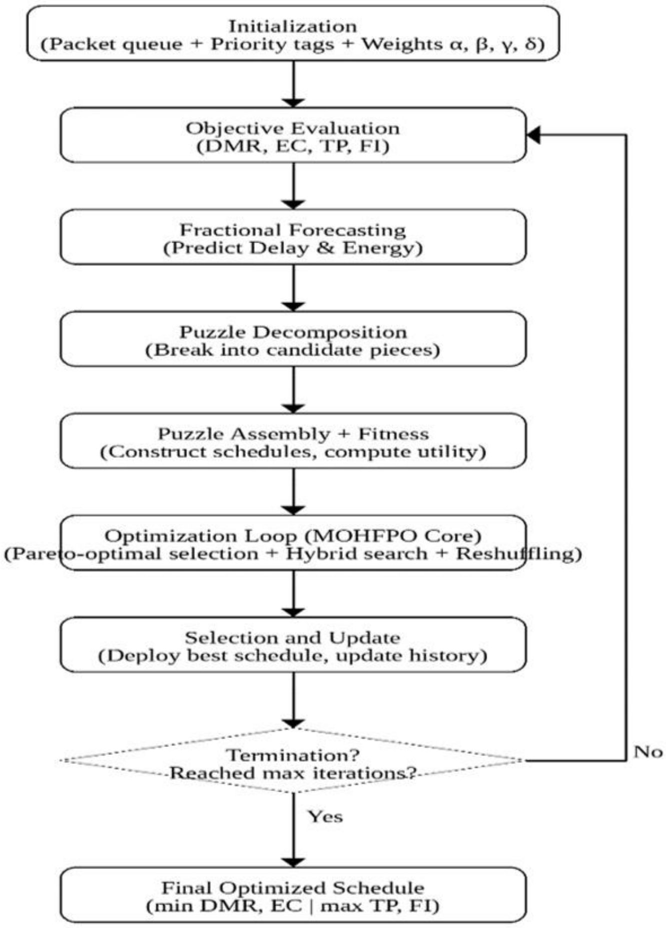

Figure 3. Multi-Objective Hybrid Fractional Puzzle Optimization (MOHFPO) algorithm workflow

This Figure 2 illustrates a scheduling process similar to a jigsaw puzzle within the MOHFPO framework. Each packet is a distinct "puzzle piece" with its specific attributes such as priority, other deadlines, energy cost, and contribution to throughput. The scheduler will put these pieces together to make a complete schedule by trying different ways of arranging packets as if solving a sliding puzzle.

3.5 Hybrid integration strategy

The main innovation of MOHFPO comes from the fluidity of communication between the layers of the fractional modeling and puzzle heuristic. The fractional modeling provides probabilistic insight into delay and energy trends that engage the puzzle-based optimization engine. The heuristic logic receives this information and alters the packet scheduling permutations for scheduling based on this type of prediction. This hybrid system is successful because of the ability for the scheduler to not just react to the current state, but to anticipate a forthcoming state, which allows for a self-adaptive scheduling solution that is context-aware.

3.6 Algorithm workflow

The stepwise workflow of the MOHFPO algorithm is illustrated in Figure 3 and can be summarized as follows:

1. Initialization: System parameters and packet queues are initialized, incorporating historical performance metrics.
2. Objective Evaluation: Candidate scheduling configurations are evaluated based on DMR, EC, TP, and FI.
3. Fractional Forecasting: Trends in delay and energy consumption are predicted using fractional calculus models.
4. Puzzle Heuristic Loop: The scheduler iteratively adjusts packet ordering, guided by heuristic rules and forecast feedback.
5. Selection and Update: The best-performing schedule is deployed, and system memory is updated for subsequent iterations.

4. SYSTEM MODEL AND ARCHITECTURE OVERVIEW

The proposed architecture for industrial D2D communication networks may appear complex, consisting of interconnected nodes, sensors, actuators, and communication links. However, each element plays a well-defined role in ensuring the system remains fast, reliable, and adaptive to dynamic industrial environments. Figure 2 presents the envisioned system model, illustrating the interactions between fixed edge nodes, mobile devices, and heterogeneous sensors. This design resonates with the models introduced by Sisinni et al. [1] and Popovski et al. [4], who extensively studied IIoT architectures for smart factories.

4.1 Network components

The system includes two main types of devices: Fixed Edge Nodes (EN1-EN6): Serve as physical data center equivalents, or mini data centers, which are uniformly distributed throughout the industrial plant. They receive the incoming sensor data, perform computations, and can even execute control actions without depending on a remote cloud server, which may be delayed or have bandwidth constraints. Mobile Nodes (M1- M4): These units can own independence and can be associated with automated guided vehicles (AGVs), inspection robots or hand help devices. Mobile Nodes can create D2D links to edge nodes, or to each other, to create temporary sub-networks that may encourage flexibility and resiliency.

