Optimization of Task Offloading and Resource Allocation in the Internet of Everything: A Comprehensive Survey

Dayong Wang [1,*], Liping Lei [2], Zhen Wang [1,3], and Cui Cui [4]

[1] Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia
[2] School of Political Science and Public Administration, East China University of Political Science and Law, Shanghai China
[3] School of Data Science and Artificial Intelligence, Wenzhou University of Technology, Wenzhou, China
[4] Department of Business Engineering and Technology, Shandong College of Economics and Business, Weifang, China
Email: dayong.wang.cs@outlook.com (D.W.); leiliping@ecupl.edu.cn (L.L.); zhenwang@graduate.utm.my (Z.W.); cnsdtracy@gmail.com (C.C.)
*Corresponding author

***Abstract*—The Internet of Everything (IoE) enables ubiquitous connectivity among heterogeneous devices but poses significant challenges for task offloading and resource allocation due to high dynamics and strict resource constraints. To address these complexities, optimization plays a pivotal role in efficiently orchestrating these resources. This survey comprehensively reviews optimization methodologies for IoE task offloading. Methodologically, this study relies on a comprehensive literature synthesis to distill evolutionary algorithmic trends and evaluate their applicability to specific IoE scenario constraints. The review is structured around six major paradigms: classic, Lyapunov-based, heuristic, game-theoretic, AI-native, and hybrid optimization. For each, representative techniques, assumptions, and performance characteristics are summarized. Comparative analysis highlights the evolution from model-based formulations toward intelligent, adaptive, and collaborative frameworks. Key challenges are identified, including scalability bottlenecks, learning instability in non-stationary environments, and complex multi-objective trade-offs. Finally, emerging research directions are outlined. This work bridges theoretical foundations with intelligent methodologies, offering a unified perspective for advancing cognitive and sustainable IoE ecosystems.**

***Keywords*—adaptive scheduling, decision making, internet of everything, resource allocation, scheduling optimization, task offloading**

I. Introduction

With rapid advances in Internet of Things (IoT) technology, the Internet has evolved from merely connecting devices to enabling comprehensive interconnections among people, objects, data, and processes, a trend known as the Internet of Everything (IoE) [1, 2]. IoE encompasses a vast array of smart terminals and diverse application scenarios, including sensor networks, smart homes, autonomous driving systems, and the industrial Internet [3, 4]. The explosive growth of devices and data poses core challenges for IoE system design, requiring efficient data processing and transmission while ensuring real-time performance, reliability, and security [5, 6]. In particular, with the coordinated development of edge and cloud computing, IoE systems are increasingly evolving toward distributed intelligence.

In such IoE scenarios, large volumes of data and computational tasks flow among end devices, edge nodes, and cloud servers, making task offloading a key approach for performance optimization [2, 7]. Task offloading alleviates the computational burden on resource-constrained end devices by migrating computation-intensive or latency-sensitive tasks to edge servers or the cloud, thereby improving overall system responsiveness and energy efficiency. In recent years, task offloading techniques have evolved from initial static-rule-based strategies to dynamic decision-making methods that consider task characteristics, network conditions, and resource constraints [6–9]. In IoE environments, task offloading must not only optimize the performance of individual devices but also achieve efficient coordination across massive heterogeneous devices and multi-layer computing platforms, which introduces significant complexity and research challenges [5, 10, 11] .

However, the performance of task offloading heavily depends on the effectiveness of scheduling and resource allocation strategies [12]. In IoE systems, computing resources, storage, and network bandwidth are all limited, making optimized scheduling across multiple devices, nodes, and tasks crucial for improving offloading efficiency. Scheduling and resource allocation optimization must consider task execution order and priority while also accounting for network latency, energy consumption, and system load [8, 13]. In recent

Manuscript received November 06, 2025; revised December 20, 2025; accepted December 31, 2025; published March 26, 2026.

years, numerous task offloading optimization methods for IoE have emerged, including joint scheduling strategies based on heuristic algorithms, convex optimization, reinforcement learning, and deep learning, providing effective means to enhance overall system performance [14−16].

Despite numerous research efforts, existing task offloading optimization methods still face several challenges. First, the dynamic and heterogeneous nature of IoE environments makes traditional static optimization models inadequate for adapting to real-time changes in network and computing conditions [17, 18]. Second, scheduling across large-scale devices and tasks is typically Nondeterministic Polynomial-time hard (NP-hard), and current approximation algorithms involve trade-offs between computational complexity and optimization effectiveness [19, 20]. Finally, in practical applications, task offloading must also consider security, privacy protection, and energy constraints, further increasing the complexity of designing optimization strategies. Therefore, achieving efficient, scalable, and secure task offloading and resource management while ensuring performance improvement remains a core challenge in IoE research.

There have been numerous survey papers on task offloading and resource optimization [1, 2, 7, 12, 14]. However, existing studies still exhibit significant limitations. Most surveys focus on a single network environment (such as 6G networks) or a single computing paradigm (such as mobile edge offloading), lacking a comprehensive analysis of task offloading issues in the IoE with its multi-layered and heterogeneous network environments. In addition, discussions on heterogeneous devices, dynamic network variations, and multi-task cooperative scheduling remain limited, making it difficult to comprehensively capture the real-world requirements of complex task offloading in IoE scenarios. Therefore, it is necessary to conduct a comprehensive survey that covers diverse network environments, computing paradigms, and optimization methods, in order to comprehensively review existing research achievements, identify research gaps, and provide insights for the design and optimization of future IoE task offloading. Specifically, the main contributions of this study are summarized as follows:

- This paper comprehensively reviews studies from 2020 onward on task offloading and resource allocation optimization in IoE and discusses the strengths and limitations of different implementation approaches.
- Highlights the challenge of jointly optimizing scheduling and resource allocation in task offloading.
- Compares and clarifies the suitability of different optimization schemes across diverse application scenarios.
- The open issues are discussed and highlighted.
- Potential research opportunities and future research directions are also identified.

To enhance the readability and clarity of this paper, Table I provides the abbreviations employed in the text together with their full expressions.

TABLE I. LIST OF ABBREVIATIONS USED IN THIS PAPER

Abbreviations	Full description
A3C	Asynchronous Advantage Actor-Critic
AADMM	Adaptive Alternating Direction Method of Multipliers
AFS	Artificial Fish Swarm
AO	Alternating Optimization
BCD	Block Coordinate Descent
CHROA	Chaotic Horse Ride Optimization Algorithm
D2D	Device-to-Device
DDPG	Deep Deterministic Policy Gradient
DQN	Deep Q-Learning Network
DRL	Deep Reinforcement Learning
EDF	Earliest Deadline First
EHEC_SD3	Energy Harvesting Edge Computing with Softmax Deep Double Deterministic
ESS	Evolutionarily Stable Strategy
F-DRL	Federated Reinforcement Deep Learning
GAT	Graph Attention networks
GNN	Graph Neural Networks
GWO	Gray Wolf Optimization
HJB	Hamilton–Jacobi–Bellman
IoE	Internet of Everything
IoT	Internet of Things
LEO	Low Earth Orbit
LRU	Least Recently Used
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
MA-DRL	Multi-Agent Deep Reinforcement Learning
MBS	Multidimensional Bisection Searching
MEC	Mobile Edge Computing
ML	Machine Learning
MLR	Multivariate Linear Regression
MR	Max Rate
NP-hard	Nondeterministic Polynomial-time hard
NSGA	Non-dominated Sorting Genetic Algorithm
PB	Priority Base
PF	Proportional Fair
QoS	Quality of Service
RR	Round Robin
SA	Simulated Annealing
SAGIN	Space–Air–Ground Integrated Networks
SC	Strongest Channel
SOM	Self-Organizing Maps
SVM	Support Vector Machines
UAV	Unmanned Aerial Vehicle
UE	User equipment
WEC	Wireless Edge Computing
WF-CSA	Wolf Fish Collaborative Search Algorithm
WPMEC	Wireless Powered Multi-access Edge Computing

The rest of this paper is organized as follows: Section II reviews the related work. Section III introduces the foundations of task offloading and resource allocation in IoE. Section IV discusses the optimization paradigms and taxonomy. Section V provides the analysis and discussion. Finally, Section VI concludes the paper.

II. RELATED WORKS

The Internet of Everything (IoE) and computation offloading have experienced rapid advancements in recent years. This section provides an overview of representative studies and survey papers that have contributed to shaping the current landscape of this research area.

Jamil *et al.* [21] reviewed task offloading optimization schemes in 6G network environments, particularly discussing the delay–energy–carbon tri-objective Pareto frontier under security constraints, and proposed a four-layer protection architecture integrating 6G, security, and edge computing concepts. An AI-RAN online defense framework was also presented. However, the survey focused solely on the 6G network environment.

Kong *et al.* [2] investigated IoE technologies enabled by edge computing and proposed a unified definition for edge-based IoE. The work further deconstructed and comparatively analyzed service placement versus task offloading, also addressing service placement and migration optimization techniques. Nevertheless, it was limited to edge computing scenarios.

Jamil *et al.* [7] conducted a systematic, comprehensive, and detailed comparative study by discussing various scheduling algorithms, key optimization metrics, and evaluation tools in fog computing and IoE environments. Similarly, Nagabushnam *et al.* [14] reviewed IoE optimization methods focused on task scheduling and energy consumption, rigorously evaluating each algorithm in terms of objectives, strategies, and energy management capabilities, emphasizing energy consumption, computation latency, task completion time, and quality of service. However, both studies were restricted to fog computing scenarios.

Souza *et al.* [12] applied a rapid review method to analyze the collaboration among four popular computing architectures in the IoE paradigm, namely edge computing, cloud computing, block-chain/network services, and fog computing. This work comparatively analyzed the functionalities and limitations of the four computing architectures, highlighting the importance of edge and cloud computing for enhancing coordination, efficiency, and network optimization. Nonetheless, it lacked an analysis and comparison of task scheduling and resource allocation optimization methods.

Mika and Juha [22] surveyed IoE-related technologies in the context of smart cities, covering multiple computing paradigms including cloud, edge, and fog. However, it did not specifically focus on task offloading.

Based on the preceding discussion, two major shortcomings in the existing research have been identified, as summarized in Table II. The first is that previous surveys have primarily concentrated on specific network communication environments. Furthermore, although extensive research has been conducted on task offloading and resource allocation optimization under the paradigms of edge, fog, and cloud computing, a unified and comprehensive survey covering diverse computing paradigms is still missing.

These observations motivated this work to provide a comprehensive survey on task offloading and resource allocation optimization in IoE, aiming to bridge the identified research gaps. The primary added value of this survey lies in its comprehensive integration of heterogeneous computing paradigms within the IoE context, moving beyond the siloed scope of previous investigations. We extend the existing body of knowledge by categorizing recent algorithmic breakthroughs and analyzing their applicability in dynamic, multi-tier network environments. Readers will benefit from a critical analysis of state-of-the-art solution frameworks that address the complexity of modern IoE tasks, providing a clear roadmap for selecting appropriate offloading strategies under varying resource constraints and latency requirements.

TABLE II. A COMPARISON WITH RELEVANT SURVEYS

Work	Year	Contribution	Limitation
[21]	2021	Secure IoE task scheduling and resource allocation	Limited to 6G network environments
[7]	2022	Compared IoE task scheduling algorithms, optimization metrics, and evaluation tools	Limited to Fog computing architectures
[2]	2022	In-depth comparative analysis of task offloading, scheduling, and resource allocation schemes	Limited to Edge computing architectures
[12]	2024	Covered four popular computing architectures in IoE paradigms	Lacked analysis of task scheduling and resource allocation optimization
[22]	2025	Studied applications, implementation challenges, and widely adopted roadmaps	Lacked focus on task offloading
[14]	2025	Compared each algorithm's objectives, strategies, and energy management capabilities	Limited to Fog computing architectures

III. RESEARCH METHODOLOGY

This section outlines the literature review methodology employed in this study. Clear search strategies and inclusion criteria are established to ensure comprehensive coverage and reduce selection bias.

To guide the data extraction and synthesis process effectively, this study formulates a set of specific research questions that address the core technical dimensions of task offloading and resource allocation.

- RQ1: What is the comparative research emphasis across the specific problem domains of task offloading decision-making, resource allocation, and their joint optimization?
- RQ2: What are the dominant optimization techniques and algorithmic paradigms currently adopted in the literature?
- RQ3: Which system architectures and computing frameworks (e.g., MEC, Cloud-Edge Continuum) are primarily supported and modeled in existing studies?
- RQ4: What are the dominant optimization objectives predominantly targeted in the current literature?
- RQ5: How do distinct optimization paradigms align with and adapt to the diverse and heterogeneous application scenarios of IoE?
- RQ6: What are the critical open issues and technical limitations inherent in current optimization solutions?
- RQ7: What are the emerging trends and high-potential directions for future research in the field of IoE edge intelligence?

The literature search was primarily conducted using major academic databases, including ISI Web of Science, Scopus, IEEE Xplore, and ScienceDirect. In addition, keyword searches were also performed in Google Scholar to reduce the risk of missing relevant studies. The keywords used in this review are summarized in Table III.

TABLE III. KEYWORDS

Keyword	Intention
Internet of everything	Scoping the Internet of Everything domain
IoE	To avoid omissions resulting from the use of abbreviations
Internet of things	Gathering hybrid task offloading studies across IoE and IoT domains
IoT	To avoid omissions resulting from the use of abbreviations
Task offload	Defining the scope of the task offloading domain
Scheduling	Collecting studies on task-offloading optimization
Resource Allocation	Gathering studies on resource allocation for task-offloading
Optimization	Collecting studies on task-offloading optimization

This study restricts the literature review to the period from January 2020 to October 2025. Task offloading and resource allocation techniques for IoE networks prior to 2020 have already been extensively discussed in previous survey studies [2, 7, 21]. Moreover, the year 2020 marked the large-scale commercialization of 5G communication technologies and the initial establishment of the 6G vision, which collectively provided IoE networks with unprecedented ultra-low latency and massive connectivity capabilities. These advances have fundamentally reshaped the constraint conditions and network topologies underlying task offloading in IoE systems [23−25].

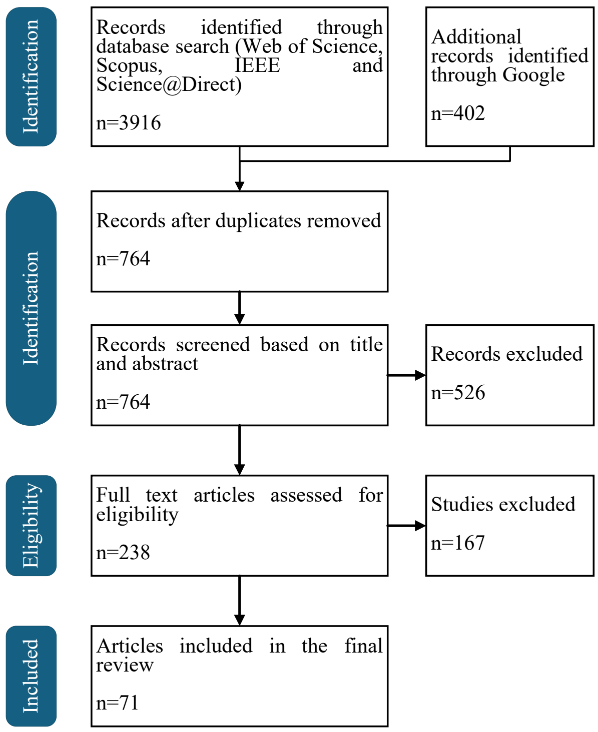

Fig. 1. Flowchart of the study selection process.

To maintain a focused scope, only studies that specifically investigate task offloading scheduling and resource allocation optimization in IoE scenarios were selected. In addition, publications addressing both IoE and IoT contexts were carefully examined, and only those primarily oriented toward IoE were included in this study. Fig. 1 illustrates the literature collection and screening process adopted in this work.

