A QoS-Aware Resource Allocation Method for Internet of Things using Ant Colony Optimization Algorithm and Tabu Search

Shuling YIN[1], Renping YU[2]
Hubei Open University, Wuhan 430074, China[1]
Hubei Huazhong Electric Power Technology Development Co., Ltd.[2]

***Abstract*—In today's computing era, the Internet of Things (IoT) stands out for its implementation of automation, high-quality ecosystems, creative and efficient services, and higher productivity. IoT has found applications in various fields, such as education, healthcare, agriculture, military, and industry, where diverse resource requirements present a major challenge. To address this issue, we propose a novel QoS-aware resource allocation method for IoT systems. Our approach combines the Ant Colony Optimization (ACO) and Tabu Search (TS) algorithms to manage resources effectively, minimize energy consumption, reduce communication delays, and enhance overall system performance. Experimental results demonstrate the efficiency and effectiveness of our approach, with significant improvements in QoS metrics compared to traditional methods. By merging ACO and TS algorithms, our research contributes to the advancement of IoT capabilities, energy conservation, and business optimization.**

***Keywords*—Internet of things; resource allocation; virtualization; Ant Colony Optimization; Tabu Search**

I. Introduction

The Internet of Things (IoT) enables the integration of the virtual and physical worlds, facilitating communication between various devices without human intervention [1]. This has led to a growing interest in IoT research due to its ability to enable intelligent and ubiquitous services through data aggregation, processing, analysis, and mining [2, 3]. However, the performance of IoT systems is influenced by several factors and resources, such as user requirements, energy consumption, diverse applications, storage capacity, communication needs, network bandwidth, and computing power. These resources are heterogeneous in nature, meaning they vary in their capabilities and characteristics [4]. IoT networks face resource allocation challenges, particularly in networks with heterogeneous properties. Resource allocation involves effectively managing and allocating limited resources to achieve optimal objectives [5]. The resources in IoT networks are divided into two categories: node resources and channel resources. Node resources, also known as physical resources, include storage, computational power, and energy resources. On the other hand, channel resources pertain to communication channels and networks, encompassing aspects such as channel bandwidth, load balancing, and traffic analysis [6].

Resource allocation refers to the task of efficiently assigning available resources to complete a set of tasks while considering specific conditions and constraints [7]. The target is to optimize resource utilization and enhance the performance of the IoT platform. The resources in IoT devices are often limited due to factors such as energy constraints, processing power, and storage capacity [8]. However, IoT devices have the potential to provide various services and functionalities. Efficient resource allocation is crucial for optimizing the utilization of these limited resources and ensuring that tasks are completed effectively [9]. The heterogeneous and distributed characteristics of the devices and resources complicate ioT resource allocation. IoT devices come in different types with varying capabilities and characteristics. They may have different energy levels, processing capacities, and storage capacities. The resource allocation algorithm needs to consider these differences and allocate resources accordingly [10].

Integrating machine learning, deep learning, Artificial Intelligence (AI), and urban public transportation systems plays a pivotal role in efficiently allocating resources in the IoT. These technologies collectively form the backbone of intelligent resource management in urban environments. Machine learning algorithms enable IoT systems to adapt and optimize resource allocation strategies by analyzing vast amounts of data [11, 12]. Deep learning, a subset of machine learning, excels in pattern recognition and feature extraction, making it invaluable for understanding complex urban dynamics [13, 14]. AI systems bring decision-making capabilities to IoT devices, allowing them to dynamically adjust resource allocation based on real-time conditions and user preferences [15, 16]. Urban public transportation systems are a prime example of IoT in action, encompassing connected vehicles, smart traffic management, and passenger information systems. By leveraging the data generated within these transportation networks, machine learning models can predict traffic patterns, optimize routes, and enhance energy efficiency, leading to reduced congestion and environmental impact [17].