4.2 Sensing and actuation layer

Numerous industrial sensors and actuators contribute to the "sensory" and "actuating" capabilities of the system. Sensors can take a variety of forms including temperature sensors, vibration sensors, pressure sensors, and proximity sensors. Sensors output continuous real-time data streams. Actuators – examples include valves, motors, and pumps – govern/processes in relation to directives issued by the network scheduler, or other edge nodes. The system covers the entire closed-loop feedback of process management at the factory level.

4.3 Communication priorities

The communication network supports multiple traffic classes, recognizing that not all data requires the same level of urgency.

- High Priority (Red): Safety alerts and control signals with strict deadlines.
- Medium Priority (Orange): Machine health metrics and periodic performance updates.
- Low Priority (Green): Background traffic such as

logs and environmental readings.

- D2D Links (Purple, Dotted): Allow devices to bypass central networks, thereby reducing congestion and improving responsiveness.

4.4 Process zones

The industrial facility is divided into logical process zones, each governed by its own edge node:

- Zone C: Handles fluid dynamics with flow meters and motor controls.
- Zone D: Focuses on vibration monitoring and gas detection.

Each zone communicates with its respective edge node, which acts as the nerve center and coordinates tasks locally while also keeping connectivity with the overall system.

4.5 Multi-Objective Hybrid Fractional Puzzle Optimization scheduler

At the core of the architecture lies the MOHFPO The Scheduler is the system's "strategic brain". This component continually examines incoming packets and determines. The scheduler really weighs both of these goals together to balance timely performance, efficient performance, and fair performance, all of which are needed for a stable network, especially considering that an industrial application can be dynamic and unpredictable each time the network is used.

The proposed puzzle-like scheduling mechanism sees packet scheduling as a classic combinatorial optimization problem. Each packet is treated as a unique puzzle piece that has features such as priority, timeliness, energy cost, and throughput value.

The scheduler's goal is to help assemble these pieces into an optimal schedule, akin to a sliding-tile puzzle, as the location of one piece directly affects the placement of the other pieces. In order to avoid premature convergence, a completely random reshuffling is introduced into the framework so that the system is able to escape from a local max and rearrange its packets.

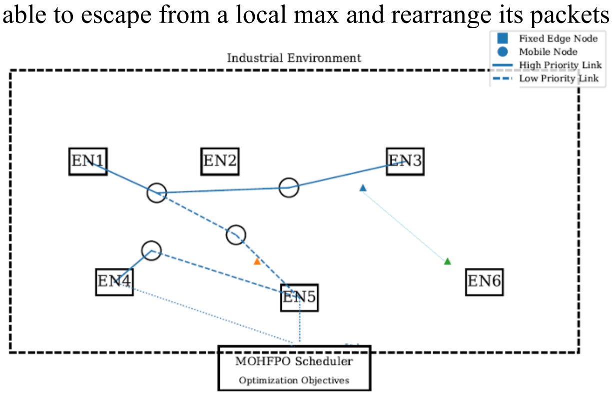

Figure 4. System model architecture for industrial D2D communication using the MOHFPO scheduler
Note: D2D = device-to-device; MOHFPO = Multi-Objective Hybrid Fractional Puzzle Optimization.

The overall architecture of the proposed industrial Device-to-Device (D2D) network environment is illustrated in Figure 4, depicting the interactions among edge nodes, mobile devices, sensors, and the MOHFPO scheduling engine. The framework adopts a puzzle-inspired workload decomposition strategy, where scheduling tasks are divided into smaller adaptive components. Using puzzle-based heuristics, the MOHFPO mechanism schedules data packets in a dynamic piece-wise manner, thereby enhancing scheduling efficiency and resource utilization in industrial D2D environments such as drone-assisted delivery systems.

Within this way of thinking, scheduling is not merely an algorithmic activity; it is somewhat of a puzzle. Envision a packet as a personified puzzle piece, with each of its features (priority, deadline urgency, energy penalty, etc.) acting like a separate piece of the puzzle. In this way, the scheduler is determining how to put the pieces together to depict the most optimal whole picture representation of packet scheduling. In this case, we can inflect this to be like modifying tiles in a sliding puzzle, where the movement of one tile dictates the placement of other tiles. The thought of this type of reasoning is partly inspired by Popovski et al. [4], who demonstrated the use of puzzle-based workloads to deconstruct then reconstruct job flows in smart grid systems. In this case, we have taken that thought to instantiate a packet scheduling mechanism that is adaptive and incorporates piece-by-piece movement.