IV. FOUNDATIONS OF TASK OFFLOADING AND RESOURCE ALLOCATION IN IOE

This chapter introduces the fundamental concepts and methodologies related to task offloading and resource management in the IoE. It first presents the basic concepts, architectural characteristics, and key challenges of the IoE. Then, it analyzes the classification and decision-making approaches of task offloading technologies and their applications in the IoE. Next, it discusses task scheduling optimization, including task execution order, prioritization, and cooperative scheduling strategies. Finally, it presents resource allocation methods, with a focus on the efficient allocation and scheduling of computing, storage, and network resources. Together, these discussions establish a theoretical foundation for understanding and designing efficient task offloading schemes in the IoE.

A. Internet of Everything

The IoE is an evolution and extension of the traditional IoT [26−28]. It not only emphasizes interconnection among devices but also focuses on the comprehensive integration of people, things, data, and processes, as illustrated in Fig. 2. The computing system of the IoE typically consists of four layers: the user interaction layer, the network layer, the edge computing layer, and the cloud computing layer. Among them, User Equipment (UE), edge nodes, and cloud servers together form the foundational framework of a distributed computing network.

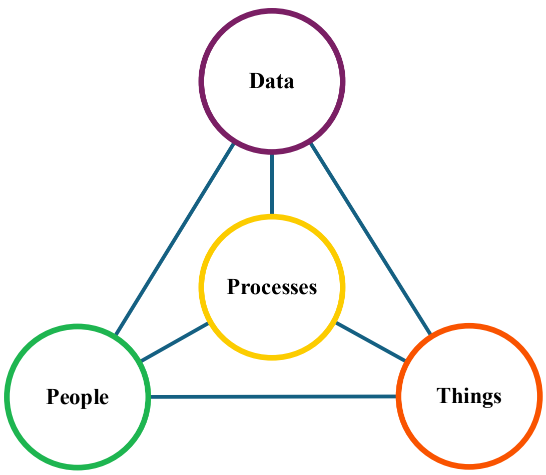

Fig. 2. The four-pillar conceptual framework of the IoE.

In the IoE environment, a wide variety of devices are involved, including smartphones, wearable devices, industrial sensors, autonomous vehicles, and various embedded systems [29, 30]. These devices differ significantly in computing power, storage capacity, communication capabilities, and energy constraints [31, 32]. Meanwhile, the complexity and diversity of IoE application scenarios impose stringent requirements on real-time performance and reliability. Applications such as intelligent transportation, industrial automation, telemedicine, and environmental monitoring demand high data processing speed and rapid system response.

Furthermore, with the massive scale of IoE devices and services, the volume of data grows exponentially, requiring the system to achieve efficient collaboration in a distributed and multi-layered environment [33, 34]. Therefore, the IoE architecture not only emphasizes interconnectivity and information sharing but also must ensure scalability, dynamic adaptability, and robust security and privacy protection [35, 36]. These characteristics collectively pose significant challenges for task offloading and resource management in IoE systems.

B. *Task Offloading*

Task offloading is a key technology in IoE systems that helps alleviate the computational burden on UEs while enhancing system responsiveness and energy efficiency, as shown in Fig. 3. Its core concept is to migrate computation-intensive or latency-sensitive tasks from resource-constrained UEs to more powerful edge nodes or cloud servers for execution [37, 38]. In the IoE environment, task offloading is not a simple migration process. It requires comprehensive consideration of task types, computational workloads, data dependencies, network bandwidth, and the load conditions of both UEs and edge nodes.

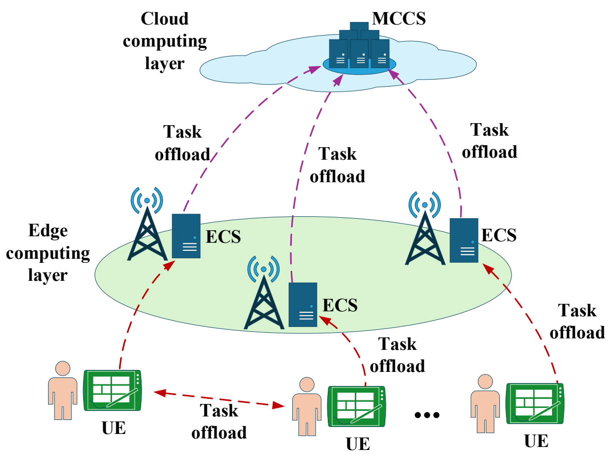

Fig. 3. Schematic diagram of task offloading in IoE.

As the scale of IoE systems expands and application scenarios become increasingly diverse, task offloading strategies have evolved from early static rule-based methods to dynamic and intelligent decision-making mechanisms. For example, partial offloading strategies divide tasks into multiple subtasks, allowing some computations to be executed locally while others are offloaded to the edge or cloud. This approach helps reduce energy consumption while maintaining fast response times. Dynamic offloading strategies can adjust offloading decisions in real time according to network conditions, available resources, and task priorities, enabling the system to adapt to environmental uncertainty and variability.

Through well-designed task offloading strategies, the processing capability of UEs and the overall system throughput can be significantly improved, while data transmission delay and total energy consumption are reduced. These improvements provide a strong foundation for delivering high-performance services in the IoE.

C. *Task Scheduling*

In the process of task offloading, task scheduling serves as the core mechanism for achieving efficient computation and optimal resource utilization. The primary objective of scheduling is to determine the execution order, allocation, and timing of tasks across different computing nodes in order to maximize system performance and meet diverse service requirements [10, 39].

As presented in Fig. 4, computational tasks can be offloaded not only among UEs but also to the edge computing network for faster processing. When edge computing resources become insufficient, tasks can be further offloaded to the fog or cloud computing layers for execution. This hierarchical offloading process increases the complexity of task scheduling and decision-making in IoE systems. Therefore, task scheduling in the IoE faces significant challenges due to the heterogeneity of devices and computing capabilities, dynamic network conditions, and the large number of tasks with uncertain arrival times.

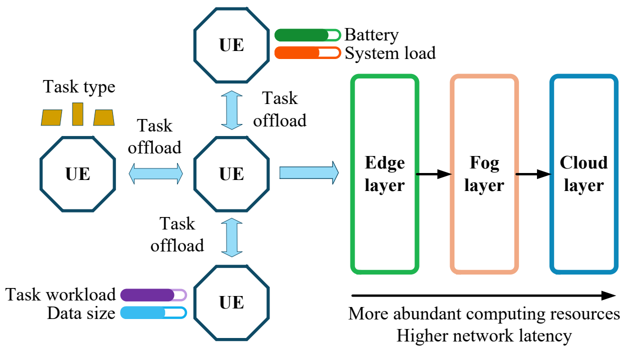

Fig. 4. Scheduling in task offloading.

Against this background, scheduling strategies have evolved from static scheduling to dynamic scheduling. Static scheduling plans task allocation and execution order in advance, which is suitable for scenarios with relatively stable workloads. Dynamic scheduling, on the other hand, can respond in real time to changes in network conditions, resource availability, and task priorities, enabling the system to maintain high efficiency in complex environments.

In this line of research, tasks are typically modeled as computation jobs characterized by their workload and deadline constraints, as in Eq. (1).

$$T_i = \left\{ D_i, C_i, \tau_i^{\max} \right\} \tag{1}$$

where D_i denotes the size of the data associated with the task, C_i represents the amount of computation required to complete the task, and $\tau_i^{\max}$ specifies the maximum tolerable latency of the task.

For tasks that are not offloaded and instead executed locally, their computation time and energy consumption are typically modeled as shown in Eqs. (2)–(3).

$$t_i^{loc} = \frac{C_i}{f_i^{loc}} \tag{2}$$

$$e_i^{loc} = k\left(f_i^{loc}\right)^2 C_i \tag{3}$$

where f_i^{loc} denotes the local CPU frequency, and k is the energy coefficient dependent on the chip architecture.

Conversely, if the task is offloaded to a remote node for execution, its execution time and energy consumption are typically modeled as shown in Eqs. (4) and (5).

$$t_i^{rem} = \frac{C_i}{f_i^{rem}} + t_i^{comm} \tag{4}$$

$$e_i^{rem} = q\left(f_i^{rem}\right)^2 C_i + e_i^{comm} \tag{5}$$

where f_i^{rem} denotes the CPU frequency allocated to the task by the target computing node, and q is the energy consumption coefficient of the target node, which depends on its chip architecture.

The task offloading decision model can be expressed as shown in Eq. (6).

$$x_i = \begin{cases} 0 & \text{Local execution} \\ 1 & \text{Remote execution} \end{cases} \tag{6}$$

where x_i equals 0, it indicates that the task is executed locally, and when x_i equals 1, it indicates that the task is offloaded to a remote node for execution. Therefore, the multi-objective cost function for task offloading can be expressed as follows:

$$\begin{aligned} J_i = {} & x_i \times \left(\alpha_i t_i^{rem} + \beta_i e_i^{rem}\right) \\ & + \left(1 - x_i\right) \times \left(\alpha_i t_i^{loc} + \beta_i e_i^{loc}\right) \end{aligned} \tag{7}$$

where $\alpha_i, \beta_i, \in [0, 1]$ denote the user's preference weights for delay and energy consumption, respectively, and $\alpha_i + \beta_i = 1$.

D. Resource Allocation

Resource allocation forms the foundation for the efficient implementation of task offloading, and its core lies in the rational distribution of computing, storage, and network resources among UEs, edge nodes, and cloud servers, as shown in Fig. 5. In the IoE environment, resource allocation must address highly heterogeneous computing platforms, limited bandwidth, and dynamically changing task demands. Different nodes vary significantly in computing power, storage capacity, and network conditions, while a large number of tasks often compete for limited resources at the same time [40, 41]. Therefore, the resource allocation mechanism must not only improve overall system performance but also ensure the prioritized execution of critical tasks.

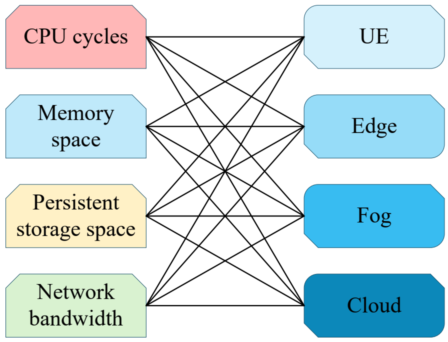

Fig. 5. Resource allocation in task offloading.

To address these challenges, researchers have proposed various resource allocation strategies. Optimization-based approaches build mathematical models to achieve global optimization of resource utilization. Heuristic and meta-heuristic algorithms provide a balance between efficiency and performance through approximate solutions. More recently, learning-based methods have emerged, where intelligent models are trained using historical data to make rapid resource allocation decisions in dynamic environments.

The allocation of computing and communication resources directly affects task latency, as shown in Eq. (4). Therefore, the joint optimization objective for task-offloading scheduling and resource allocation can be formulated as presented in Eq. (8).

$$\begin{aligned} \min \ & \left(1 - x_i\right) \times t^{loc} + x_i \times \left(\frac{D_i}{R_i} + \frac{C_i}{f_i^{rem}}\right) \\ \text{s.t.} \ & \sum\left(x_i \times M_i\right) \le M_{rem} \\ & \sum\left(x_i \times S_i\right) \le S_{rem} \end{aligned} \tag{8}$$

where R_i denotes the bandwidth allocated for transmitting the task data. M_i and S_i represent the memory and persistent storage required for executing the task, respectively, while M_{rem} and S_{rem} denote the available memory capacity and persistent storage capacity at the remote computing node.

Overall, in the IoE environment, task offloading not only involves making decisions about migrating computational tasks from UEs to the edge or cloud but also requires addressing multiple challenges arising from scheduling strategies and resource allocation optimization. The presence of complex heterogeneous devices, dynamically changing network conditions, and concurrent task execution leads to a high degree of coupling among task offloading, scheduling, and resource management. Optimization in a single dimension is often insufficient to meet the comprehensive requirements of system performance, energy efficiency, and latency.

Against this background, designing efficient offloading strategies, intelligent scheduling methods, and effective resource allocation mechanisms has become a central focus of IoE research. The following sections will systematically explore specific optimization approaches and recent research advances related to these challenges, providing insights for improving the overall performance of task offloading in the IoE.

V. Optimization Paradigms and Taxonomy

In the IoE environment, the performance improvement of task offloading depends not only on the offloading decisions themselves but also on the optimization of scheduling strategies and resource allocation methods. Due to the heterogeneity and multi-layered structure of IoE systems, as well as the dynamic nature of tasks and network conditions, a single optimization approach is often insufficient to meet the comprehensive requirements of multiple performance metrics such as latency, energy consumption, throughput, and resource utilization.

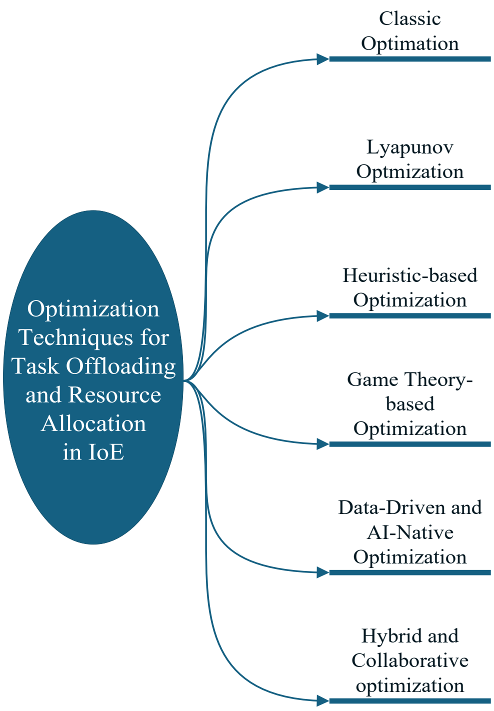

Fig. 6. Taxonomy of optimization techniques for task offloading and resource allocation in IoE.

To address these challenges, researchers have proposed various optimization paradigms tailored to different scenarios and application needs. These include classical optimization, Lyapunov-based methods, heuristic algorithms, game-theoretic models, data-driven and artificial intelligence approaches, as well as hybrid optimization strategies that combine multiple paradigms, as illustrated in Fig. 6. Each paradigm has its own theoretical foundation, applicable scenarios, and performance characteristics, along with distinct advantages and limitations.

The following sections will systematically review the principles, design concepts, and applications of these optimization methods in IoE task offloading according to their technical paths, providing a comprehensive theoretical framework for understanding and designing efficient offloading strategies.

A. *Classic Optimation*

Classical optimization methods serve as the fundamental paradigm for research on task offloading and resource allocation in the IoE. These methods typically rely on deterministic mathematical modeling and analytical solutions, such as convex optimization, linear programming, or greedy algorithms, to minimize latency, energy consumption, or system cost while achieving efficient resource allocation. In recent years, such approaches have been widely applied across various IoE architectures. Examples include task scheduling based on multivariate linear regression and greedy algorithms, joint optimization of energy consumption and latency using convex and alternating optimization, and scheduling strategies based on task priority or Earliest Deadline First (EDF) policies. Table IV summarizes representative studies in recent years that are based on classical optimization methods.

In Ref. [42], a task scheduling strategy was developed based on a Multivariate Linear Regression (MLR) model, where a linear mapping between task attributes and edge server load was established to jointly optimize task processing delay and edge node workload. This type of method features a clear structure and low computational complexity, providing stable performance in environments with fixed network topologies and consistent task patterns. However, its main limitation lies in the reliance on static parameter assumptions, which prevents it from effectively capturing dynamic changes in network topology and task characteristics.

In a similar vein, Manogaran and Rawal [43] modelled the joint resource allocation problem in a fog computing scenario using convex optimization. The proposed approach achieved near-optimal performance comparable to exhaustive search but with significantly lower computational complexity, demonstrating the global convergence and high precision of convex optimization techniques. Such research highlights the mathematical structure of optimization problems in IoE systems, though its applicability is often limited by the convexity assumptions required for modelling. In IoE environments, communication link fluctuations, node heterogeneity, and stochastic task arrivals tend to increase model nonconvexity, making it challenging for traditional analytical methods to maintain both accuracy and convergence efficiency.