In recent years, researchers have turned to nature-inspired meta-heuristic algorithms to tackle optimization problems in various domains, including IoT resource allocation. These algorithms mimic natural phenomena and use techniques such as solution perturbations and stochasticity to avoid local optima and achieve optimal or near-optimal solutions. Meta-heuristic optimization algorithms have gained popularity due to their ability to handle various applications and optimization problems. These meta-heuristic algorithms can be applied to IoT resource allocation problems to meet various objectives, such as energy efficiency, bandwidth utilization, and task allocation. Previous research efforts have explored different optimization techniques for resource allocation in IoT systems. Genetic, ACO, and Particle Swarm Optimization (PSO) algorithms are some of the methods applied to address this challenge. However, these approaches often suffer from slow convergence or suboptimal solutions, particularly when dealing with complex, combinatorial optimization problems that arise in IoT resource allocation scenarios. Therefore, it is necessary to develop more efficient and effective algorithms to cope with the complexity and variability of IoT environments.

This paper proposes a novel QoS-aware resource allocation method for IoT systems that utilizes a hybrid approach that incorporates both the Ant Colony Optimization (ACO) and the Tabu Search (TS) algorithms. The ACO algorithm is derived from the foraging behavior of ants and effectively solves combinatorial optimization problems. It employs a population of artificial ants to construct solutions by probabilistically selecting resources based on pheromone trails and heuristics. On the other hand, the TS algorithm is a local search-based metaheuristic that intensifies the search process by maintaining a tabu list, preventing revisiting previously visited solutions and encouraging the exploration of new solutions. By combining these two powerful optimization approaches, we aim to overcome the limitations of conventional resource allocation methods and provide a more efficient and effective solution regarding QoS-aware resource allocation in IoT systems.

In this context, our motivation for this research lies in the critical need for efficient resource allocation in IoT systems. Managing limited resources becomes paramount as IoT applications continue to increase across diverse domains such as education, healthcare, agriculture, military, and industry. IoT systems rely on a multitude of resources, including but not limited to storage, computational power, and energy. These resources are heterogeneous and often constrained, posing significant challenges to resource allocation. Our proposed approach addresses these challenges by prioritizing Quality of Service (QoS) in resource allocation. The potential benefits of our research are multifaceted. By effectively managing resources, our method aims to minimize energy consumption, reduce communication delays, and enhance overall system performance. This not only leads to improved operational efficiency but also contributes to sustainability efforts by reducing energy usage. Furthermore, our approach holds the promise of enabling more reliable and responsive IoT applications. As IoT plays an increasingly integral role in critical domains such as healthcare and industrial automation, optimizing resource allocation can directly impact service quality and reliability. Our research makes several significant contributions to the field of IoT resource allocation:

- Novel hybrid approach: We propose a novel resource allocation method that combines the power of ACO and TS algorithms. This hybrid approach harnesses the strengths of both algorithms to address the limitations of conventional resource allocation methods.

- QoS prioritization: Our method places a strong emphasis on QoS, aiming to improve energy efficiency, reduce communication delays, and enhance overall system performance. This contributes to a more reliable and responsive IoT ecosystem.

- Efficiency and effectiveness: Through extensive testing and experimentation, we demonstrate the efficiency and effectiveness of our approach. Our results indicate significant improvements in QoS metrics when compared to traditional resource allocation methods.

The rest of the paper is organized in the following manner. Section II presents an introduction to IoT and resource allocation and describes existing resource allocation methods. Section III discussed the proposed method. Simulation results are reported in Section IV. Section V summarizes the main contributions of the study.

II. Background

A. IoT Resource Allocation

Resource allocation is of utmost importance in the IoT ecosystem, where a multitude of interconnected devices collaborate to provide diverse services and applications. In the IoT, resources such as network bandwidth, computational power, storage capacity, and energy are scarce and must be allocated efficiently among numerous devices and applications [18]. Effective resource allocation in the IoT involves determining the optimal assignment of resources to meet the diverse requirements of IoT applications while considering factors such as QoS, energy efficiency, and network stability. A major challenge of IoT resource allocation stems from the dynamic and heterogeneous nature of IoT environments. IoT devices possess varying capabilities, communication protocols, and QoS requirements, further complicating the allocation process.

Additionally, IoT networks encounter fluctuations in resource availability due to device mobility, changing network conditions, and varying demands from different applications. Resource allocation algorithms in the IoT must be adaptive, scalable, and capable of handling network dynamics. Furthermore, considering resource limitations in IoT deployments, efficient allocation strategies are essential to prevent bottlenecks, ensure optimal resource utilization, and enhance the overall performance of IoT systems [19].