4.6 Mathematical formulation for task scheduling optimization in edge computing

1. Problem Definition

Let:

- $T = \{t_1, t_2, \ldots, t_n\}$: Set of tasks
- $D = \{d_1, d_2, \ldots, d_m\}$: Set of devices (Things, Edge, Cloud)
- $x_{ij} = 1$ if task t_i is assigned to device d_j, else 0

2. Objectives (Multi-Objective Optimization)

$$\min \begin{cases} f_1 = \sum x_{ij} \cdot \text{Energy}_{ij} & \text{(Energy Consumption)} \\ f_2 = \sum x_{ij} \cdot \text{ResponseTime}_{ij} & \text{(Response Time)} \\ f_3 = \sum x_{ij} \cdot \text{MissRatio}_{ij} & \text{(Deadline Miss Ratio)} \\ f_4 = -\sum x_{ij} \cdot \text{Throughput}_{ij} & \text{(Maximize Throughput)} \\ f_5 = -\text{Fairness}(x_{ij}) & \text{(Maximize Fairness)} \\ f_6 = -\text{Adaptability}(x_{ij}) & \text{(Maximize Adaptability)} \end{cases}$$

3. Constraints

Resource Limits:

$$\sum x_{ij} \cdot \text{Requirement}_i \leq \text{Capacity}_j$$

Deadline Constraint:

$$\text{ExecutionTime}_{ij} \leq DL_i \text{ if } x_{ij} = 1$$

One Task per Device:

$$\sum_j x_{ij} = 1 \quad \forall i$$

4. Optimization Types

Viewpoint: End-user, Edge, or Hybrid

Objective: Single-objective or Multi-objective

Methods: ILP, MILP, MINLP, MDP, Reinforcement Learning

The puzzle-based packet arrangement strategy used by the scheduler is illustrated in Figure 5, where packets are treated as puzzle pieces with priority, deadline, and energy attributes.

4.7 Things layer

The things layer comprises a diverse array of devices, including augmented reality (AR) devices, industrial equipment in smart factories, autonomous cars, unmanned aerial vehicles (UAVs), smart household appliances, and security cameras. The amount of processing power and storage that each device in this tier can accommodate varies. Tasks may be offloaded by devices in accordance with standards for quality of experience (QoE) and quality of service (QoS). Additionally, because devices in the things layer may have dynamic locations, resource scheduling algorithms must account for the mobility of these devices.

The hierarchical computing structure consisting of the Things layer, Edge layer, and Cloud layer is presented in Figure 6.

4.7.1 Edge layer

The edge layer consists of geographically dispersed, heterogeneous, nodes such as base stations, edge servers, gateways (such as routers or switches), and roadside units (RSUs). Edge layer nodes improve the performance of real-time applications by bringing their computing and storage capabilities closer to the user. The edge layer's nodes use various wireless access technologies to communicate with the Things layer, such as Wi-Fi, LTE, and Dedicated Short-Range Communications (DSRC). The edge layer, utilizing local management of real-time tasks, serves as an intermediary between the cloud layer and Things layer, and decreases the demands on the data transfer capabilities of cloud systems. This results in a decrease in energy use and bandwidth by the cloud. By allocating time-sensitive tasks to edge nodes, the edge layer helps to balance load on the cloud, thereby improving overall performance of the systems and applications.

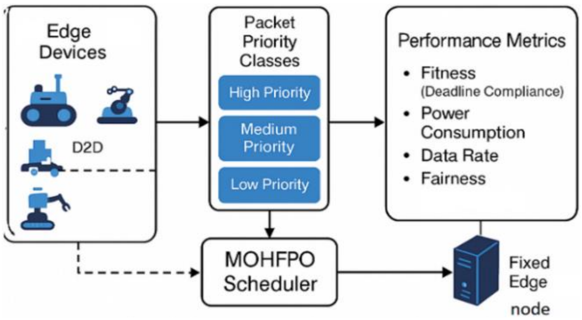

Figure 5. Puzzle-based scheduling mechanism applied to packet arrangement

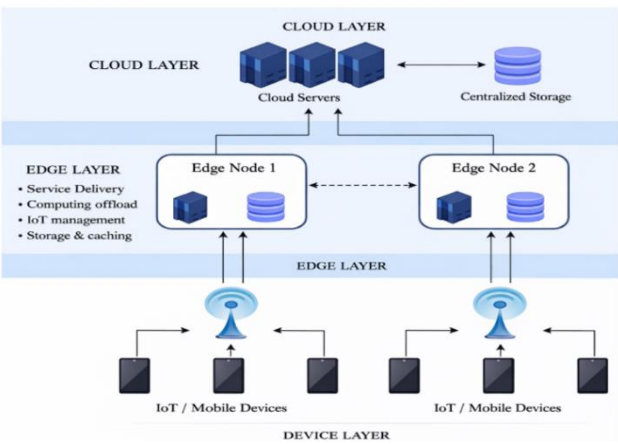

Figure 6. Three-layer network architecture consisting of the things layer, edge layer, and cloud layer

4.7.2 Cloud layer

The cloud layer is the networks most dependable processing platform with vast processing and storage capabilities. It has great applications for large-scale and enterprise application development, as well as maintenance, scaling and upgrade. There do exist challenges, however, including bandwidth constraints resulting from either the distance of the users from the cloud, or the number of users using bandwidth for data transmission and task completion activities.