A resource allocation method based on Multidimensional Bisection Searching (MBS) was proposed to improve service freshness [44]. This study targets IoE application scenarios that demand high data timeliness. By iteratively searching for the optimal allocation point within a multidimensional resource space, the method dynamically adjusts task processing and transmission rates to maintain the timeliness of service results. Unlike traditional models that focus on latency or energy consumption, this approach treats information freshness as the core performance metric and theoretically establishes a quantitative relationship between task update intervals and resource allocation. Experimental results demonstrate that this method effectively reduces data expiration rates and enhances service response consistency. Its main advantage lies in achieving fast approximate solutions with limited computational resources, offering strong real-time performance and implementation efficiency. However, the method overlooks latency minimization in its modelling process and does not fully account for the trade-off between freshness and delay, which may lead to performance degradation under heavy-load or task-intensive conditions.

To achieve a practical balance between responsiveness and optimality, greedy and priority-based scheduling strategies have been widely applied in IoE systems, particularly in real-time task scenarios that are resource-constrained and latency-sensitive. In a cloud-fog collaborative architecture, a greedy algorithm was initially employed for resource allocation to reduce communication delay and cost [45]; this approach was subsequently enhanced by incorporating a task priority mechanism to ensure resources for critical tasks, thereby improving latency and responsiveness [46]. These methods offer the advantages of simplicity, rapid response, and suitability for online deployment, and they can generate near-optimal decisions efficiently, especially on resource-limited edge nodes.

However, their limitations are also evident. First, greedy strategies inherently lack a global perspective and can only guarantee local optimality. Second, task priorities are usually statically defined, making them difficult to adjust dynamically according to environmental conditions, which can lead to performance instability under heavy workloads or high node contention. In addition, Song *et al.* [47] and Picano and Fantacci [48] adopted scheduling algorithms based on task priority and Earliest Deadline First (EDF) strategies, achieving good performance in terms of fairness and deadline constraints. Nevertheless, these approaches remain constrained by static task models and fail to fully consider task coupling and cross-layer resource competition. Lakhan *et al.* [49] and Salim *et al.* [50] further extended the greedy concept by introducing blockchain and energy-aware mechanisms, enabling resource scheduling to incorporate both security and energy efficiency. However, their performance tends to degrade and computational overhead increases in highly dynamic networks. Overall, greedy and priority-based algorithms offer significant advantages in engineering implementation and real-time responsiveness. Yet, in complex and dynamic environments, their reliance on local optimality and limited scalability continues to pose major challenges.

As the complexity of IoE systems continues to grow, researchers have increasingly adopted more systematic numerical optimization frameworks to address nonlinear and coupled resource allocation problems. An Alternating Optimization (AO) method was first employed in MEC environments to balance spectrum and energy efficiency by decomposing the high-dimensional joint channel and power allocation problem into solvable sub-problems [51]; this convex optimization framework was later extended to UAV-assisted MEC scenarios for the joint optimization of trajectory, altitude, and task scheduling to minimize completion time [18]. These studies demonstrate that when the problem structure exhibits convexity or decomposability, AO and convex programming methods can achieve an effective trade-off between global performance and computational feasibility. However, these methods typically rely on accurate system state and channel information, making them highly sensitive to parameter errors and model uncertainties. Consequently, their stability may degrade in highly dynamic IoE networks. Moreover, the computational overhead associated with online optimization remains considerable, posing challenges for real-time deployment in latency-sensitive IoE applications.

A joint optimization approach integrating greedy decision-making with the Best-Fit mechanism dynamically adjusts node computing capacities to reduce deadline violations and average latency, achieving a balance between local responsiveness and global resource utilization [52]. Building on this concept, a hybrid optimization framework combining Block Coordinate Descent (BCD) and Simulated Annealing (SA) was developed to exploit the convergence stability of BCD and the global search capability of SA [53]. This hybrid design embodies the principle of deterministic convergence and stochastic exploration, showing superior robustness and global optimization performance in non-convex IoE offloading problems. However, these methods suffer from high computational complexity and parameter sensitivity, as performance depends heavily on initialization and annealing settings. With increasing system scale and node density, iterative computation and synchronization overheads rise sharply, limiting their real-time applicability. Future research may integrate hybrid optimization with distributed computation, hierarchical control, or reinforcement learning to enhance scalability.

To further improve system coordination, an Adaptive ADMM-based scheduling algorithm was proposed to minimize overall system cost in cloud–edge collaborative frameworks [54]. By introducing adaptive penalty adjustments that respond to network dynamics, this method accelerates convergence and reduces migration and energy costs, demonstrating strong suitability for heterogeneous IoE environments. Nonetheless, the assumption of fixed user mobility overlooks its influence on bandwidth and task migration, which may affect performance under highly dynamic conditions.

From an application perspective, the applicability of classical optimization methods differs considerably across various IoE computing architectures. In cloud–fog–edge collaborative frameworks (e.g., [43, 45, 46, 49, 50, 52, 55]), research primarily focuses on cross-layer resource allocation and task offloading strategies, with latency minimization and energy efficiency as the main objectives. These approaches typically employ greedy or convex optimization techniques to achieve hierarchical resource allocation, thereby alleviating cloud overload and improving overall system utilization. However, the heterogeneity of cloud–fog–edge systems introduce challenges such as inconsistent cross-domain parameters, asynchronous resource states, and incomplete information, which make centralized optimization approaches difficult to apply directly.

TABLE IV. CLASSIC OPTIMIZATION

Work	Year	Problems addressed	Utilized technique	Optimization objective	Architecture	Advantages	Limitations
[42]	2021	Task scheduling	Multivariate Linear Regression	Task Processing Delay and ECS Load Balancing Degree	Edge	Low computational complexity.	Not support dynamic network topology changes.
[45]	2021	Resource allocation	Greedy	Reduce communication latency, lower cost	Cloud-Fog Continuum	High infrastructure resource utilization	Scalability issues
[46]	2021	Task scheduling and Resource allocation	Priority-driven	Decrease service delay	Cloud-Edge Continuum	Critical task resources ensured	Greedy approaches find it difficult to reach globally optimal solutions
[51]	2023	Resource allocation	Alternating optimization	Improve spectrum efficiency & energy efficiency	MEC	Support maximizing computation rate under different DIBF schemes	Impact of task failures due to timeout not considered
[47]	2023	Task scheduling	Based on task priority	Improve QoE (Quality of Experience)	MEC	High fairness	Not support UE differentiation
[43]	2023	Resource allocation	Convex optimization	Reduce latency and energy consumption	Fog	Low-complexity joint design very close to the exhaustive optimum	Only supports fixed computation offloading scheme
[48]	2023	Resource allocation	Earliest Deadline First (EDF)	Increase ability to complete services before deadline	MEC	High reliability of service completion before deadline	EDF method difficult to achieve global optimization
[44]	2023	Resource allocation	Multidimensional bisection searching	Improve service freshness	Custom	Service freshness guarantee	Latency-reduction optimization ignored
[52]	2023	Task scheduling and Resource allocation	Greedy decision and Best-fit computing power adjustment	Reduce deadline violations and average task duration	Cloud-Edge Continuum	Minimized the communication distance and latency between the UE and the computing node	Lack of global optimization capability
[49]	2024	Task scheduling	Greedy	Reduce processing time and lower power consumption	Cloud-Edge Continuum	Supports blockchain-based data transmission	Ignores resource failures and invalid application registrations in the network.
[18]	2024	Task scheduling	Convex Optimization	Reduce task completion time	UAV-MEC	Jointly schedules task allocation, UAV flight trajectory, and hovering altitude	Ignores the UAV battery capacity constraint
[50]	2024	Task scheduling	Greedy	Reduce energy consumption and avoid task failures	Cloud-Edge Continuum	Supports differentiated task priority requirements.	Lacks support for dynamic changes in the network environment.
[53]	2024	Task scheduling and Resource allocation	Block coordinate descent (BCD) and Simulated annealing (SA)	Reduce task completion latency and energy consumption	MEC	In small-scale task offloading, it can stably provide solutions that are better than single heuristic methods and more global than pure BCD	In large-scale scenarios, challenges arise from excessive iterations, sensitivity to parameters, and high computational overhead
[54]	2024	Task scheduling	Adaptive alternating direction method of multipliers (AADMM)	Minimize total system cost	Cloud-Edge Continuum	Low computational complexity.	Ignores changes in UE mobility speed
[55]	2025	Resource allocation	Least Recently Used (LRU)	Reduce service latency and energy consumption	Cloud-Edge Continuum	Reduced inter-server communication, thereby lowering latency	When the edge device's cache far exceeds task needs, latency-reduction via high indirect hit rate may be limited

In the Mobile Edge Computing (MEC) architecture (e.g., [47, 48, 51, 53]), the focus shifts toward joint communication–computation optimization and task scheduling fairness. Under such settings, resource constraints are more stringent, and methods based on convex or alternating optimization are commonly used for channel allocation, energy management, and multi-task concurrency control, effectively enhancing spectrum utilization and service quality.

In UAV-assisted MEC (UAV-MEC) systems [18], the spatial mobility of UAV nodes further couples trajectory planning, power control, and task assignment, resulting in highly non-convex and time-varying optimization problems that challenge traditional analytical methods. Additionally, Guo *et al.* [55] introduced a Least Recently Used (LRU)-based caching strategy, demonstrating the potential of integrating classical optimization with cache management to support data-intensive IoE systems through latency-aware lightweight optimization.

In summary, classical optimization methods demonstrate several advantages in IoE task offloading and resource allocation, including solid theoretical foundations, high computational efficiency, and implementation simplicity. These methods are particularly suitable for static networks and analytically tractable scenarios. However, they generally rely on the assumption of complete global information and lack dynamic adaptability, making them less effective in addressing the time-varying and heterogeneous characteristics of IoE environments. Recent research trends have gradually shifted from purely deterministic optimization toward hybrid optimization frameworks that integrate dynamic modeling, heuristic adjustment, and adaptive mechanisms. This evolution provides the theoretical groundwork for subsequent developments in Lyapunov-based optimization, game-theoretic optimization, and AI-driven approaches.

B. Lyapunov Optmization

Lyapunov optimization, as an optimization framework rooted in stochastic network control theory, provides an effective mathematical tool for addressing such problems. By constructing a Lyapunov function that captures the system's queue state, this method optimizes long-term average performance while maintaining system stability, thereby enabling adaptive scheduling for time-varying resource management. Compared with classical optimization, Lyapunov optimization does not rely on future information or offline computation; instead, it makes decisions based on instantaneous system states, making it particularly suitable for IoE scenarios characterized by random task arrivals and frequent network topology changes. A summary of these studies is provided in Table V.

A Lyapunov optimization framework was introduced into MEC resource slicing management to enable adaptive service allocation and delay minimization under dynamic network loads [56]. By constructing a Lyapunov drift model that captures the relationship between queue length and resource utilization, the study transforms long-term performance optimization into a per-slot online decision process. Specifically, the algorithm dynamically evaluates network conditions in each time slot and adjusts resource slice allocation based on the drift-minimization principle, thereby optimizing both energy consumption and task latency while maintaining system stability. Its main advantage lies in being statistically independent, allowing adaptive optimization without prior system information. However, as Lyapunov optimization essentially adopts a greedy slot-by-slot decision mechanism, it may yield suboptimal outcomes in scenarios with multidimensional constraints or strong nonlinearity, making it difficult to guarantee strict global optimality. Moreover, its convergence performance depends heavily on the drift-control parameter, whose robustness remains an open challenge.

Building upon this, Luo *et al.* [57] extended the Lyapunov optimization framework in MEC systems by proposing a joint task scheduling and resource allocation algorithm to balance network performance and latency reduction. The study formulated a multi-dimensional dynamic queue model that jointly considers task offloading decisions between UE and edge nodes, as well as their effects on delay and energy consumption. By minimizing the weighted sum of the expected Lyapunov drift and power consumption function, the algorithm dynamically adjusts task distribution and computational resource allocation in each time slot, achieving a balance among latency, energy efficiency, and stability. This method maintains queue stability under complex network dynamics and significantly reduces average task delay even under heavy loads. However, the study also noted that for high-dimensional coupled optimization problems, the Lyapunov framework suffers from rapidly increasing computational complexity, particularly in multi-node, multi-resource cooperative scenarios. Additionally, its sensitivity to environmental fluctuations may cause drift oscillations in highly non-stationary IoE settings, potentially degrading system stability.

Overall, research on Lyapunov optimization in IoE task offloading and resource allocation remains relatively limited. Nevertheless, its core principle, which focuses on system stability analysis and long-term performance optimization through drift minimization, offers a promising direction for online decision-making in dynamic environments. Compared with traditional deterministic optimization, this approach can maintain queue stability and reduce average latency under random task arrivals and channel variations, demonstrating strong adaptability. However, its high computational complexity, sensitivity to parameter tuning, and difficulty in achieving global optimality in high-dimensional and non-stationary environments limit its practical deployment. Consequently, current studies mainly concentrate on theoretical exploration and performance evaluation in small to medium-scale IoE systems.

TABLE V. LYAPUNOV OPTIMIZATION

Work	Year	Problems addressed	Utilized technique	Optimization objective	Architecture	Advantages	Limitations
[56]	2022	Resource allocation	Lyapunov optimization	Optimize resource-slice selection	MEC	Increased utilization of slice service resources, thereby reducing service delay and energy consumption	Cannot guarantee global optimality
[57]	2025	Task scheduling and Resource allocation	Lyapunov optimization	Balanced network performance and reduced latency	MEC	Optimized task distribution between the UE and edge computing nodes, reducing transmission and computation latency	Faces challenges of high-dimensional computational explosion and sensitivity to non-stationary environments in complex scenarios

C. Heuristic-Based Optimization

With the expansion of IoE systems and the increasing complexity of network structures, traditional analytical optimization and Lyapunov-based methods face significant limitations in terms of real-time performance and scalability. Heuristic-based optimization has emerged as an effective approach for addressing large-scale non-convex optimization problems due to its relatively low computational complexity and high flexibility. These methods draw inspiration from natural swarm behaviors, evolutionary processes, or intelligent search mechanisms to obtain high-quality approximate solutions within limited time, making them particularly suitable for IoE scenarios characterized by complex task coupling and analytically intractable models. Compared with traditional optimization, heuristic approaches can perform adaptive searches without relying on problem convexity or precise system modeling. However, the optimality and convergence of their solutions are often difficult to guarantee. In recent years, various heuristic and meta-heuristic algorithms, such as evolutionary algorithms, swarm intelligence techniques, and hybrid search mechanisms, have been proposed to optimize task scheduling and resource allocation in IoE environments. A summary of these approaches is presented in Table VI.

Aqeel *et al.* [58] introduced a Chaotic Horse Ride Optimization Algorithm (CHROA) with an embedded chaotic mapping mechanism to address resource allocation and load balancing in cloud–fog collaborative environments. By incorporating a chaotic search operator to enhance population diversity and designing a fitness function that jointly evaluates node load and energy consumption, the method achieves a balance between load distribution and energy efficiency. Its key strengths lie in preventing premature convergence through chaotic dynamics and integrating multi-objective trade-offs within the fitness evaluation. The approach demonstrates strong adaptability to non-convex optimization and high scalability, making it suitable for large-scale heterogeneous systems. However, the absence of explicit task deadline constraints may cause violations under stringent latency requirements. Moreover, parameters such as population size and chaotic coefficients considerably influence performance, demanding careful tuning and robustness validation.

In another line of work, Altin *et al.* [59] employed two well-established multi-objective evolutionary algorithms, NSGA-II and SPEA2, to optimize task scheduling in cloud–fog environments with the dual goals of minimizing transmission cost and mitigating network congestion. Through customized encoding and crossover/mutation operators combined with crowding-based selection strategies, the approach yields a Pareto-optimal set of scheduling decisions. It effectively handles conflicting objectives and mitigates network bottlenecks, improving both throughput and system stability. Nonetheless, the experiments assume idealized channel conditions and overlook short-term wireless link fluctuations, leaving its performance in highly dynamic IoE networks yet to be validated.