To address the resource allocation challenges in the IoT, a range of approaches have been proposed, each tailored to specific IoT scenarios and requirements. These approaches span from centralized algorithms to distributed mechanisms. Centralized resource allocation methods involve a central entity or server that receives requests from IoT devices and allocates resources based on predefined criteria or optimization objectives [20]. These algorithms centralize the decision-making process and effectively manage resource allocation in certain IoT environments. Distributed resource allocation algorithms aim to distribute the decision-making process among IoT devices themselves, enabling them to collaborate and negotiate for resources autonomously [21]. These algorithms promote self-organization and adaptability in resource allocation, making them suitable for dynamic and decentralized IoT systems. Optimization techniques such as genetic algorithms, swarm intelligence, and game theory have been applied to solve the resource allocation problem in the IoT [22]. These techniques consider QoS, energy efficiency, load balancing, and fairness while allocating resources to IoT devices. They provide efficient and optimized resource allocation solutions based on mathematical models and optimization objectives. Machine learning and artificial intelligence-based approaches are gaining prominence in IoT resource allocation. These approaches leverage historical data and real-time analytics to make intelligent resource allocation decisions. By learning from past experiences and adapting to changing IoT conditions, machine learning-based algorithms can optimize resource allocation to meet dynamic IoT requirements. Edge computing offers a promising model for resource allocation in the IoT. By deploying computation and storage capabilities closer to IoT devices at the network edge, edge computing reduces latency, optimizes bandwidth usage, and enables localized resource allocation decisions. This decentralized approach minimizes the reliance on cloud resources and enhances the overall efficiency and responsiveness of IoT systems [23].

B. Related Work

Wang, et al. [24] focused on addressing the distributed resource allocation problem in energy-efficient data forwarding for resource-constrained Industrial IoT (IIoT) systems [9]. They approached this problem by formulating it as a Decentralized, Partially Observable Markov Decision Process (Dec-POMDP), taking into account the decentralized and partially observable nature of the system. To tackle this challenge, they proposed an innovative algorithm named Dual-Attention assisted Deep Reinforcement Learning (DADR) for energy-efficient resource allocation. The DADR algorithm leverages a dual-attention assisted deep reinforcement learning (DRL) model within the Convolutional Attention Module, Dual-Attention, and Experience Reconstruction (CTDE) framework. The actor-network of the DADR algorithm incorporates a multi-scale convolutional attention module (CAM) to extract feature information from local states across various dimensions. Introducing a novel critic network, which employs a dual-attention module and an experience reconstruction module, enables comprehensive and precise evaluation of the system state from a global perspective. This critic network effectively addresses non-stationary and partially observable issues in multi-agent systems while maintaining scalability in dynamic environments without requiring modifications to the model structure. By combining CAM and Multi-Head Self-Attention (MHSA), the DADR algorithm enhances the representation learning capability of the DRL model.

Consequently, it provides improved optimization directions for energy efficiency and data transmission reliability. To assess the performance of the DADR algorithm, the researchers conducted simulations. The results of these simulations demonstrate the superiority of DADR over existing resource allocation algorithms and Multi-Agent Reinforcement Learning (MARL) models in terms of network stability, transmission reliability, and network lifetime.

Kim and Ko [25] introduced a service resource allocation approach that aims to minimize data transmissions among users' mobile devices while effectively addressing the constraints associated with such environments. To address the resource allocation problem, they transform it into a variant of the degree-constrained minimum spanning tree problem. Subsequently, they apply a genetic algorithm to efficiently generate near-optimal solutions within a shorter timeframe. The authors devise a fitness function and an encoding scheme specifically tailored to optimize the application of the genetic algorithm. Through the utilization of these components, the proposed approach demonstrates a remarkably high success rate, achieving near-optimal solutions in an average of 97% of cases. Moreover, it surpasses the brute force approach by significantly reducing the time required for solution generation.