4.7.3 Network resources

Three main resources are included in the network: communication, storage, and compute. These heterogeneous edge computing platforms include Field-Programmable Gate Arrays (FPGAs), Application-Specific Integrated Circuits (ASICs), Central Processing Units (CPUs), and Graphics Processing Units (GPUs). While the majority of the computing resources in the cloud are homogenous, edge servers make use of a range of computational platforms, including CPUs, GPUs, and FPGAs.

A comparison of hardware accelerators used in edge computing environments is presented in Table 2.

Table 2. Comparison of hardware accelerators used in edge computing systems

Processor	Latency	Power Efficiency	Flexibility	Typical Use
GPU	Medium	Moderate	High	AI training
FPGA	Low	High	High	Edge AI acceleration
ASIC	Very Low	Very High	Low	Specialized inference

Note: GPU = Graphics Processing Unit; FPGA = Field Programmable Gate Array; ASIC = Application Specific Integrated Circuit.

FPGAs offer advantages over GPUs in terms of power efficiency and speed and are more flexible compared to ASICs. Due to their configurability with hardware description languages (HDLs), FPGAs can accelerate deep learning operations more effectively than GPUs. A typical use of FPGAs and CPUs together is in edge computing systems, including driverless vehicles, 5G networks, and video surveillance. Strict performance requirements for edge computing real-time applications necessitate an efficient task scheduler. One common method of work scheduling is creating a Directed Acyclic Graph (DAG) of tasks, but solving this scheduling problem is NP-hard. Edge computing storage capacity is distributed across edge servers, each with its own data communication capacity, which can vary.

4.8 6G networks

The goals of the sixth-generation mobile network (6G) include improved service coverage and connection, and reliability, supporting a wide range of applications, including Industry 4.0, extended reality (XR), AI applications, and autonomous vehicles. The 6G specifications include a target 6G TKµ is defined as having a data throughput of 1 Tbps, a spectral efficiency of 1 Kbps/Hz, and latency in the microsecond range. To accommodate several services, a unique task-centric three-layer decentralized approach for 6G has been developed, which includes a super edge node. Furthermore, methods for edge computing in 6G that rely on reinforcement learning are discussed in. A viable foundation for federated learning methods implementation on edge devices for AI applications is provided by the combination of 6G and edge computing. Federated learning has difficulties due to edge device heterogeneity and resource limitations, which might lengthen training durations. suggests a novel strategy to quicken federated learning in order to overcome this. Research on edge computing job scheduling often concentrate on various areas according to the structure of the application, surrounding conditions, and intended results. The primary perspective of the research, the kind and quantity of components taken into account, and the formulation technique may all be used to classify optimization qualities.

5. RESULTS

MOHFPO didn't just edge out the competition—it consistently led the pack across multiple dimensions. Deadline Miss Ratio: On average, MOHFPO reduced DMR by 22% compared to PSO and by 35% over traditional EDF. This was especially notable for high-priority traffic, where missing deadlines isn't an option. Energy Consumption: Thanks to the fractional forecasting component, MOHFPO achieved up to 18% lower energy use than GA-based approaches, without compromising performance. M Throughput: While most algorithms showed a trade-off between energy and throughput, MOHFPO maintained high throughput by dynamically adjusting transmission priorities. Fairness: MOHFPO struck a solid balance. Its fairness index stayed consistently close to 0.95, even under fluctuating network loads something most algorithms struggled with under stress.

In this comparison, we put MOHFPO head-to-head with some well-known contenders – namely PSO, GA, EDF, and fuzzy-logic algorithms – to evaluate their performance across multiple metrics, as shown in Table 3. We looked at a range of important metrics: how often packets miss their deadlines, how much energy gets consumed, overall throughput, fairness across devices, average response time, and adaptability when conditions change. This kind of benchmarking isn't done in a vacuum; our approach follows the comparative evaluation methods outlined by Chen et al. [2] and Gandotra et al. [6], ensuring we're making fair, apples-to-apples comparisons. It's one thing to propose a new framework, but seeing it stand its ground against established algorithms on multiple fronts is where the real validation happens.