TABLE VI. HEURISTIC-BASED OPTIMIZATION

Work	Year	Problems addressed	Utilized technique	Optimization objective	Architecture	Advantages	Limitations
[58]	2023	Resource allocation	Chaotic Horse Ride Optimization Algorithm (CHROA)	Improving load balancing and reducing energy optimization	Cloud-Fog Continuum	Integrated optimization of load balancing and energy efficiency	The potential risk of deadline violations is neglected
[59]	2024	Task scheduling	NSGA-II and SPEA2	Reduce data transmission cost	Cloud-Fog Continuum	Proactively avoid network congestion	Ignores transmission fluctuations between UE and AP
[60]	2024	Task scheduling	Custom approximation algorithm	Minimize the total cost of processing tasks	Cloud-Edge Continuum	Low computational complexity.	Assumes that all jobs are of the same type and arrive simultaneously at time zero, and that the edge servers have the same speed.
[61]	2024	Task scheduling and Resource allocation	wolf fish collaborative search algorithm (WF-CSA)	Minimize total delay and maximize communication success rate	Cloud-Edge Continuum	Improved communication success rate via dynamic edge-node trust mechanism	Validation was conducted only in low density and simple scenarios

Building on a different perspective, Li and Ou [60] proposed a low-complexity approximate scheduling algorithm designed for homogeneous batch jobs that arrive simultaneously under identical edge server speeds. The method minimizes total processing cost—including both computation and transmission—by leveraging structural reduction and approximate assignment rules. While computationally efficient and easy to implement, its reliance on restrictive assumptions such as task homogeneity and synchronized arrivals limits generalization to real-world IoE environments characterized by heterogeneous tasks and asynchronous arrivals.

Meanwhile, Tian *et al.* [61] presented a hybrid swarm intelligence method, the Wolf Fish Collaborative Search Algorithm (WF-CSA), aimed at jointly optimizing task scheduling and resource allocation in cloud–edge collaborative systems. The algorithm introduces a dynamic trust and reliability evaluation mechanism for edge nodes, prioritizing task assignment to nodes with stable and reliable connections. Results show substantial gains in communication success rate and latency reduction under low-density conditions. However, the evaluation scenarios remain relatively simple, and the method's scalability and parameter sensitivity require further investigation in large-scale, interference-prone, or mobile IoE environments.

In summary, heuristic optimization methods draw inspiration from natural collective behaviors and evolutionary search mechanisms to achieve near-optimal solutions for non-convex and complex IoE optimization problems with relatively low computational cost, making them well-suited for large-scale and dynamic scenarios. These methods provide significant advantages in implementation flexibility, model independence, and fast convergence, thereby enhancing system performance in complex environments. Nevertheless, they typically lack theoretical guarantees of optimality, exhibit sensitivity to parameter tuning, and face challenges in maintaining stability and scalability under highly dynamic IoE conditions. Hence, while heuristic optimization shows strong potential for practical engineering deployment, its further advancement depends on improving adaptability and robustness in dynamic and uncertain environments.

D. Game Theory-based Optimization

Game-theoretic optimization provides an effective decision-making framework for IoE task offloading and resource allocation by modeling the competitive and cooperative interactions among intelligent agents such as user devices, edge nodes, and service providers. Unlike deterministic optimization, game-theoretic models capture the strategic interdependence and conflict of interests among multiple entities, allowing the system to seek equilibrium solutions such as Nash equilibria in complex environments. An overview of these approaches can be found in Table VII.

A two-tier game model was introduced to address task scheduling in fog computing and Device-to-Device (D2D) environments [62]. The upper-tier game models resource competition among fog nodes, while the lower-tier game handles task offloading among user devices. The approach innovatively incorporates human activity patterns into the network load model, making it more representative of realistic IoE scenarios. Experimental results demonstrate that this framework significantly enhances task offloading efficiency and service responsiveness. However, since participating agents require partial global information to update their strategies, convergence speed is limited in incomplete or dynamic information environments. Moreover, equilibrium stability under high-density deployments remains insufficiently explored.

Building upon this direction, Wang and An [63] introduced a differential game framework to address dynamic task scheduling in MEC. Users and edge service providers are modeled as continuous-time players, and the Nash equilibrium is derived by solving the Hamilton–Jacobi–Bellman (HJB) equations. The objective is to minimize user service costs while maximizing provider profits, thereby achieving a bilateral optimum between users and service providers. This study effectively balances service costs and provider benefits over time, maintaining sustainable resource utilization and economic efficiency. Nonetheless, the assumption of constant communication noise does not reflect the stochastic nature of IoE wireless channels, limiting its applicability. Furthermore, solving HJB equations involves high computational complexity, which constrains its use in real-time scheduling.

TABLE VII. GAME THEORY-BASED OPTIMIZATION

Work	Year	Problems addressed	Utilized technique	Optimization objective	Architecture	Advantages	Limitations
[62]	2021	Task scheduling	Two-tier game	Improve task offloading efficiency	Fog and Device-to-device	Considers the impact of human activities	Global information is hard to obtain, limiting convergence performance.
[63]	2024	Task scheduling	Differential game theory	Minimizes the edge computing service cost for users while maximizing the benefits for edge service providers.	MEC	Obtain the optimal strategy at the Nash equilibrium.	Assumes that the noise in satellite communications is constant, which does not match real-world conditions.
[64]	2024	Resource allocation	Evolutionary game-theoretic approach	Improving the QoS of Edge-based IoE Services	Edge	Increasing network throughput leads to a reduction in task latency.	The effect of optimizing edge computing resource allocation on the overall system performance has been overlooked.

An evolutionary game-theoretic approach has been proposed for resource allocation in edge IoE services [64], where multiple edge nodes iteratively adjust their strategies following replicator dynamics until convergence to an Evolutionarily Stable Strategy (ESS). This mechanism enables distributed self-organization without centralized control, leading to improved network throughput and reduced task latency. However, the study did not fully examine the global impact of resource optimization on energy fairness or node saturation, and the convergence stability of the algorithm depends on the initial strategy distribution and evolution parameters.

Overall, game-theoretic optimization demonstrates notable advantages in distributed modeling, locally rational decision-making, and self-organizing stability for IoE task offloading and resource allocation. Its core strength lies in achieving dynamic equilibrium through strategic interactions among competitive or cooperative agents, effectively mitigating the computational and communication overhead of centralized optimization. Additionally, game-theoretic frameworks offer high scalability and adaptability, making them suitable for multi-layer IoE architectures (such as cloud–edge–device collaboration) and multi-objective optimization scenarios involving latency, energy consumption, and profit trade-offs.

E. Data-Driven and AI-Native Optimization

As the complexity and dynamism of IoE systems continue to increase, traditional approaches such as analytical optimization, heuristic methods, and game-theoretic models struggle to maintain optimal performance when facing high-dimensional state spaces, time-varying network conditions, and uncertain task patterns. To address these challenges, data-driven and AI native optimization has emerged as a mainstream direction in IoE task offloading and resource allocation research. Unlike conventional methods that rely on explicit mathematical modeling, this paradigm leverages intelligent algorithms such as Machine Learning (ML) and Deep Reinforcement Learning (DRL) to directly learn optimal or near-optimal scheduling strategies through environmental interaction, marking a paradigm shift from optimization-based computation to intelligent decision making. The specific related studies are listed in Table VIII.

Data-driven and AI-native optimization methods have rapidly developed in the domain of IoE task offloading and resource allocation, emerging as an effective approach to handle high system dynamics and complex coupling. Unlike traditional optimization methods that rely on precise modeling, these approaches leverage DRL as the core, learning optimal strategies through interaction with the environment and extracting decision patterns from empirical data, thereby eliminating dependence on task arrival distributions or channel models. Within a single-agent reinforcement learning framework, algorithms such as Deep Q-learning Network (DQN), Deep Deterministic Policy Gradient (DDPG), and Synchronous Advantage Actor-Critic (A3C) have been applied in centralized or semi-centralized scenarios to jointly optimize latency, energy consumption, and system throughput. For example, one study combined Graph Attention networks (GAT) with DRL to capture cloud–edge–end node topologies, achieving a dynamic trade-off between energy and delay [65]. Another applied DDPG to optimize wireless resource allocation, enabling joint decisions on task offloading and power control in continuous action spaces [66]. A further approach incorporated end-device battery constraints into the reward function, minimizing both delay and energy consumption simultaneously [67]. These studies demonstrate that single-agent DRL possesses strong non-linear modeling and adaptive capabilities for handling complex, high-dimensional decision-making problems.

One study proposed a task offloading method combining a double-learning structure with an energy-aware framework to balance latency, energy consumption, and system utility [68]. In a mobile edge computing environment, the approach uses a two-layer learning structure: the lower layer employs DRL to learn offloading strategies, while the upper layer dynamically adjusts the reward function based on an energy estimation model, allowing the policy to adapt to the remaining battery levels of devices. Experimental results show that this method can significantly extend device battery life in energy-constrained scenarios while reducing overall energy consumption without sacrificing system throughput. Its key advantage lies in integrating energy state feedback into the learning process, enhancing the model's energy adaptively. However, the approach is sensitive to energy prediction accuracy; any deviation in the energy model can lead the policy to behave overly conservatively or aggressively. In addition, the two-layer learning structure increases computational complexity, posing challenges for deployment on low-capacity end devices.

To accommodate distributed, heterogeneous, and privacy-sensitive IoE systems, researchers have introduced Multi-Agent Deep Reinforcement Learning (MA-DRL) and Federated Reinforcement Deep Learning (F-DRL) frameworks. One approach combines MA-DRL with F-DRL, allowing each end device to learn locally and periodically upload policy updates for global aggregation and strategy sharing, thereby achieving network-wide collaborative optimization without exposing raw data [69]. Another study leverages a Double Deep Q-network (DDQN) with a Distributed Optimization (D2OP) mechanism to jointly optimize response time and load balancing, demonstrating the feasibility of multi-agent frameworks in complex MEC systems [70]. Additionally, integrating the A3C algorithm with digital twin technology enables simulation in a virtual environment to accelerate policy convergence and significantly improve training efficiency [71]. These methods offer the advantage of enabling collaborative learning and knowledge sharing while maintaining a distributed system structure. However, they also face challenges such as high communication overhead, non-stationary policies, and difficulties in controlling the frequency of aggregation.

Some studies have explored hybrid intelligent optimization frameworks that integrate AI algorithms with classical methods to maintain learning capability while reducing training complexity. One approach uses Support Vector Machines (SVM) for task classification combined with DRL to reduce the action space and improve policy convergence [72]. Another introduces Johnson's pre-sorting rules as prior knowledge for DRL, enabling faster batch task scheduling [5]. A dual-algorithm structure combining DDQN and DDPG models scheduling and resource allocation as discrete and continuous action problems, respectively, effectively avoiding inefficient searches in mixed action spaces [73]. Further work integrates Dueling DQN with Double DQN and incorporates a dynamic auction compensation mechanism to balance resource utilization and fairness [74]. Hybrid frameworks generally outperform standalone deep learning models, but due to coupling between different modules, joint training can be unstable or exhibit convergence oscillations, which may require phased training or parameter freezing for improvement.

In continuous control and energy-constrained scenarios, DRL applications exhibit considerable diversity. One study proposed the Energy Harvesting Edge Computing with Softmax Deep Double Deterministic (EHEC_SD3) algorithm based on DRL to achieve energy-aware task offloading and resource allocation in wireless energy-supplied multi-access edge computing systems, effectively preventing terminal devices from running out of power [75]. Another work combined Self-Organizing Maps (SOM) with DRL for resource optimization in hybrid wired/wireless IoE environments, reducing the state space dimensionality through feature clustering and thus alleviating training burdens [76]. Meanwhile, predictive enhancement models such as deep recurrent learning and Graph Neural Networks (GNN) have been used to capture temporal correlations of tasks and channels. Another study leveraged future task load and user behavior prediction to improve quality of service, but due to the complexity of model training and slow convergence, its online adaptability remains limited [77].

One study focused on long-term system utility optimization for intelligent scheduling, using a deep reinforcement learning framework to dynamically allocate tasks in time-varying IoE networks [78]. Unlike most short-term performance-oriented approaches, this work incorporated a cumulative long-term performance term into the reward function, enabling the learned policy to balance energy consumption, latency, system load, and resource utilization over extended periods. Experimental results demonstrated superior stability and balanced utility under high-load conditions. The main contribution lies in highlighting the potential of reinforcement learning for multi-stage, long-term decision-making. However, the method also exposed issues such as slow convergence and training instability, particularly under highly variable task arrival distributions, which could lead to policy drift. Additionally, the model did not fully consider task safety and communication reliability, limiting its applicability in critical IoE scenarios such as industrial control and vehicular networks.

From an overall perspective, data-driven and AI-native optimization methods demonstrate significant flexibility and adaptability compared to traditional analytical and heuristic approaches. They can directly learn high-dimensional mappings in unknown environments, make real-time decisions through policy networks, and simultaneously optimize multiple performance metrics in complex systems. However, these methods also face several common challenges. First, the training process relies heavily on extensive environment interactions, resulting in low sample efficiency. Second, the models often lack interpretability, making it difficult to theoretically analyze stability and optimality. Third, practical deployment demands considerable computational resources and energy consumption, which limits long-term operation on low-power devices. Therefore, improving learning efficiency, reducing model complexity, and enhancing generalization and stability remain key directions for advancing AI-native optimization in IoE task offloading and resource management.

TABLE VIII. DATA-DRIVEN AND AI-NATIVE OPTIMIZATION

Work	Year	Problems addressed	Utilized technique	Optimization objective	Architecture	Advantages	Limitations
[65]	2021	Task scheduling	Reinforcement learning (DRL) and graph attention network (GAT)	Maximize the trade-off utility between delay efficiency and energy consumption efficiency	Cloud-Edge Continuum	Trade-off between delay efficiency and energy efficiency	Only considers the high energy consumption of UEs while ignoring the constraint of remaining battery capacity.
[69]	2021	Task scheduling and Resource allocation	Multi-agent DRL (MA-DRL) and federated DRL(F-DRL)	Minimize overall energy consumption while meeting latency requirements	Cloud-Edge Continuum	Supports application scenarios in ultra-dense, highly dynamic networks	Only proposed the system framework without detailing its components.
[66]	2022	Task scheduling and Resource allocation	Deep deterministic policy gradient (DDPG)	Reduce total computation overhead	Wireless Edge Computing (WEC)	Jointly optimize task offloading decisions and wireless resource coordination	Does not consider computing resource allocation alongside offloading decisions

[67]	2022	Task scheduling and Resource allocation	Deep reinforcement learning	Minimize mobile device delay cost and energy cost	MEC	Considers battery constraints in joint multi-objective optimization	There are challenges in the rational allocation of weights among energy consumption, processing delay, and communication delay
[68]	2022	Task scheduling and Resource allocation	Round Robin (RR), Strongest Channel (SC), Max Rate (MR), Proportional Fair (PF) and Priority Base (PB)	Minimise the total time	Cloud-Edge Continuum	Low computational complexity	Lacks global optimization capability
[72]	2022	Resource allocation	Deep reinforcement learning (DRL) and Support vector machine (SVM)	Reduce energy consumption and latency	Cloud-Edge Continuum	Introducing an SVM classification model improves prediction accuracy and reduces the computational load of DRL	UE mobility impact on resource allocation is overlooked
[77]	2022	Resource allocation	Deep Recurrent Learning	Improving Quality of Service (QoS)	WN interconnection	Predicting and estimating IoE applications and users to minimize conflicts	Long training times, slow convergence, and limited adaptability to dynamic environments
[78]	2022	Task scheduling	DRL	Maximize the expected long-term system payoff	MEC	The algorithm has polynomial time complexity	Does not discuss potential impacts during the data transmission process.
[70]	2023	Task scheduling	DDQN with D2OP	Reduce response time, load balance, and increasing task success ratio	MEC	Joint optimization of task response time and SLB was conducted, with careful consideration of the trade-off between the two objectives.	Data security issues in the scheduling process were mentioned but not explored in detail.
[5]	2023	Task scheduling and Resource allocation	DRL and Johnson's Rule-Based Presorting	Minimize the total completion time	Cloud-Edge Continuum	Johnson's rule-based presorting reduces the computational load of the DRL and improves response speed	The division of presorted sublists introduces multiple dependencies in the multi-actuator environment, creating significant challenges for global optimization
[71]	2024	Task scheduling and Resource allocation	Asynchronous advantage actor-critic (A3C)	Minimize long-term cost of the tasks along the trip	Cloud-Edge Continuum	Digital twin–enabled deep reinforcement learning facilitates the generation of globally optimal strategies	The quadratic growth of sample complexity results in a significant increase in computational load
[76]	2025	Resource allocation	Self-Organized Map (SOM)-based Deep Reinforcement Learning (DRL)	Reduce latency and energy consumption	Cloud-Edge Continuum	Supporting hybrid wired and wireless communication reduces energy consumption and latency	The classification capability of SOM affects the overall performance of resource allocation
[75]	2025	Resource allocation	DRL-based EHEC_SD3	Ensures consistent energy consumption between EH and IoT devices during task execution	Wireless powered multi-access edge computing (WPMEC)	Jointly considers UE battery levels and wireless charging allocation to prevent battery depletion	Task processing speed optimization is ignored
[73]	2025	Task scheduling and Resource allocation	Double deep Q-network (DDQN) and deep deterministic policy gradient (DDPG)	Minimizing the task completion delay	Cloud-Edge Continuum	DDQN performs scheduling, and DDPG allocates resources. Joint optimization prevents local optima caused by decoupled problem-solving	Overlapping computational costs and amplified non-stationarity may result in training divergence
[74]	2025	Task scheduling and Resource allocation	Double DQN and Dueling DQN	Enhance computational efficiency and resource utilization	Cloud-Edge Continuum	The proposed scheme can adapt to dynamic environments while achieving a balance between energy efficiency and accuracy	Even though the auction operates dynamically, the compensation factor only adjusts bids based on failed auctions and does not take into account resource changes, task arrival fluctuations, or network bandwidth variations

F. Hybrid and Collaborative Optimization

As IoE systems grow in scale, heterogeneity, and dynamism, single optimization methods often struggle to meet real-time, accuracy, and stability requirements simultaneously. Hybrid and collaborative optimization address this by integrating traditional analytical models, such as convex optimization, Lyapunov optimization, and game theory, with intelligent methods like deep reinforcement learning, heuristic search, or evolutionary algorithms. This approach balances theoretical interpretability with data-driven adaptability, improving convergence, robustness, and real-time performance in non-stationary environments. Table IX summarizes representative research efforts in this domain.