In the study conducted by Tsai [26], the challenges associated with resource allocation in IoT systems are addressed. These systems are characterized by diverse user requirements, different types of appliances, limited network bandwidth, and computation power, all of which pose limitations to the performance of IoT systems and necessitate effective resource allocation solutions. To tackle this problem, the author proposes an algorithm that combines the concepts of data clustering and metaheuristics. The algorithm focuses on allocating the large-scale devices and gateways within the IoT system in a manner that minimizes the total communication cost between them. By optimizing the resource allocation, the algorithm aims to enhance the overall performance of the IoT system. The proposed algorithm is evaluated through simulations, and the results demonstrate its superiority over other resource allocation algorithms considered in the study. Specifically, the algorithm outperforms alternative approaches in terms of reducing the total data communication costs, highlighting its effectiveness in optimizing resource allocation for IoT systems.

Deng, et al. [27] address the challenge of trustworthiness management in edge computing (EC) systems, which play a crucial role in handling the increasing number of IoT devices connected to the edge of the network. They focus on ensuring compliance with service-level agreements (SLAs), which serve as an important indicator of trustworthiness for IoT services. To tackle this challenge, the authors propose a solution that involves modeling the state of the service provisioning system and the resource allocation scheme as a Markov decision process (MDP). They encode the trustworthiness gain, measured by the degree of SLA compliance, and use it as the objective for resource allocation adjustments. To obtain an optimal resource allocation policy, the authors employ reinforcement learning (RL) techniques. They train a policy using RL methods, which enables the dynamic generation of resource allocation schemes based on the system's current state. The trained policy is designed to maximize the trustworthiness gain of the services by allocating resources appropriately. The proposed approach is evaluated through experiments conducted on the YouTube request dataset. The results demonstrate that the edge service provisioning system utilizing the proposed approach outperforms baseline approaches by at least 21.72% in terms of performance.

Nematollahi, et al. [28] have proposed a novel architecture for offloading jobs and allocating resources for the IoT by incorporating Fog Computing (FC). They aim to address the limitations of low processing power and the need for efficient data processing and management in IoT applications. The architecture consists of three main components: sensors, controllers, and FC servers. The authors introduce the concept of the subtask pool approach in the second layer, which enables the offloading of work from IoT devices to FC servers. To optimize resource allocation, they combine the Moth-Flame Optimization (MFO) algorithm with Opposition-based Learning (OBL), forming the OBLMFO algorithm. In the second layer, a stack cache approach is implemented to ensure resource allocation is balanced and prevent system load imbalance. The authors also leverage blockchain technology to guarantee the accuracy of transaction data, enhancing the reliability and transparency of resource distribution in the IoT system. To evaluate the performance of the OBLMFO model, the authors conducted experiments using the Python environment with a diverse set of jobs. The results demonstrate that the OBLMFO model achieved a 12.1% reduction in the delay factor and a 6.2% reduction in energy consumption compared to existing approaches.

Nguyen, et al. [29] propose a generalized federated learning (FL) algorithm to address the challenges encountered in FL, including non-independent and identically distributed data and heterogeneity of user equipment (UE). The objective of their approach is to reduce the global communication burden and enhance the convergence rate of FL. The proposed FL algorithm builds upon the current state-of-the-art federated averaging (FedAvg) by introducing a weight-based proximal term to the local loss function. This modification enables the algorithm to perform stochastic gradient descent in parallel on a sampled subset of UEs during each global round, effectively reducing the communication overhead. The researchers provide a convergence upper bound that illustrates the tradeoff between the convergence rate and the number of global rounds. The analysis shows that convergence can still be ensured even with a small number of active UEs per round.

Liu [30] proposed a resource allocation algorithm for mobile edge computing. The algorithm aims to optimize base station performance over the long term by considering various factors, such as cable channel congestion, energy consumption, latency, communication cost, and task arrival characteristics. An energy consumption deficit queue based on Lyapunov drift penalties is introduced. This queue couples the energy consumption and time of small base stations, ensuring that energy consumption constraints are met during optimization. To calculate the offloading weight for task allocation, the authors employ game theory and propose an offloading weight formula derived from the Shapley value. The offloading weight is computed impartially, factoring in the return of various tasks. Simulations on the MATLAB platform were used to evaluate the proposed algorithm's performance. The algorithm can attain Nash equilibrium within a finite number of iterations, according to the results. Furthermore, the algorithm outperforms other comparison strategies in terms of the number of successfully offloaded tasks, time delay, and energy consumption.