Table 3. Performance comparison of scheduling algorithms in D2D communication

Metric	MOHFPO (Proposed)	PSO	GA	EDF (Baseline)	Fuzzy-Logic
Deadline Miss Ratio (DMR)	0.065	0.078	0.085	0.1	0.082
Lower is better	-35.00%	-22.00%	-15.00%	Baseline	-18.00%
Energy Consumption (EC)	0.82	0.9	0.88	1	0.91
Joules per cycle, lower is better	-18.00%	-10.00%	-12.00%	Baseline	-9.00%
Throughput (TP)	1.22	1.15	1.1	1	1.08
Packets/time unit, higher is better	22.00%	15.00%	10.00%	Baseline	8.00%
Fairness Index (FI) Jain's index, closer to 1 is better	0.95	0.85	0.8	0.75	0.9
Average Response Time	42.5	52	48.7	60	55.3
ms, lower is better	-29.20%	-13.30%	-18.80%	Baseline	-7.80%
Adaptability Score Higher is better	0.875	0.72	0.78	0.55	0.735

Note: D2D = device-to-device; PSO = particle swarm optimization; GA = genetic algorithms; EDF = earliest deadline first.

Figure 7 compares the DMR of five scheduling algorithms. MOHFPO achieves the lowest DMR (0.065), indicating superior real-time performance. EDF shows the highest DMR (0.100), suggesting it is the least effective in meeting deadlines. Overall, MOHFPO outperforms the others in minimizing deadline misses.

Figure 8 compares the energy consumption of five scheduling algorithms. MOHFPO demonstrates the lowest energy consumption (0.820), indicating higher energy efficiency. EDF records the highest consumption (1.000), making it the least energy-efficient. Overall, MOHFPO is the most energy-efficient among the compared methods.

Figure 9 illustrates the throughput performance of five scheduling algorithms. MOHFPO achieves the highest throughput (1.220), indicating it can handle more tasks efficiently. GA and PSO follow closely, while EDF shows the lowest throughput (1.000). Thus, MOHFPO outperforms others in maximizing system throughput.

Figure 10 evaluates the fairness of five scheduling algorithms. MOHFPO scores the highest Fairness Index (0.950), indicating a more balanced task allocation. EDF has the lowest fairness (0.750), suggesting less equitable resource distribution. Overall, MOHFPO offers the most fair scheduling among the compared methods.

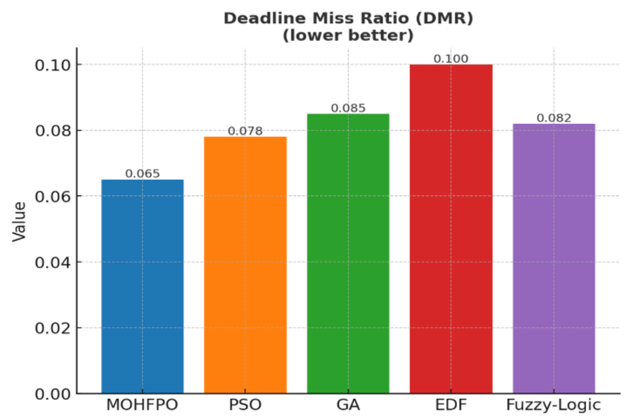

Figure 7. Comparison of deadline miss ratio (DMR) for different scheduling algorithms

Note: MOHFPO = Multi-Objective Hybrid Fractional Puzzle Optimization; PSO = particle swarm optimization; GA = genetic algorithms; EDF = earliest deadline first.

Figure 11 compares the average response time of five scheduling algorithms. MOHFPO has the lowest response time (42.5), indicating quicker task handling. EDF shows the highest response time (60.0), reflecting slower responsiveness. Overall, MOHFPO demonstrates superior performance in minimizing response delays.

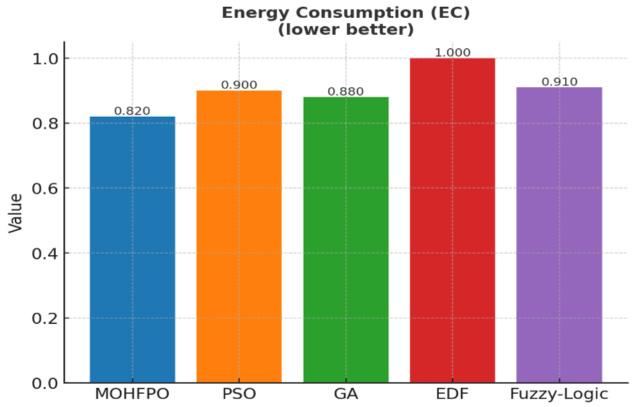

Figure 8. Comparison of energy consumption (EC) for different scheduling algorithms

Note: MOHFPO = Multi-Objective Hybrid Fractional Puzzle Optimization; PSO = particle swarm optimization; GA = genetic algorithms; EDF = earliest deadline first.