TABLE IX. HYBRID AND COLLABORATIVE OPTIMIZATION

Work	Year	Problems addressed	Utilized technique	Optimization objective	Architecture	Advantages	Limitations
[79]	2020	Task scheduling	Lyapunov and DRL(Actor-Critic)	Reduce system cost	MEC	Estimate edge server status via digital twin (DT), improving decision accuracy	Lacks comparison with similar RL methods
[80]	2020	Task scheduling	Game theory and DRL	Reduce latency & energy, increase scalability	Cloud-Edge Continuum	Decentralized decisions, avoiding single-point overload	Slow convergence; hard to achieve global optimum
[81]	2021	Resource allocation	DRL and Heuristic	Improving QoS and QoE	MEC	Both vertical and horizontal scalability are considered jointly	The impact of network fluctuations on allocation decisions is ignored
[82]	2022	Resource allocation	Quasi oppositional based learning with the traditional SRO	Minimizing system average cost	Cloud-Edge Continuum	The optimization process simultaneously ensures balanced load distribution across nodes	Has limited capability to swiftly adapt to large-scale network dynamics
[83]	2023	Task scheduling	Offloading decisions based on preferences and NSGA-II	Minimizing the latency	MEC	Considering the user's preference	Slow convergence, costly repair, sensitive to parameters
[19]	2024	Resource allocation	Leader-based Optimization (LO) and Adaptive Differential Evolution (ADE)	Increase energy efficiency	Edge	High reliability and scalability	Faces challenges in rapidly adapting to dynamic changes and may exhibit limitations in real-time decision-making speed
[84]	2025	Resource allocation	Lyapunov optimization and metaheuristic search	Balance latency and accuracy	Cloud-Edge Continuum	Improves the trade-off among network latency, inference delay, and analysis accuracy	All tasks from the UE are offloaded, with no consideration for partial local execution on the UE
[85]	2025	Resource allocation	Particle swarm optimization (PSO) and Decentralized federated learning (DFL)	Minimizing the mean latency	Edge	Fully leverages overlapping inter-satellite links to enhance DFL latency efficiency	Transmitting FL data on low-bandwidth, high-latency links poses performance challenges
[86]	2025	Task scheduling and Resource allocation	Convex optimization and multi-agent reinforcement learning (MARL)	Reduce the overall energy consumption of the terminals	MEC	The hybrid approach consumes significantly less energy than the single-agent algorithm, QMIX, SAC	Individual MEC server resource usage is ignored
[87]	2025	Task scheduling and Resource allocation	Stackelberg game and MADDPG	Reduces task loss rate and system delay	MEC	Supports heterogeneous edge networks and well-suited for handling high-density, latency-critical tasks	Resource allocation faces challenges due to the random distribution and mobility of UEs
[88]	2025	Task scheduling and Resource allocation	Successive convex approximation and deep reinforcement learning (DRL)	Reduce CO2 emissions	Edge	Reduce CO2 emissions by optimizing key factors, such as IoT device transmit power, DT-estimated processing rates, and task offloading decisions	Lack of optimization for task completion time

In Ref. [79], digital twin predictions have been combined with Actor-Critic reinforcement learning under a Lyapunov optimization framework to enable dynamic task offloading in 6G millisecond time slots. In this scheme, the digital twin estimates the queue length at each edge node one millisecond in advance, while the Actor network outputs continuous offloading ratios based on these estimates, and the Lyapunov drift-plus-penalty module ensures minimal long-term average cost. Experiments on a custom 6G simulation platform demonstrate an 18% reduction in total system cost and a 25% decrease in queue overflow probability compared with nearest-first strategies. However, the baseline was weak, lacking comparisons with homogeneous RL methods such as DDPG or Multi-Agent Deep Deterministic Policy Gradient (MADDPG), and no sample efficiency or convergence time was reported. The assumption that all tasks must be offloaded also neglects local computation capabilities, limiting applicability in terminal-edge collaborative scenarios.

Urban hotspot-coldspot load imbalances have been addressed through a decentralized offloading-caching framework incorporating congestion-price negative feedback [80]. In this two-level game, the upper-layer edge controller acts as a Stackelberg leader broadcasting dynamic prices based on global congestion, while terminals or micro-edge nodes, as followers, run local DDPG agents to decide offloading ratios and caching variables in a three-dimensional state space of price, local CPU, and channel. Simulations with 500 terminals across

45 nodes indicate a 22% reduction in average task delay, 15% lower terminal energy consumption, and 40% decrease in hotspot CPU utilization variance. Despite these improvements, the DDPG agent converges slowly, requiring a long warm-up for online deployment. The price update interval of 100 milliseconds also exceeds typical wireless channel coherence times, ignoring fast fading, and global optimality remains theoretical.

To handle massive IoE workloads, a two-level vertical-horizontal online scaling framework has been proposed [81]. The upper level employs an improved Dyna-Q to determine discrete scaling actions at fine granularity, while the lower-level places container instances using a lightweight Best-Fit heuristic. The state space captures CPU, memory, and bandwidth utilization, with rewards defined as a weighted sum of QoS gain, resource cost, and migration penalty, interfaced with Kubernetes-HPA. Experiments in a 120-camera factory scenario show a 28% improvement in QoS, a 21% increase in resource utilization, and a 35% reduction in scaling triggers. Limitations include simplified Gaussian channel modeling, lack of explicit energy consideration in rewards, and insufficient comparison with recent RL baselines such as A3C, PPO, or SAC.

Quasi-oppositional learning has been explored for Cybertwin 6G to reconcile global view and real-time load balancing requirements [82]. By generating solutions and their opposites during population initialization, the method expands search coverage and accelerates convergence. Simulations indicate reduced system costs compared to ECORA, but the 30-second re-optimization period is inadequate for second-level traffic bursts, and fitness evaluation exceeds two seconds for over 5000 nodes. Furthermore, topology changes require full reinitialization, incurring high computational overhead.

To overcome NSGA-II's uniform treatment in mobile edge scenarios, user preferences have been integrated into multi-objective evolutionary optimization [83]. Preference weights for delay and energy are extracted from questionnaires and incorporated through reference-point distance filtering, steering the Pareto front toward regions of user interest. Simulated binary crossover operators preserve preference directions. Experiments on mobile video editing show substantial improvements in user satisfaction and proximity to ideal solutions. However, repair operator complexity scales poorly with population size, the process runs offline, and comparisons are limited to early NSGA versions.

In ultra-dense, energy-constrained IoE environments, a two-level hybrid algorithm combining leader selection and adaptive differential evolution has been proposed [19]. Leaders are selected based on residual energy to broadcast current optimal solutions, while non-leader nodes dynamically adjust mutation and crossover strengths. Sensor network experiments demonstrate superior energy efficiency, reliability, and convergence speed compared to traditional GA-PSO hybrids. Nevertheless, full leader election and evolution per generation impose heavy computational loads, and comparisons remain limited to evolutionary methods, excluding reinforcement learning or convex optimization alternatives. Carbon emissions are not incorporated in the objective function, limiting compliance with green network requirements.

Lyapunov queue stability combined with simulated annealing and tabu search has been applied in an inner-outer loop for real-time video streaming [84]. The outer loop allocates resources based on backlog cost every millisecond, and the inner loop searches batch size and quantization bitwidth every 20 milliseconds to balance latency and accuracy. Experiments with 200 ultra-HD streams show a 6% increase in inference accuracy and minimal queue overflow. However, all frames are assumed offloaded, leaving terminal GPUs/NPU idle, and channel modeling simplifications may reduce real-world applicability.

In Ref. [85], Low Earth Orbit (LEO) constellations are treated as aerial parameter servers for remote IoT. Particle swarm optimization is used to find time-efficient routes across inter-satellite links, while federated learning selects clients based on reliability thresholds and uploads gradients in parallel. Overlapping aggregation balances waiting times. Simulations with the Walker constellation and thousands of ground terminals show that average learning latency is reduced by about 20% compared to fixed routing, while reliability remains at enterprise-grade levels. However, full-precision gradients are transmitted over the already scarce Ka-band, quickly saturating the bandwidth. Techniques such as gradient compression, quantization, or sparsification are absent, making satellite-to-ground links the practical bottleneck. Furthermore, when satellite failures trigger rerouting, only latency is recalculated without considering energy consumption, which may further drain already limited terminal batteries.

Research on combining non-orthogonal multiple access with mobile edge computing aims to improve energy efficiency and reduce training time. A study adopts a convex approximation followed by multi-agent DRL [86]. The convex step provides initial offloading and power allocations, reducing training iterations and lowering energy consumption by over 10% compared to pure multi-agent approaches. Limitations include ignoring node-level thermal constraints and excluding carbon emissions from rewards.

To support autonomous decision-making in dense IoE networks, market-driven approaches have modeled edge operators as price setters, allowing mobile terminals to determine offloading and transmission power based on dynamic prices [87]. In this framework, a Stackelberg-MADDPG structure reduces task loss by nearly 30% compared to Q-learning under random mobility. However, simplified mobility models, limited cross-paradigm comparisons, and hidden energy metrics reduce practical interpretability.

In scenarios where both latency and carbon emissions are critical, a carbon-aware framework has been proposed, using successive convex approximation in the outer loop and Double-DQN in the inner loop to minimize CO_2 emissions while maintaining video latency below 80 milliseconds [88]. Simulations with 1,000 nodes show a 25% reduction in emissions, but all computation occurs at the edge, leaving terminal resources idle. In addition, partial offloading or dynamic frequency scaling is unsupported, and comparisons are limited to a few static strategies, leaving the carbon reduction potential uncertain.

Overall, hybrid and collaborative optimization has emerged as a key research direction in IoE task offloading and resource allocation. Its core principle is to leverage complementary strengths of different optimization frameworks, combining the theoretical robustness of analytical models with the adaptability of data-driven algorithms to achieve a balance between performance and complexity in dynamic environments. Traditional methods such as Lyapunov optimization, game theoretic modeling, convex programming, reinforcement learning, and heuristic search provide complementary strengths in stability analysis, equilibrium formulation, constraint modeling, and global optimization. When combined with artificial intelligence techniques, these methods enable autonomous learning and cross-scenario adaptability. Challenges remain, including algorithmic coupling complexity, limited interpretability, and difficulties in distributed deployment.

VI. Analysis and Discussion

In the previous chapter, the main research directions in IoE task offloading and resource allocation were systematically reviewed from the perspective of optimization paradigms, including classical optimization, Lyapunov optimization, heuristic methods, game-theoretic models, AI-native optimization, and hybrid and collaborative optimization. Building on this foundation, this chapter aims to summarize and reflect on these studies, examining the core principles of IoE optimization and the evolution trends of methodologies from a holistic perspective. Specifically, the chapter seeks to extract key findings through comparative and integrative analysis of existing works, highlighting the complementary strengths and applicable boundaries of different optimization paradigms, identifying unresolved scientific and engineering challenges, and exploring potential directions for future research. Such a systematic analysis not only clarifies the developmental trajectory of IoE optimization but also provides theoretical references and methodological guidance for subsequent studies.

A. Key Findings Recap

Based on the quantitative analysis of the surveyed literature the following key findings are identified and discussed respectively.

1) Distribution of research focus across problem domains

The statistical distribution of research focus reveals a distinct tripartite landscape in IoE edge computing optimization, as shown in Fig. 7. A significant portion of the literature continues to prioritize isolated sub-problems, with 16.32% of studies addressing task scheduling exclusively (focusing on offloading decisions) and 18.36% concentrating solely on resource allocation (focusing on spectrum or computational power distribution). Together, these decoupled approaches constitute the majority (approx. 35%), reflecting a pragmatic methodology where researchers isolate variables to mitigate the computational intractability inherent in large-scale IoE networks. However, the emergence of joint optimization, which accounts for 16.32% and effectively equals the share of task scheduling, signals a critical paradigm shift. This substantial proportion highlights the growing recognition of the intrinsic coupling between communication and computation resources. It suggests that while decoupled optimization remains a vital low-complexity benchmark, the field is irreversibly gravitating towards holistic, multi-dimensional synergy to meet the stringent end-to-end QoS requirements of next-generation IoE applications, despite the algorithmic challenges involved.

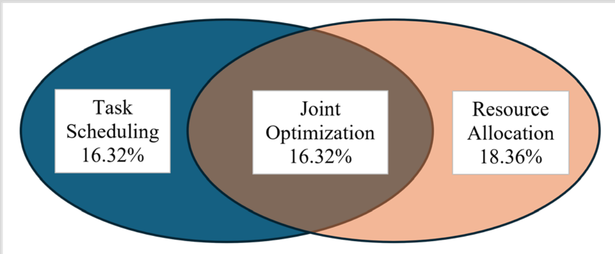

Fig. 7. Distribution of research topics.

2) Adopted technologies

Based on the foregoing analysis, it is evident that research on IoE task offloading and resource allocation has evolved from early analytical models toward intelligent, adaptive, and collaborative approaches, with each paradigm exhibiting distinct characteristics in theoretical foundation, optimization mechanism, and system applicability. Classical optimization methods, represented by convex optimization, linear programming, and alternating iteration, offer strong theoretical interpretability, low computational complexity, and high solution accuracy, making them particularly suitable for deterministic scenarios where system states are known and constraints are analytically tractable. However, their limitation lies in poor adaptability to dynamic environments and high-dimensional non-convex problems, making it difficult to maintain globally optimal performance under the time-varying conditions of real IoE networks.

Lyapunov optimization methods demonstrate unique advantages in stability control and long-term performance trade-offs, achieving dynamic balancing of delay, energy consumption, and queue length through drift-based constraints. They are particularly suitable for online decision-making under time-varying workloads. However, these methods typically rely on accurate estimation of system parameters and have limited scalability in complex heterogeneous environments, which explains their relatively limited application in IoE task offloading.