TABLE I. IOT RESOURCE ALLOCATION METHODS

Method	Approach	Key features	Performance
DADR	Deep reinforcement learning model based on partially observable Markov decision processes	Centralized training and distributed execution framework Actor-network with a multi-scale convolutional attention module. Novel critic network based on dual-attention module	Outperforms existing resource allocation algorithms and multi-agent reinforcement learning models
Genetic Algorithm	Transformation of the resource allocation problem into a variant of the degree-constrained minimum spanning tree problem	Fitness function and encoding scheme for optimization	Achieves near-optimal solutions in 97% of cases on average and outperforms brute force approach
Data clustering and metaheuristics	Algorithm leveraging data clustering and metaheuristics	Minimization of total communication cost. Optimization of IoT system performance	Outperforms other resource allocation algorithms and reduces total data communication costs
Markov decision process with reinforcement learning	Modeling of state of service provisioning system and resource allocation as MDP	Trustworthiness gain as objective. Dynamic generation of resource allocation schemes	Outperforms baseline approaches by at least 21.72%
Fog Computing with moth-flame optimization and opposition-based learning	Architecture incorporating fog computing for resource allocation optimization	Subtask pool approach. Stack cache approach. Blockchain technology for accuracy of transaction data	Achieves reduction in delay factor and energy consumption
Generalized federated learning	Weight-based proximal term in local loss function and parallel stochastic gradient descent on a sampled subset of user equipment	Convergence rate improvement. Reduction of global communication burden	Requires less training time and energy consumption compared to full user participation
Task offloading and resource allocation algorithm	Optimization of long-term performance of small base stations and consideration of various factors	Energy consumption deficit queue. Offloading weight model based on Shapley value	Achieves Nash equilibrium and outperforms other comparison strategies

Table I presents a summary of IoT resource allocation methods along with their key features and performance characteristics. The reviewed methods often exhibit limitations that make them less suitable for the resource allocation problem. Some optimization techniques, such as genetic algorithm, can suffer from slow convergence, especially in large-scale IoT systems. This can hinder real-time decision-making, which is essential in many IoT applications. PSO and traditional heuristic methods may produce suboptimal solutions when dealing with the complex, combinatorial nature of IoT resource allocation problems. Many existing methods do not inherently prioritize QoS metrics. They may not effectively address the specific QoS requirements of diverse IoT applications. We chose the proposed hybrid method, which combines ACO and TS algorithms, for several compelling reasons:

- Combinatorial optimization: IoT resource allocation is inherently a combinatorial optimization problem, as it involves allocating limited resources to tasks with varying requirements. ACO excels in solving such problems by mimicking the foraging behavior of ants and constructing solutions through probabilistic resource selection. This aligns with the nature of the resource allocation problem in IoT systems.

- Local search enhancement: While ACO provides global exploration, TS offers local search capabilities by maintaining a Tabu list. TS intensifies the search process by preventing revisits to previously explored solutions, which is crucial for avoiding suboptimal solutions in IoT resource allocation.

- QoS focus: Our primary objective is to prioritize QoS in IoT resource allocation. Existing methods may lack the necessary mechanisms to give due consideration to QoS metrics, such as energy efficiency and reduced communication delays. Our hybrid approach allows us to explicitly address these QoS concerns.

III. Proposed Method

This section introduces a novel resource allocation algorithm called ACO-TS, which combines the ACO and TS algorithms. ACO-TS offers improved efficiency in terms of cost and execution time compared to existing approaches. We delve into the details of ACO-TS in the following subsections: Firstly, we define the problem and outline the objective function. Next, we provide a comprehensive explanation of the proposed method, highlighting its key features and mechanisms.