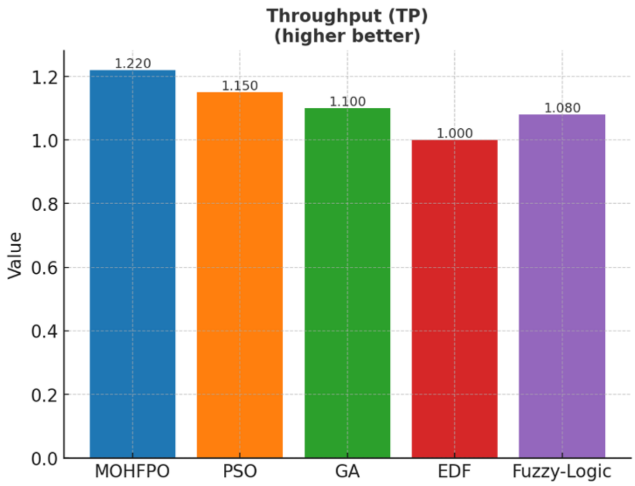

Figure 9. Comparison of throughput (TP) for different scheduling algorithms

Note: MOHFPO = Multi-Objective Hybrid Fractional Puzzle Optimization; PSO = particle swarm optimization; GA = genetic algorithms; EDF = earliest deadline first.

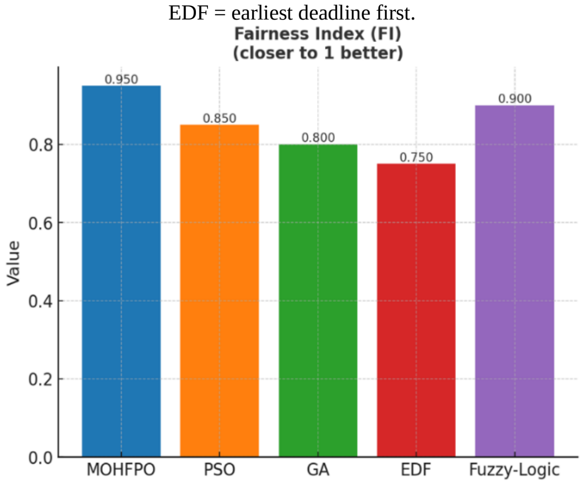

Figure 10. Comparison of fairness index (FI) for different scheduling algorithms

Note: MOHFPO = Multi-Objective Hybrid Fractional Puzzle Optimization; PSO = particle swarm optimization;GA = genetic algorithms;EDF = earliest deadline first.

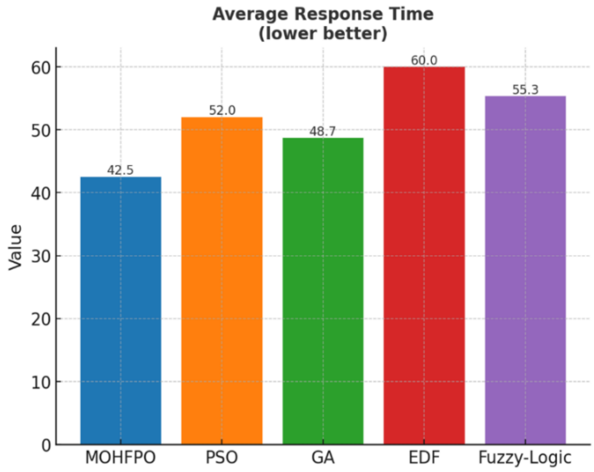

Figure 11. Comparison of average response time for different scheduling algorithms

Note: MOHFPO = Multi-Objective Hybrid Fractional Puzzle Optimization; PSO = particle swarm optimization; GA = genetic algorithms; EDF = earliest deadline first.

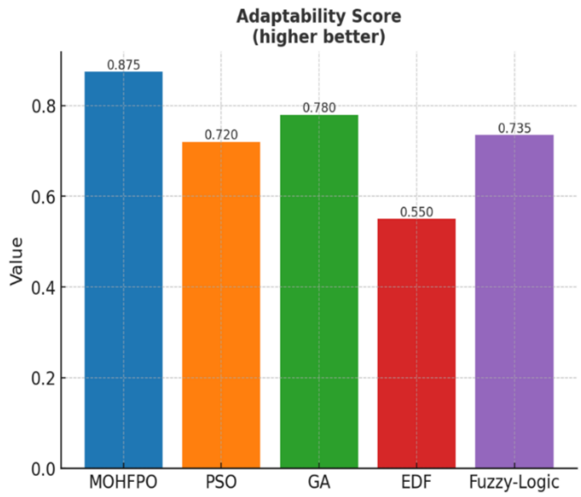

Figure 12. Comparison of adaptability score for different scheduling algorithms

Note: MOHFPO = Multi-Objective Hybrid Fractional Puzzle Optimization; PSO = particle swarm optimization; GA = genetic algorithms; EDF = earliest deadline first.