Heuristic and metaheuristic optimization methods, with their flexible algorithm design, low computational complexity, and model independence, have become important tools for tackling non-convex, combinatorial, and high-dimensional optimization problems. Typical examples include genetic algorithms, particle swarm optimization, and ant colony optimization, which perform well in resource-constrained or analytically intractable scenarios. However, these methods generally lack guarantees of optimality, are sensitive to parameters and initial conditions, and can easily become trapped in local optima in highly dynamic IoE systems, limiting their stability and generalizability.

Game-theoretic optimization methods focus on interactive decision-making and strategy equilibrium among multiple agents, providing an effective modeling framework for resource competition in multi-user and multi-node IoE scenarios. Non-cooperative games, Stackelberg games, and cooperative game models are widely applied in edge computing and network slicing to capture complex competitive and collaborative relationships. However, their main challenges lie in the high computational complexity of equilibrium computation, the extensive information requirements, and limited ability to respond in real time to dynamic network conditions.

Data-driven and AI-native optimization methods represent the forefront of current research, leveraging deep reinforcement learning, graph neural networks, and federated learning to achieve end-to-end adaptive decision-making, thereby overcoming the reliance of traditional models on prior system knowledge. These approaches demonstrate significant advantages in high-dimensional state spaces, nonlinear constraints, and multi-objective optimization, enabling the automatic learning of near-optimal strategies in unknown or dynamic environments. However, the high computational cost, convergence stability, and limited interpretability of AI models remain key bottlenecks, restricting their large-scale deployment on energy-constrained IoE devices.

Hybrid and collaborative optimization methods have emerged as a recent comprehensive trend, aiming to integrate the strengths of different paradigms to overcome the limitations of single approaches. By combining Lyapunov stability constraints with reinforcement learning adaptability or leveraging collaborative frameworks between game theory and deep learning, these methods achieve a balance between system stability and intelligent decision-making. Although hybrid approaches significantly enhance performance and robustness, their algorithmic coupling, numerous parameters, and high communication and training costs remain challenges, requiring further refinement to support large-scale heterogeneous IoE environments.

However, the distribution of adopted technologies is notably unbalanced, as illustrated in Fig. 8. A striking bipolar dominance is observed where Classic Optimization and Data-Driven and AI-Native Optimization each command an equal and dominant share of 30%. This equal split reflects a critical transitional phase in the field. Classical methods remain the indispensable gold standard for theoretical benchmarking and static sub-problems, whereas the surge in AI-Native approaches highlights the urgent necessity to address the stochasticity and complexity of modern IoE networks that traditional models struggle to handle. Meanwhile, Hybrid and Collaborative Optimization has captured a significant 22% share to emerge as a powerful third pole. This trend indicates a growing consensus that purely model-based or purely data-driven approaches have reached their respective ceilings. Consequently, researchers are shifting towards methodological synergy by combining the adaptability of AI with the stability of mathematical rigor. In contrast, specialized techniques like Heuristic, Game Theory, and Lyapunov optimization occupy smaller and niche proportions of 8%, 6%, and 4% respectively. This does not imply their obsolescence but rather suggests that they are increasingly integrated as auxiliary components within broader hybrid frameworks instead of being deployed as standalone solutions.

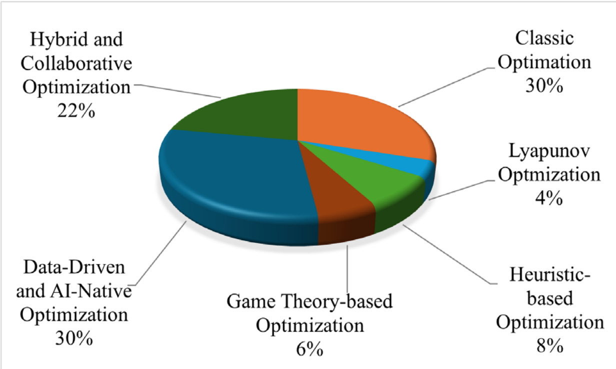

Fig. 8. Distribution of adopted technologies.

3) *Supported system architectures*

The quantitative analysis of system architectures presented in Fig. 9 reveals a decisive preference for hierarchical and collaborative frameworks over isolated computing paradigms.

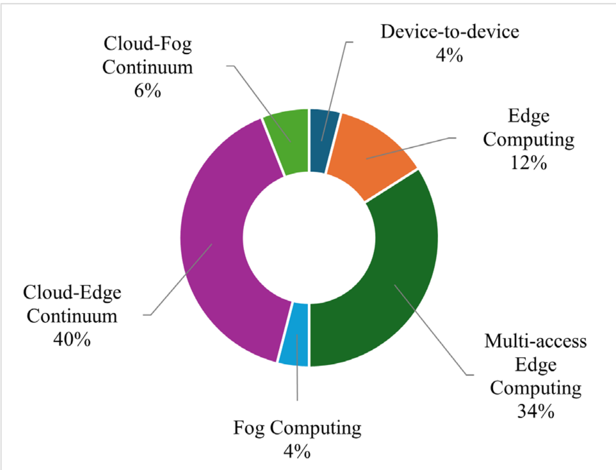

Fig. 9. Supported system architectures across studies.

The Cloud-Edge Continuum commands the largest share at 40%, which underscores a fundamental consensus in the field that edge limitations in storage and computing power necessitate a vertical collaboration with the cloud to handle intensive workloads. Following closely is Multi-access Edge Computing at 34%, highlighting the dominance of telecommunications-standardized architectures that integrate computing capabilities directly into the cellular network infrastructure to support ultra-low latency 5G applications. Together, these two dominant categories constitute nearly three-quarters of the research landscape. This concentration indicates that the research community has largely converged on multi-tier hierarchical models where the synergy between different computing layers is the primary optimization objective. In contrast, other architectures occupy niche positions. Edge Computing as a standalone concept account for 12%, while Cloud-Fog Continuum and Fog Computing represent 6% and 4% respectively, reflecting their specific applicability in decentralized IoT scenarios rather than general-purpose cellular networks. Similarly, Device-to-device communication holds a 4% share, suggesting that horizontal collaboration among end-users is currently viewed as a supplementary offloading mechanism rather than a core architectural backbone.

4) *Breakdown of optimization objectives*

The statistical breakdown of optimization objectives presented in Fig. 10 reveals a distinct hierarchy in research priorities. Service Delay and Energy Consumption stand out as the two predominant objectives with engagement levels reaching approximately 14 and 12 respectively. This clear bimodal prioritization underscores the fundamental conflict in IoE edge computing.

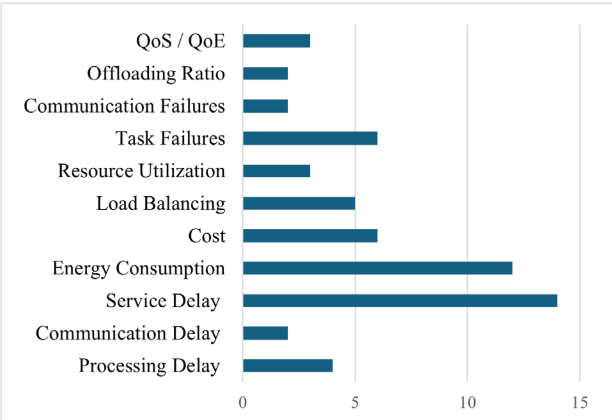

Fig. 10. Engagement of optimization objectives.

On one hand the paramount need to minimize service delay reflects the stringent latency requirements of emerging time-sensitive applications and mission-critical systems. On the other hand, the intense focus on energy consumption highlights the critical constraint of battery life in mobile IoE devices necessitating Green Computing strategies. Together these two metrics constitute the core delay and energy tradeoff that drives the majority of algorithmic designs.

Beyond these primary metrics a significant second tier of objectives has emerged led by Cost and Task Failures both with engagement levels around 6. The prominence of these metrics indicates that the field is maturing from purely theoretical performance maximization towards practical robustness and economic viability. The focus on minimizing task failures suggests a growing concern for system reliability and fault tolerance in unstable wireless environments while the emphasis on cost reflects the need to optimize the economic expenditure of renting cloud-edge resources.

Other metrics like Load Balancing Processing Delay and QoS/QoE appear with lower frequencies not because they are unimportant but because they are often treated as constraints within the broader objectives of minimizing total delay or energy.

5) *Comparative analysis of algorithmic properties*

It is crucial to acknowledge that a direct quantitative comparison of absolute metrics, such as energy consumption values or latency, is methodologically infeasible across the surveyed literature. This limitation stems from the significant heterogeneity in simulation platforms, network topologies, channel models, and baseline assumptions employed by different studies. Therefore, instead of providing a numerical comparison that lacks statistical validity, Table X presents a systematic qualitative synthesis of the inherent performance characteristics for each optimization paradigm. As detailed in the table, classical optimization approaches remain the primary standard for solution optimality and convergence speed in static sub-problems, yet they struggle with the scalability required for ultra-dense IoE networks. In contrast, heuristic and meta-heuristic methods offer a model-free alternative that effectively navigates non-convex solution spaces with moderate complexity, though often at the expense of guaranteed optimality. Similarly, game-theoretic models excel in capturing the interactive dynamics of multi-agent competition but face challenges in computational complexity when seeking equilibrium in large-scale systems. Notably, the analysis highlights a distinct trade-off where AI-native approaches demonstrate superior adaptability to non-stationary environments and high scalability during the inference phase, but they incur high upfront training complexity and strictly depend on data quality. This comparative insight underscores why hybrid paradigms, which aim to balance these conflicting dimensions, are emerging as the most promising direction for future research.

6) *Evolution of optimization paradigms and scenario adaptability*

The survey of the existing literature reveals that the evolution of task offloading and resource allocation algorithms in IoE networks is essentially a dynamic trade-off among computational accuracy, decision timeliness, and environmental uncertainty. As network architectures evolve from small-scale, static sensing systems toward cross-dimensional infrastructures with integrated space–air–ground–sea coverage, a single optimization paradigm is no longer sufficient to cope with the rapidly increasing system entropy.

In scenarios where channel conditions and network topologies remain relatively stable, classical convex optimization and Lyapunov stochastic optimization have respectively established theoretical performance bounds and long-term queue stability guarantees. However, with the advent of ultra-dense networks, the curse of dimensionality and intensified multi-user competition significantly exacerbate computational complexity. In this context, heuristic algorithms and game-theoretic approaches effectively alleviate NP-hard complexity by moderately sacrificing local optimality or adopting distributed decision-making mechanisms.

Furthermore, in highly dynamic and nonlinear environments such as vehicular networks and Space–Air–Ground Integrated Networks (SAGIN), traditional model-driven approaches gradually lose effectiveness due to modeling difficulties and solution latency. This has accelerated a paradigm shift toward data-driven, AI-native optimization, where deep reinforcement learning leverages millisecond-level inference capabilities to cope with highly dynamic environments.

Ultimately, future research trends are expected to converge toward hybrid and cooperative optimization frameworks, in which AI-based environment prediction is combined with mathematical model–based constraint solving. This complementary mechanism bridges the gap between model-driven and data-driven approaches, enabling globally coordinated optimization of latency, energy consumption, and reliability in heterogeneous, dynamic, and large-scale IoE ecosystems. Table XI presents the adaptability mapping between six optimization paradigms and representative task offloading scenarios.

In summary, the development of IoE optimization techniques follows a progression from analytical approaches to heuristic search, then to learning-driven strategies, and finally to collaborative hybrid methods. Classical and Lyapunov optimization provide theoretical foundations, heuristic and game-theoretic approaches enhance multi-dimensional decision-making, and AI-based and hybrid methods enable intelligent and self-organizing resource management. Regarding objectives, these techniques have evolved from focusing on single performance metrics such as delay or energy consumption to multi-objective joint optimization. In terms of architecture, solutions have shifted from centralized computation toward distributed and collaborative decision-making, reflecting the overall trend of IoE systems advancing toward self-optimization.

TABLE X. PERFORMANCE AND CHARACTERISTIC COMPARISON OF OPTIMIZATION PARADIGMS

Paradigm	Computational Complexity	Solution Optimality	Convergence Speed	Scalability (Large-scale)	Adaptability (Dynamic Env.)	Dependency on Prior Knowledge
Classic Opt.	Low / Medium	High (Global Optimal)	Fast	Low	Low	High (Need Model)
Lyapunov Opt.	Low	Medium (Near Optimal)	Fast	Medium	Medium	Medium
Heuristic	Medium	Medium (Local Optimal)	Medium	Medium	Low	Low (Model-free)
Game Theory	High	High (Nash Equilibrium)	Slow	Low	Medium	High (Information)
AI-Native	High (Training)/Low (Inference)	Medium/High	Slow (Training)	High	High	Low (Data-driven)
Hybrid	High	High	Medium	High	High	Medium

TABLE XI. ADAPTABILITY MATCHING MATRIX OF OPTIMIZATION ALGORITHMS AND IOE SCENARIOS

Typical Scenarios	1. Classic Opt.	2. Lyapunov Opt.	3. Heuristic	4. Game Theory	5. AI-based	6. Hybrid & Collab.
1. Small-scale Static	●	○	◑	○	○	○
2. Ultra-Dense Networks	○	◑	◑	●	◑	●
3. Wide-area Sparse	◑	○	●	◑	◑	◑
4. High-Mobility	○	◑	○	○	●	●
5. UAV-assisted Networks	●	◑	◑	◑	●	●
6. Device-Edge Synergy	◑	◑	◑	●	◑	●
7. Device-Edge-Cloud	○	◑	●	◑	◑	●
8. SAGIN (Space-Air-Ground)	○	◑	◑	○	●	●
9. Energy Harvesting	◑	●	○	◑	◑	●
10. Task Dependency (DAG)	◑	○	●	○	●	●

Note: ● High Suitability, ◑ Medium Suitability, ○ Low Suitability.

B. Open Issues

Although significant progress has been made in IoE task offloading and resource allocation optimization, several scientific and engineering challenges remain unresolved. These issues reflect key bottlenecks in the evolution of the field toward intelligent, self-organizing, and scalable systems. The main open problems can be summarized as follows:

1) System complexity and scalability bottlenecks

The continuous growth in IoE system scale and resource heterogeneity leads to an exponential increase in the state space. While classical centralized optimization provides theoretical optimality, it becomes computationally intractable in such high-dimensional spaces; similarly, game-theoretic approaches often suffer from prohibitive overhead when calculating Nash equilibrium for massive multi-agent systems.

Consequently, traditional centralized or single-layer optimization algorithms face the curse of dimensionality in ultra-large-scale networks. Even distributed and collaborative optimization frameworks, which alleviate some computational pressure, remain limited by communication overhead, synchronization delays, and model coupling complexity.

2) Learning challenges in dynamic and non-stationary environments

IoE environments exhibit pronounced non-stationary and non-ergodic characteristics. Task arrival rates, channel conditions, and device energy consumption fluctuate over time, causing the performance of algorithms based on static modeling or short-term objectives to degrade rapidly. Existing research lacks theoretical analysis of regret bounds under online learning, making it difficult to quantify learning efficiency in dynamic distributions. Furthermore, under non-stationary and small-sample conditions, reinforcement learning and neural optimization models are prone to overfitting or catastrophic forgetting, highlighting the need for more robust dynamic learning mechanisms.

3) Lack of theoretical interpretability and performance bounds

Most hybrid or AI-driven optimization algorithms remain at the empirical stage, lacking a unified theoretical analysis framework. Unlike Lyapunov or convex optimization, which offer rigorous mathematical proofs for stability and optimality gaps, emerging deep learning-based approaches operate largely as black boxes with opaque decision logic. Consequently, convergence, stability, and optimality are difficult to rigorously prove, and performance lower bounds and complexity measures are seldom quantified. This theoretical gap limits reproducibility and reliability and constrains deployment in safety-critical IoE systems.

4) Missing multi-objective trade-off modeling and pareto optimization

IoE task offloading often involves multiple conflicting objectives, such as energy consumption, latency, and system cost. Most existing studies rely on weighted or hierarchical strategies for approximate solutions and have not established a complete modeling framework for the dynamic Pareto frontier. Precisely characterizing and controlling the optimal trade-offs among energy, delay, and cost under dynamic constraints remains a key scientific problem for efficient resource coordination.

5) Insufficient cross-layer and cross-domain collaborative optimization

Optimization problems in IoE typically span multiple layers, including computing, communication, caching, and energy. Current research often focuses on single-layer or local decision-making, lacking a global collaborative mechanism across edge, end, and cloud layers. Cross-domain optimization is further complicated by resource competition and information asymmetry between subsystems. Developing unified cross-layer optimization frameworks that enable efficient coordination among distributed autonomous systems remains a core theoretical and engineering challenge.

6) Gaps in Human–Machine–Thing Collaborative Decision Modeling

IoE is evolving from machine interconnection to human–machine–thing collaborative intelligence. However, most current optimization models assume algorithm-driven decision-making, ignoring human behavior, cognitive delays, and social influence. Standard game-theoretic models typically assume rational agents with fixed utility functions, failing to account for the bounded rationality and social dynamics inherent in human participation. The lack of frameworks that integrate behavioral game theory and social-aware computing leaves human–machine hybrid decision-making and three-way human–machine–thing interaction largely unexplored.

7) Limited model generalization and transferability

AI-driven optimization algorithms often perform well in specific environments but struggle to generalize across different task distributions or network topologies. While heuristic algorithms offer some degree of robustness, data-driven models are particularly prone to overfitting specific network topologies, requiring retraining from scratch when the IoE scenario changes. The absence of effective transfer learning, meta-learning, or adaptive mechanisms limits the reusability and self-evolving capabilities of algorithms in multi-scenario IoE systems. Developing approaches for rapid transfer and continual learning is crucial for sustainable intelligent optimization in IoE.

In summary, the main open issues in IoE optimization include system scalability, dynamic non-stationary learning, theoretical interpretability, multi-objective Pareto optimization, cross-layer collaborative mechanisms, human–machine integrated decision-making, and model generalization.

C. Future Research Directions

Building on the preceding analysis, future research on IoE task offloading and resource allocation optimization is expected to advance toward intelligent, explainable, adaptive, and collaborative frameworks. The following seven directions represent key potential breakthroughs and emerging trends in the field:

1) Scalable and structured optimization frameworks for ultra-large-scale IoE systems

As the scale of IoE nodes and resource dimensions continues to expand, overcoming the curse of dimensionality becomes critical. Research can focus on graph coarsening and super-node abstraction to compress the state space and enable structured modeling. Coupling these approaches with message-passing-based distributed reinforcement learning frameworks, such as DGN and GQNN, can facilitate global collaborative decision-making through local information propagation. In addition, hierarchical multi-agent optimization architectures and lightweight communication mechanisms will be essential for real-time optimization in ultra-large-scale IoE networks.

2) Online learning and dynamic convergence theory for non-stationary environments

In dynamic IoE scenarios, system states exhibit non-stationary and non-ergodic characteristics, for which the theoretical guarantees of traditional reinforcement learning remain insufficient. Future research should establish regret lower bounds and dynamic policy gradient convergence conditions for online optimization. In particular, within the cloud, edge, and end three-tier architecture, it is essential to explore the stability analysis and theoretical convergence limits of hierarchical policy gradients. Moreover, integrating elastic memory and meta-learning mechanisms could enable models to achieve adaptive iteration and maintain long-term performance under non-stationary environments.

3) Theory and data fusion for explainable optimization systems

Future intelligent optimization in IoE should evolve from black box strategies toward operational knowledge. By embedding Lyapunov stability, convex constraints, and game equilibrium conditions into deep reinforcement learning frameworks, it is possible to achieve explainable optimization, ensuring that policy learning processes remain consistent with underlying physical or theoretical principles. Furthermore, integrating symbolic reasoning, causal graph modeling, and visual decision logic can enhance the verifiability and transparency of optimization algorithms, enabling the transition from black box models to knowledge-driven systems.

4) Multi-objective pareto optimization and self-balancing mechanisms among energy consumption, latency, and cost

Future IoE systems need to achieve dynamic optimal balance across multiple objectives. Research can leverage evolutionary multi-objective optimization, hierarchical reinforcement learning, or policy transfer mechanisms to approximate the Pareto frontier among energy consumption, latency, and cost. In addition, developing utility function learning frameworks for multi-objective mapping can enable adaptive trade-offs by learning dynamic weights from user preferences and system constraints, thereby maximizing overall QoS satisfaction.

5) Human machine thing collaborative intelligence and behavior digital twin integrated optimization

Future IoE systems are expected to evolve toward a stage of human machine thing collaborative intelligence. To achieve genuine collaborative intelligence, a dual cycle architecture that integrates behavioral and digital twin models should be established. By inferring human utility functions from social IoT and human machine interaction data, it becomes possible to couple social behaviors with task offloading optimization, enabling joint incentive and offloading decision making. Meanwhile, online twin consistency correction should be explored to dynamically align the virtual and physical systems, ensuring real time coherence between them. This direction will drive the IoE paradigm from mechanical self-organization toward intelligent autonomy with social awareness and human behavior understanding.

6) Cross layer and cross domain collaborative optimization framework

In the integrated cloud-edge-end architecture, it is essential to develop verifiable hierarchical optimization theories and cross domain resource collaboration mechanisms. Future research should focus on establishing a joint framework that combines distributed reinforcement learning and hierarchical game theory to analyze convergence conditions and stability of multi-level control strategies, particularly considering differences in time scales and task dependencies across layers. By constructing semantic mappings of cross domain resources and dynamic task offloading protocols, autonomous coordination and joint optimization among heterogeneous subsystems can be achieved.

7) Zero sample and few sample transfer learning mechanisms

In IoE systems, environmental conditions change frequently and data collection is often costly, creating an urgent need for zero sample and few sample transfer learning strategies to enable rapid cross scenario deployment and low-cost adaptation. By integrating meta learning, self-distillation, and experience reuse mechanisms, the system can quickly generate transferable initial policies during task switching, significantly reducing training overhead while enhancing self-evolution capability.

In summary, the future of IoE optimization is expected to transition from static modeling to dynamic self-evolution, from black-box learning to explainable knowledge-driven approaches, and from individual intelligence to human–machine–thing collaborative intelligence. Theoretically, multi-level convergence analysis and actionable knowledge modeling will bridge the gap between AI and optimization theory; methodologically, distributed learning, graph optimization, and digital twin technologies will enable system-level adaptability; and practically, IoE systems will advance toward cognitive, resilient, and sustainable optimization capabilities.

VII. Conclusion

This paper provides a comprehensive review of major research progress in task offloading and resource allocation optimization for the Internet of Everything (IoE), transcending previous siloed studies by offering an ecosystem perspective and a clear algorithmic roadmap for addressing the complexities of this domain. Focusing on the characteristics of high heterogeneity, strong dynamics, and multi objective constraints, the study summarizes the theoretical foundations and performance features of several optimization paradigms, including classical optimization, Lyapunov optimization, heuristic optimization, game theoretic optimization, AI native optimization, and hybrid collaborative optimization.

Our quantitative analysis reveals a decisive evolutionary trajectory: while classical methods remain fundamental, there is a distinct paradigm shift toward data-driven intelligence. Specifically, the emergence of hybrid and collaborative optimization as a rapidly growing research focus confirms the necessity of merging theoretical stability with AI adaptability. However, IoE optimization still faces multiple challenges, including limited scalability in ultra-dense networks, weak theoretical support for learning in non-stationary environments, and insufficient interpretability. Crucially, the unique "People" and "Process" dimensions of IoE remain under-optimized, with a notable gap in human-machine collaborative decision-making frameworks.

Future research must bridge these gaps by shifting from single algorithmic designs toward systematic intelligent decision-making. Emphasis should be placed on theory-data fusion for explainability, digital twin-driven behavioral modeling, and cross-layer autonomous control. Consequently, IoE networks will evolve from static connectivity into cognitive ecosystems capable of learning, reasoning, and self-organization, providing fundamental support for the next generation of intelligent social infrastructure.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Dayong Wang and Liping Lei collected the relevant articles; Dayong Wang, Liping Lei, Zhen Wang, and Cui Cui analyzed and compared the collected studies; Dayong Wang and Zhen Wang drafted the initial version of the manuscript, while Liping Lei and Cui Cui refined the writing and optimized the organization of the content. All authors had approved the final version.

References

[1] V. C. F. Costa, L. Oliveira, and J. Souza, "Internet of Everything (IoE) Taxonomies: A survey and a novel knowledge-based taxonomy," *Sensors*, vol. 21, no. 2, Jan. 2021. doi: 10.3390/s21020568

[2] X. Kong, Y. Wu, H. Wang, and F. Xia, "Edge computing for internet of everything: A survey," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23472–23485, Dec. 2022. doi: 10.1109/JIOT.2022.3200431

[3] W. Z. Khan, W. Rafique, N. Haider, S. Hakak, and M. Imran, "Internet of everything: Enabling technologies, applications, security and challenges," *Authorea Preprints*, 2023.

[4] B. Brik, K. Dev, Y. Xiao, G. Han, and A. Ksentini, "Guest editorial introduction to the special section on AI-powered Internet of Everything (IoE) services in next-generation wireless networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 2952–2954, Sept. 2022. doi: 10.1109/TNSE.2022.3195385

[5] X. Zhou *et al.*, "Edge-enabled two-stage scheduling based on deep reinforcement learning for internet of everything," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3295–3304, Feb. 2023. doi: 10.1109/JIOT.2022.3179231

[6] F. M. Ali, N. A. M. Yunus, N. N. Mohamed, M. M. Daud, and E. A. Sundararajan, "A systematic mapping: Exploring internet of everything technologies and innovations," *Symmetry*, vol. 15, no. 11, 1964, Nov. 2023. doi: 10.3390/sym15111964

[7] J. H. Ijaz, M. Shojafar, K. Munir, and R. Buyya, "Resource allocation and task scheduling in fog computing and internet of everything environments: A taxonomy, review, and future directions," *ACM Comput. Surv.*, vol. 54, no. 11, 2022.

[8] S. H. Alsamhi *et al.*, "Towards sustainable industry 4.0: A survey on greening IoE in 6G networks," *Ad Hoc Netw.*, vol. 165, 103610, Dec. 2024. doi: 10.1016/j.adhoc.2024.103610

[9] Z. Xue, W. Guo, Z. Cui, and W. Zhang, "The global evaluation strategy for many-objective partial collaborative computation offloading problem," *Concurr. Comput. Pract. Exp.*, vol. 35, no. 2, e7474, 2023. doi: 10.1002/cpe.7474

[10] M. Tang and V. W. S. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Trans. Mob. Comput.*, vol. 21, no. 6, pp. 1985–1997, June 2022. doi: 10.1109/TMC.2020.3036871

[11] G. Ma, W. Li, W. Fang, Q. Fu, and Z. Zhou, "A new computing offload algorithm for maximizing user revenue under cloud edge architecture," in *Proc. 3rd International Conference on Cognitive Based Information Processing and Applications*, 2024, pp. 437–450. doi: 10.1007/978-981-97-1979-2_38

[12] D. Souza, G. Iwashima, V. C. Farias da Costa, C. E. Barbosa, J. M. de Souza, and G. Zimbrão, "Architectural Trends in Collaborative Computing: Approaches in the Internet of Everything Era," *Future Internet*, vol. 16, no. 12, 445, Dec. 2024. doi: 10.3390/fi16120445

[13] C. Serôdio, J. Cunha, G. Candela, S. Rodriguez, X. R. Sousa, and F. Branco, "The 6G ecosystem as support for IoE and private networks: Vision, requirements, and challenges," *Future Internet*, vol. 15, no. 11, 348, Nov. 2023. doi: 10.3390/fi15110348

[14] G. Nagabushnam, Y. Choi, and K. H. Kim, "Energizing the fog: a systematic survey on task scheduling strategies for energy optimization," *Clust. Comput.*, vol. 28, no. 6, 375, June 2025. doi: 10.1007/s10586-025-05200-8

[15] D. G. Zhang *et al.*, "New computing tasks offloading method for MEC based on prospect theory framework," *IEEE Trans. Comput. Soc. Syst.*, vol. 11, no. 1, pp. 770–781, Feb. 2024. doi: 10.1109/TCSS.2022.3228692

[16] J. Zhang, D. G. Zhang, M. Qiao, E. H. Lin, T. Zhang, and P. Zhang, "New offloading method of computing task based on gray wolf hunting optimization mechanism for the IOV," *IEEE Trans. Netw. Serv. Manag.*, vol. 22, no. 3, pp. 2264–2277, June 2025. doi: 10.1109/TNSM.2025.3539865

[17] T. Kesavan V, V. R, W. K. Wong, and P. K. Ng, "Reinforcement learning based secure edge enabled multi task scheduling model for internet of everything applications," *Sci. Rep.*, vol. 15, no. 1, 6254, Feb. 2025. doi: 10.1038/s41598-025-89726-2

[18] Q. Qi, T. Shi, K. Qin, and G. Luo, "Completion time optimization in UAV-relaying-assisted MEC networks with moving users," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1246–1258, Feb. 2024. doi: 10.1109/TCE.2023.3278470

[19] S. P. Singh, N. Kumar, A. Singh, K. Kant Singh, S. S. Askar, and M. Abouhawwash, "Energy efficient hybrid evolutionary algorithm for Internet of Everything (IoE)-enabled 6G," *IEEE Access*, vol. 12, pp. 63839–63852, 2024. doi: 10.1109/ACCESS.2024.3390939

[20] Z. Cao, X. Deng, S. Yue, P. Jiang, J. Ren, and J. Gui, "Dependent task offloading in edge computing using GNN and deep reinforcement learning," *IEEE Internet Things J.*, vol. 11, no. 12, pp. 21632–21646, June 2024. doi: 10.1109/JIOT.2024.3374969

[21] S. U. Jamil, M. A. Khan, and S. U. Rehman, "Edge computing enabled technologies for secure 6G smart environment-an overview," in *Proc. 12th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2020)*, 2021, pp. 934–945. doi: 10.1007/978-3-030-73689-7_88

[22] J. Mika and A. Juha, "Advancing smart city infrastructures through the real-world implementation of 6G technologies," *Fusion Multidiscip. Res. Int. J.*, vol. 6, no. 2, pp. 796–811, Aug. 2025. doi: 10.63995/WICJ7106

[23] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020. doi: 10.1109/JIOT.2020.2984887

[24] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May 2020. doi: 10.1109/MNET.001.1900287

[25] K. S. Reddy, K. Agarwal, and A. K. Tyagi, "Beyond things: A systematic study of internet of everything," in *Innovations in Bio-*

Inspired Computing and Applications, pp. 226–242, 2021. doi: 10.1007/978-3-030-49339-4_23

[26] R. N. Gupta, and S. K. Dhurandher, "Efficient caching method in fog computing for internet of everything," *Peer-Peer Netw. Appl.*, vol. 14, no. 1, pp. 439–452, Jan. 2021. doi: 10.1007/s12083-020-00952-z

[27] Z. Nezami and K. Zamanifar, "Internet of things / internet of everything: Structure and ingredients," *IEEE Potentials*, vol. 38, no. 2, pp. 12–17, Mar. 2019. doi: 10.1109/MPOT.2018.2855439

[28] P. K. Padhi and F. C. Santos, "6G enabled industrial internet of everything: Towards a theoretical framework," *Appl. Syst. Innov.*, vol. 4, no. 1, p. 11, Mar. 2021. doi: 10.3390/asi4010011

[29] M. Shojafar and M. Sookhak, "Internet of Everything, Networks, Applications, and Computing Systems (IoENACS)," *Int. J. Comput. Appl.*, vol. 42, no. 3, pp. 213–215, Apr. 2020. doi: 10.1080/1206212X.2019.1575621

[30] V. Avula, R. Nanditha, S. Dhuli, and P. Ranjan, "The internet of everything: A survey," in *Proc. 2021 13th International Conference on Computational Intelligence and Communication Networks (CICN)*, Sept. 2021, pp. 72–79. doi: 10.1109/CICN51697.2021.9574695

[31] Z. M. Jawad Kubba and H. K. Hoomod, "The internet of everything based smart systems: Applications and challenges," in *Proc. 2019 1st AL-Noor International Conference for Science and Technology (NICST)*, Oct. 2019, pp. 58–62. doi: 10.1109/NICST49484.2019.9043785