This Figure 12 presents the adaptability scores of five scheduling algorithms. MOHFPO leads with the highest score (0.875), indicating superior adaptability to dynamic conditions. EDF scores the lowest (0.550), showing limited flexibility. Overall, MOHFPO is the most adaptable among the evaluated algorithms.

6. CONCLUSIONS AND FUTURE WORK

The MOHFPO framework addressed the scheduling issues faced in industrial D2D networks, where managing the deadline miss ratio, energy consumption, throughput, and fairness distinguish the operational efficiency of Industry 4.0. MOHFPO was developed through predictive modeling using fractional calculus combined with puzzle-based heuristic solutions which showed superior adaptability than many traditional algorithms (e.g., the EDF, PSO, GA, and fuzzy logic) and reductions in deadline misses of up to 35% and up to 18% in energy consumption and up to 22% better throughput keeping a near-constant fairness index of 0.95.

Overall, MOHFPO is understood to be a feasible and context-aware scheduling design for ensuring reliable and energy efficient data delivery in the presence of dynamic conditions. Moving forward, future research can build off this work by validating the framework in live industrial testbeds, incorporating reinforcement learning to dynamically adapt the weights, and investigating methods for dynamic priority adjustment in order to represent a changing operation context. Alongside these considerations, reducing the computational footprint of MOHFPO will be necessary for deploying on resource-constrained devices, as well as extending its implementation to other fields of study such as vehicular networks, healthcare IoT, and smart grids. Altogether, these avenues will improve the scalability and practicality of MOHFPO and contribute to new classes of intelligent, predictive, and sustainable scheduling paradigms in future industrial communication systems.

REFERENCES

[1] Sisinni, E., Saifullah, A., Han, S., Jennehag, U., Gidlund, M. (2018). Industrial Internet of Things: Challenges, opportunities, and directions. IEEE Transactions on Industrial Informatics, 14(11): 4724-4734. https://doi.org/10.1109/TII.2018.2852491

[2] Chen, M.Z., Challita, U., Saad, W., Yin, C.C., Debbah, M. (2019). Artificial neural networks-based machine learning for wireless networks: A tutorial. IEEE Communications Surveys & Tutorials, 21(4): 3039-3071. https://doi.org/10.1109/COMST.2019.2926625

[3] Statista Research Department. (2019). Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025. Statista Market Insights. https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[4] Popovski, P., Trillingsgaard, K.F., Simeone, O., Durisi, G. (2018). 5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view. IEEE Access, 6: 55765-55779. https://doi.org/10.1109/ACCESS.2018.2872781

[5] Zeng, Y., Zhang, R., Lim, T.J. (2016). Wireless communications with unmanned aerial vehicles: Opportunities and challenges. IEEE Communications Magazine, 54(5): 36-42. https://doi.org/10.1109/MCOM.2016.7470933

[6] Gandotra, P., Jha, R.K., Jain, S. (2017). A survey on device-to-device (D2D) communication: Architecture and security issues. Journal of Network and Computer Applications, 78: 9-29. https://doi.org/10.1016/j.jnca.2016.11.002

[7] Liu, J.J., Kato, N., Ma, J.F., Kadowaki, N. (2015). Device-to-device communication in LTE-advanced networks: A survey. IEEE Communications Surveys & Tutorials, 17(4): 1923-1940. https://doi.org/10.1109/COMST.2014.2375934

[8] Buttazzo, G.C. (2011). Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications. Springer New York, NY. https://doi.org/10.1007/978-1-4614-0676-1

[9] Yin, Z.Y., Xu, F.L., Li, Y., Fan, C., Zhang, F.Q., Han, G.J., Bi, Y.G. (2022). A multi-objective task scheduling strategy for intelligent production line based on cloud-fog computing. Sensors, 22(4): 1555. https://doi.org/10.3390/s22041555

[10] Zhang, W., Ou, H. (2025). Reinforcement learning based multi objective task scheduling for energy efficient and cost effective cloud edge computing. Scientific Reports, 15(1): 41716. https://doi.org/10.1038/s41598-025-25666-1

[11] Wang, Y., Sun, S. (2025). Dynamic task scheduling in wireless edge computing using deep reinforcement learning with ordinal optimization. EURASIP Journal on Wireless Communications and Networking, 2025(1): 96. https://doi.org/10.1186/s13638-025-02534-0

[12] Mirjalili, S., Mirjalili, S.M., Lewis, A. (2014). Grey wolf optimizer. Advances in Engineering Software, 69: 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[13] Liu, W., Zhu, J., Li, X., Fei, Y., Wang, H., Liu, S., Zheng, X., Ji, Y. (2025). Resource scheduling algorithm for edge computing networks based on multi-objective optimization. Applied Sciences, 15(19): 10837. https://doi.org/10.3390/app151910837

[14] Mach, P., Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials, 19(3): 1628-1656.