[32] A. Shilpa, V. Muneeswaran, D. D. K. Rathinam, G. A. Santhiya, and J. Sherin, "Exploring the Benefits of Sensors in Internet of Everything (IoE)," in *Proc. 2019 5th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Mar. 2019, pp. 510–514. doi: 10.1109/ICACCS.2019.8728530

[33] T. Ghosh, A. Roy, and S. Misra, "I2M: Intelligent information management for rendering IoE services in society 5.0," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 2955–2965, Sept. 2022. doi: 10.1109/TNSE.2022.3174730

[34] Z. Lv and N. Kumar, "Software defined solutions for sensors in 6G/IoE," *Comput. Commun.*, vol. 153, pp. 42–47, Mar. 2022. doi: 10.1016/j.comcom.2020.01.060

[35] Y. Zheng, N. Wei, and Y. Liu, "Collaborative computation for offloading and caching strategy using intelligent edge computing," *Mob. Inf. Syst.*, vol. 2022, no. 1, 4840801, 2022. doi: 10.1155/2022/4840801

[36] P. Kohli, S. Sharma, and P. Matta, "Secured privacy preserving techniques analysis of 6G driven vehicular communication network in industry 5.0 Internet-of-Everything (IoE) Applications," in *Proc. 2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, Dec. 2022, pp. 1–8. doi: 10.1109/SMARTGENCON56628.2022.10084289

[37] A. Mahapatra, K. Mishra, R. Pradhan, and S. K. Majhi, "Next generation task offloading techniques in evolving computing paradigms: Comparative analysis, current challenges, and future research perspectives," *Arch. Comput. Methods Eng.*, vol. 31, no. 3, pp. 1405–1474, Apr. 2024. doi: 10.1007/s11831-023-10021-2

[38] T. Saba, A. Rehman, K. Haseeb, T. Alam, and G. Jeon, "Cloud-edge load balancing distributed protocol for IoE services using swarm intelligence," *Clust. Comput.*, vol. 26, no. 5, pp. 2921–2931, Oct. 2023. doi: 10.1007/s10586-022-03916-5

[39] H. Guo, J. Liu, and J. Lv, "Toward intelligent task offloading at the edge," *IEEE Netw.*, vol. 34, no. 2, pp. 128–134, Mar. 2020. doi: 10.1109/MNET.001.1900200

[40] Q. Li, J. Zhao, Y. Gong, and Q. Zhang, "Energy-efficient computation offloading and resource allocation in fog computing for Internet of Everything," *China Commun.*, vol. 16, no. 3, pp. 32–41, Mar. 2019. doi: 10.12676/j.cc.2019.03.004

[41] F. Saeik *et al.*, "Task offloading in edge and cloud computing: A survey on mathematical, artificial intelligence and control theory solutions," *Comput. Netw.*, vol. 195, 108177, Aug. 2021. doi: 10.1016/j.comnet.2021.108177

[42] C. Bu and J. Wang, "Computing tasks assignment optimization among edge computing servers via SDN," *Peer--Peer Netw. Appl.*, vol. 14, no. 3, pp. 1190–1206, May 2021. doi: 10.1007/s12083-021-01081-x

[43] J. Hu, T. Shui, L. Xiang, and K. Yang, "Multi-domain resource scheduling for simultaneous wireless computing and power transfer in fog radio access network," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 269–281, Jan. 2023. doi: 10.1109/TCOMM.2022.3225440

[44] A. Zhong, Z. Li, D. Wu, T. Tang, and R. Wang, "Stochastic peak age of information guarantee for cooperative sensing in internet of everything," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15186–15196, Sept. 2023. doi: 10.1109/JIOT.2023.3264826

[45] G. Manogaran and B. S. Rawal, "An efficient resource allocation scheme with optimal node placement in IoT-fog-cloud architecture," *IEEE Sens. J.*, vol. 21, no. 22, pp. 25106–25113, Nov. 2021. doi: 10.1109/JSEN.2021.3057224

[46] R. Yamuna and M. U. Rani, "Priority based task scheduling and delay optimization in mobile edge computing," *Int. J. Comput. Eng. Res. Trends*, vol. 9, no. 1, pp. 1−6, 2022.

[47] L. Song, G. Sun, H. Yu, and M. Guizani, "SD-AETO: Service Deployment-enabled Adaptive Edge Task Offloading Scheme in MEC," *IEEE Internet Things J.*, pp. 1–1, 2023. doi: 10.1109/JIOT.2023.3281603

[48] B. Picano and R. Fantacci, "End-to-end delay bound analysis for VR and industrial IoE traffic flows under different scheduling policies in a 6G network," *Computers*, vol. 12, no. 3, Mar. 2023. doi: 10.3390/computers12030062

[49] A. Lakhan *et al.*, "BEDS: Blockchain energy efficient IoE sensors data scheduling for smart home and vehicle applications," *Appl. Energy*, vol. 369, 123535, Sept. 2024. doi: 10.1016/j.apenergy.2024.123535

[50] M. M. Salim, Y. Pan, and J. H. Park, "Energy-efficient resource allocation in blockchain-based cybertwin-driven 6G," *J. Ambient Intell. Humaniz. Comput.*, vol. 15, no. 1, pp. 103–114, Jan. 2024. doi: 10.1007/s12652-022-03877-y

[51] G. Chen, Q. Wu, W. Chen, D. W. K. Ng, and L. Hanzo, "IRS-aided wireless powered mec systems: TDMA or NOMA for computation offloading," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 2, pp. 1201–1218, Feb. 2023. doi: 10.1109/TWC.2022.3203158

[52] Y. Xu, L. Chen, Z. Lu, X. Du, J. Wu, and P. C. K. Hung, "An adaptive mechanism for dynamically collaborative computing power and task scheduling in edge environment," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3118–3129, Feb. 2023. doi: 10.1109/JIOT.2021.3119181

[53] X. Gu, G. Zhang, W. Duan, S. Dang, M. Wen, and P. H. Ho, "Computing offloading for RIS-aided internet of everything: A cybertwin version," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 20443–20456, June 2024. doi: 10.1109/JIOT.2024.3370557

[54] W. Shu, X. Feng, and C. Guo, "An adaptive alternating direction method of multipliers for vehicle-to-everything computation offloading in cloud–edge collaborative environment," *IEEE Internet Things J.*, vol. 11, no. 22, pp. 35802–35811, Nov. 2024. doi: 10.1109/JIOT.2024.3360134

[55] Y. Guo, Y. Sang, B. Wang, and Y. Xu, "Collaborative diversified caching strategies for enhanced performance in edge server networks," in *Proc. 2025 2nd International Conference on Algorithms, Software Engineering and Network Security (ASENS)*, Mar. 2025, pp. 282–285. doi: 10.1109/ASENS64990.2025.11011030

[56] J. Tang, J. Nie, J. Zhao, Y. Zhou, Z. Xiong, and M. Guizani, "Slicing-based software-defined mobile edge computing in the air," *IEEE Wirel. Commun.*, vol. 29, no. 1, pp. 119–125, Feb. 2022. doi: 10.1109/MWC.001.2100303

[57] K. Luo, Y. Wang, Y. Liu, and K. Zhu, "Collaborative integration of vehicle and roadside infrastructure sensor for temporal dependency-aware task offloading in the internet of vehicles," *Int. J. Intell. Syst.*, vol. 2025, no. 1, 8064086, 2025. doi: 10.1155/int/8064086

[58] I. Aqeel *et al.*, "Load balancing using artificial intelligence for cloud-enabled internet of everything in healthcare domain," *Sensors*, vol. 23, no. 11, 5349, Jan. 2023. doi: 10.3390/s23115349

[59] L. Altin, H. R. Topcuoglu, and F. S. Gürgen, "Network congestion aware multiobjective task scheduling in heterogeneous fog environments," *IEEE Trans. Ind. Inform.*, vol. 20, no. 2, pp. 3015–3024, Feb. 2024. doi: 10.1109/TII.2023.3299624

[60] W. Li and J. Ou, "Machine scheduling with restricted rejection: An Application to task offloading in cloud–edge collaborative computing," *Eur. J. Oper. Res.*, vol. 314, no. 3, pp. 912–919, May 2024. doi: 10.1016/j.ejor.2023.11.002

[61] S. Tian, Y. Zhang, Y. Bi, and T. Yuan, "Blockchain-based 6G task offloading and cooperative computing resource allocation study," *J. Cloud Comput.*, vol. 13, no. 1, p. 95, May 2024. doi: 10.1186/s13677-024-00655-3

[62] S. Maghsudi and M. Davy, "Computational models of human decision-making with application to the internet of everything," *IEEE Wirel. Commun.*, vol. 28, no. 1, pp. 152–159, Feb. 2021. doi: 10.1109/MWC.001.2000250

[63] H. Wang and J. An, "Dynamic game based task offloading and resource pricing in LEO-multi-access edge computing," *Computing*, vol. 106, no. 2, pp. 579–606, Feb. 2024. doi: 10.1007/s00607-023-01234-1

[64] L. Agrawal, A. Mondal, and M. S. Obaidat, "T-RESIN: Throughput-aware dynamic resource orchestration for Internet of Everything-enabled software-defined edge networks," *Int. J. Commun. Syst.*, vol. 37, no. 12, e5802, 2024. doi: 10.1002/dac.5802

[65] X. Zhong and Y. He, "A cybertwin-driven task offloading Scheme based on deep reinforcement learning and graph attention networks," in *Proc. 2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct. 2021, pp. 1–6. doi: 10.1109/WCSP52459.2021.9613687

[66] D. S. Lakew, V. D. Tuong, N. N. Dao, and S. Cho, "Adaptive partial offloading and resource harmonization in wireless edge computing-assisted IoE networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3028–3044, Sept. 2022. doi: 10.1109/TNSE.2022.3153172

[67] L. Wang and G. Zhang, "Deep reinforcement learning based joint partial computation offloading and resource allocation in mobility-aware MEC system," *China Commun.*, vol. 19, no. 8, pp. 85–99, Aug. 2022. doi: 10.23919/JCC.2022.08.007

[68] S. U. Jamil, M. A. Khan, and S. U. Rehman, "Resource allocation and task off-loading for 6G enabled smart edge environments," *IEEE Access*, vol. 10, pp. 93542–93563, 2022. doi: 10.1109/ACCESS.2022.3203711

[69] X. Huang, K. Zhang, F. Wu, and S. Leng, "Collaborative machine learning for energy-efficient edge networks in 6G," *IEEE Netw.*, vol. 35, no. 6, pp. 12–19, Nov. 2021. doi: 10.1109/MNET.100.2100313

[70] Z. Tong, B. Liu, J. Mei, J. Wang, W. Li, and K. Li, "D2OP: A fair dual-objective weighted scheduling scheme in Internet of Everything," *IEEE Internet Things J.*, pp. 1–1, 2023. doi: 10.1109/JIOT.2023.3234078

[71] J. Zheng, T. H. Luan, Y. Zhang, G. Li, Z. Su, and W. Wu, "Digital twin in 6G: Embracing comprehensive network intelligence," *IEEE Wirel. Commun.*, vol. 31, no. 6, pp. 94–101, Dec. 2024. doi: 10.1109/MWC.017.2300604

[72] M. Adhikari, A. Munusamy, N. Kumar, and S. N. Srirama, "Cybertwin-driven resource provisioning for IoE applications at 6G-Enabled Edge Networks," *IEEE Trans. Ind. Inform.*, vol. 18, no. 7, pp. 4850–4858, July 2022, doi: 10.1109/TII.2021.3096672.

[73] M. Wu *et al.*, "Integrated Resource Collaboration for RIS-Assisted Digital-Twin-Empowered Internet of Everything," *IEEE Internet Things J.*, vol. 12, no. 13, pp. 23275–23287, July 2025. doi: 10.1109/JIOT.2025.3551694

[74] H. Byeon *et al.*, "Consumer technology in task offloading and edge resource allocation: AIoT and edge computing for next-generation communication," *IEEE Trans. Consum. Electron.*, vol. 71, no. 2, pp. 5356–5365, May 2025. doi: 10.1109/TCE.2025.3552205

[75] S. Pang, L. Wang, H. Gui, S. Qiao, X. He, and Z. Zhao, "UAV-IRS-assisted energy harvesting for edge computing based on deep reinforcement learning," *Future Gener. Comput. Syst.*, vol. 163, 107527, Feb. 2025. doi: 10.1016/j.future.2024.107527

[76] N. A and P. Ks, "Efficient joint resource allocation using self-organized map based deep reinforcement learning for cybertwin enabled 6G networks," *Sci. Rep.*, vol. 15, no. 1, 19795, June 2025. doi: 10.1038/s41598-025-02274-7

[77] G. Manogaran, T. N. Nguyen, J. Gao, and P. M. Kumar, "Deep learning-based service distribution model for wireless network assisted internet of everything," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3004–3014, Sept. 2022. doi: 10.1109/TNSE.2022.3155943

[78] A. Asheralieva and D. Niyato, "Optimizing age of information and security of the next-generation internet of everything systems," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20331–20351, Oct. 2022. doi: 10.1109/JIOT.2022.3173912

[79] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading Latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12240–12251, Oct. 2020. doi: 10.1109/TVT.2020.3018817

[80] G. Tefera, K. She, M. Chen, and A. Ahmed, "Congestion-aware adaptive decentralised computation offloading and caching for multi-access edge computing networks," *IET Commun.*, vol. 14, no. 19, pp. 3410–3419, 2020. doi: 10.1049/iet-com.2020.0630

[81] H. Sami, H. Otrok, J. Bentahar, and A. Mourad, "AI-based resource provisioning of IoE services in 6g: A deep reinforcement learning approach," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 3, pp. 3527–3540, Sept. 2021. doi: 10.1109/TNSM.2021.3066625

[82] D. K. Jain, S. K. S. Tyagi, S. Neelakandan, M. Prakash, and L. Natrayan, "Metaheuristic optimization-based resource allocation technique for cybertwin-driven 6G on IoE environment," *IEEE Trans. Ind. Inform.*, vol. 18, no. 7, pp. 4884–4892, July 2022. doi: 10.1109/TII.2021.3138915

[83] K. Sadatdiynov, L. Cui, L. Zhang, J. Z. Huang, N. N. Xiong, and C. Luo, "An intelligent hybrid method: Multi-objective optimization for MEC-enabled devices of IoE," *J. Parallel Distrib. Comput.*, vol. 171, pp. 1–13, Jan. 2023. doi: 10.1016/j.jpdc.2022.09.008.

[84] J. Chen, X. Li, L. Pan, and S. Liu, "A lyapunov optimization-based online algorithm for scheduling cloud-edge collaborative real-time video stream analytics tasks," *IEEE Internet Things J.*, vol. 12, no. 14, pp. 27713–27727, July 2025. doi: 10.1109/JIOT.2025.3563576

[85] P. Qin, Q. Li, and D. Xu, "Decentralized federated learning in LEO satellite-based IoE communications: Latency optimization under reliability constraints," *IEEE Internet Things J.*, pp. 1–1, 2025. doi: 10.1109/JIOT.2025.3557128

[86] J. Chen, Y. Wu, X. Huangfu, W. Ma, and H. Zhou, "MAPPO-assisted joint task offloading and resource allocation for energy efficiency in dynamic noma-mec systems," in *Proc. 2025 7th International Conference on Artificial Intelligence Technologies and Applications (ICAITA)*, June 2025, pp. 1–10. doi: 10.1109/ICAITA67588.2025.11137900

[87] Z. Tong, X. Deng, Y. Zhang, J. Mei, C. Wang, and K. Li, "MADDPG-based task offloading and resource pricing in edge collaboration environment," *J. Syst. Archit.*, vol. 165 103433, Aug. 2025. doi: 10.1016/j.sysarc.2025.103433

[88] D. V. Huynh, S. R. Khosravirad, V. Sharma, J. Kim, B. Canberk, and T. Q. Duong, "Carbon-aware edge computing for internet of everything networks: A digital twin approach," *IEEE Internet Things J.*, vol. 12, no. 15, pp. 29240–29251, Aug. 2025. doi: 10.1109/JIOT.2025.3564157