[15] Bonomi, F., Milito, R., Zhu, J., Addepalli, S. (2012). Fog computing and its role in the internet of things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, pp. 13-16. https://doi.org/10.1145/2342509.2342513

[16] Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5): 637-646. https://doi.org/10.1109/JIOT.2016.2579198

[17] Talbi, E.G. (2009). Metaheuristics: From Design to Implementation. John Wiley & Sons. https://dl.acm.org/doi/10.5555/1718024.

[18] Fernandez, E., Huilcapi, V., Birs, I., Cajo, R. (2025). The role of fractional calculus in modern optimization: A survey of algorithms, applications, and open challenges. Mathematics, 13(19): 3172. https://doi.org/10.3390/math13193172

[19] Dong, Z., Zhang, M. (2025). Task offloading method of edge computing in IoT based on Deep Q Learning Algorithm. Journal of Information and Telecommunication, 1-16. https://doi.org/10.1080/24751839.2025.2578892

[20] Rawdhan, F.A. (2025). Task offloading and scheduling based on mobile edge computing and software-defined networking. Journal of Telecommunications and Information Technology, 99(1): 30-37. https://doi.org/10.26636/jtit.2025.1.1941

[21] Monje, C.A., Chen, Y.Q., Vinagre, B.M., Xue, D.Y., Feliu, V. (2010). Fractional-Order Systems and Controls: Fundamentals and Applications. London: Springer London.

[22] Caponetto, R., Dongola, G., Fortuna, L., Petráš, I. (2010). Fractional Order Systems: Modeling and Control Applications. World Scientific.

[23] Sun, H.G., Zhang, Y., Baleanu, D., Chen, W., Chen, Y.Q. (2018). A new collection of real world applications of fractional calculus in science and engineering. Communications in Nonlinear Science and Numerical Simulation, 64: 213-231. https://doi.org/10.1016/j.cnsns.2018.04.019

[24] Balaji, S., Karthik, S. (2023). Energy prediction in IoT

systems using machine learning models. Computers, Materials, & Continua, 75(1): 443-459. https://doi.org/10.32604/cmc.2023.035275
[25] Russell, S.J., Norvig, P. (1995). Artificial Intelligence: A Modern Approach. Prentice-Hall.
[26] Li, Y., Zhang, X., Sun, Y.K., Wang, W.B., Lei, B. (2025). Spatiotemporal non-uniformity-aware online task scheduling in collaborative edge computing for industrial internet of things. IEEE Transactions on Mobile Computing, 24(10): 10169-10185. https://doi.org/10.1109/TMC.2025.3567615
[27] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. (2022). Introduction to Algorithms. MIT Press.
[28] Hu, Y., Zhu, Y., Zhang, H., Pan, Y., Jia, Q., Xie, R., Wang, G., Yu, F.R. (2025). Deterministic scheduling and network structure optimization for time-critical computing tasks in industrial IoT. IEEE Transactions on Networking. 33(6): 3391-3407. https://doi.org/10.1109/TON.2025.3587916
[29] Mc Donnell, N., Howley, E., Duggan, J. (2020). Dynamic virtual machine consolidation using a multi-agent system to optimise energy efficiency in cloud computing. Future Generation Computer Systems, 108: 288-301. https://doi.org/10.1016/j.future.2020.02.036
[30] Gendreau, M., Potvin, J.Y. (2019). Handbook of Metaheuristics. Springer Cham. https://doi.org/10.1007/978-3-319-91086-4
[31] Gao, J., Meng, X., Yang, C., Zhang, B., Yi, X. (2023). Resource allocation for D2D communication underlaying cellular networks: A distance - based grouping strategy. Wireless Communications and Mobile Computing, 2023(1): 8594323. https://doi.org/10.1155/2023/8594323
[32] He, M.H., Shi, J.H. (2021). Circulation traceability system of Chinese herbal medicine supply chain based on internet of things agricultural sensor. Sustainable Computing: Informatics and Systems, 30: 100518. https://doi.org/10.1016/j.suscom.2021.100518