A. Problem Definition

The distributed nature of resources and the need for efficient access make resource allocation in IoT challenging. The integration of multiple applications and the heterogeneity of connectivity further complicate resource allocation. Efficiency in an IoT system is measured by factors such as allocation interval, response time, and processing time, which are important QoS constraints. This study focuses on the resource allocation problem involving resource nodes and gateways. Resources are allocated to service instances, and gateways are responsible for connecting to these resources. Gateways serve as the interconnection points for IoT systems, managing the traffic of multiple resources. Effective resource allocation requires optimizing the distribution of resources among gateways to minimize communication costs. Additionally, the connectivity between gateways is crucial, and minimizing communication costs is a key objective. Various connection models, such as ring or bus connections, can be considered to achieve this goal. Fig. 1 illustrates an example of a resource-gateway connection. Communication costs depend on the chosen communication model. The objective of the problem is to determine the resource allocation pattern that minimizes communication costs. Another important aspect is load balancing, which involves distributing resources among gateways to prevent bottlenecks. The objective function section describes how load balancing and communication costs are calculated in the problem.

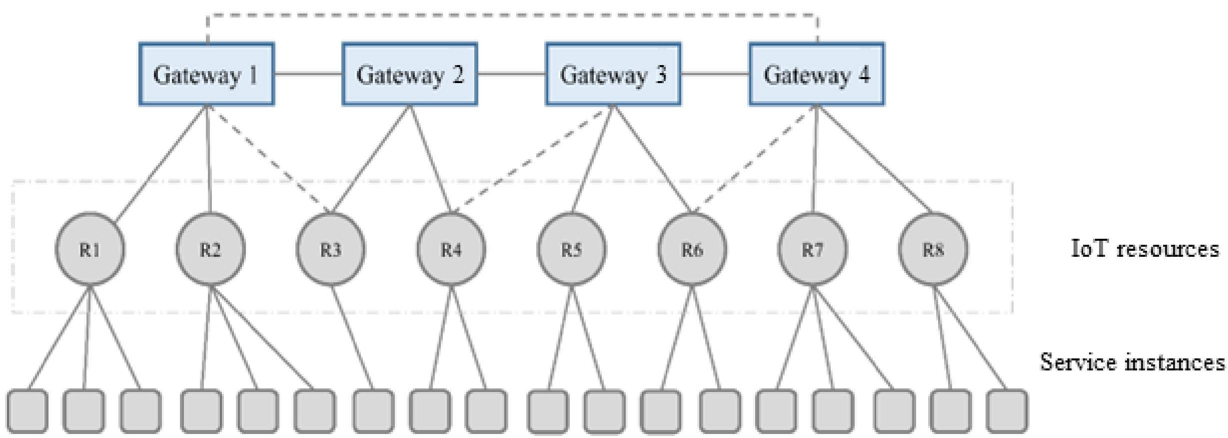

Fig. 1. An example of a resource-gateway connection.

B. Objective Function

In the considered network model, it is assumed that all resource nodes have the ability to communicate with each other. Therefore, to evaluate the different solutions to the resource allocation problem, it is necessary to calculate the total cost of network communications. It is assumed that all resources send messages to each other, and the objective function aims to minimize the total cost of these messages. The total cost is calculated using Eq. 1, which represents the mathematical expression for determining the cost. The proposed algorithm focuses on minimizing this objective function as its main goal.

$$T_c = \frac{\sum_{j=1}^{|V_g|}\left(d_j^r \times d^g\right)}{P} \tag{1}$$

In Eq. (1), the variable d^g represents the overall communication cost among gateways, d_j^r represents the total data transfer cost between the j^{th} gateway and all resources connected, and V_g represents the total number of gateways in the network. It is important to note that communication within a gateway can also result in messages being exchanged between gateways. The exponential nature of the communication cost within a gateway greatly affects the total communication cost. Thus, the equation $(d_j^r \times d^g)$ is the most appropriate way to calculate the maximum communication costs. The objective is to minimize the value of the objective function. Since gateways have the ability to send messages to all resources, we need to multiply d_j^r by d^g. By summing up these values for all gateways, we can obtain an estimation of the total communication costs for the gateways. The numerator part of the T_c fraction in the objective function can be calculated by performing this calculation for each gateway.

C. Proposed Hybrid Algorithm

The proposed hybrid method combines the ACO algorithm and the TS technique to tackle the IoT resource allocation problem, considering QoS requirements. This combination leverages the strengths of both algorithms to achieve efficient and effective resource allocation. The ACO algorithm, which mimics the foraging behavior of ants, simulates the behavior of ants in search of food [31]. The algorithm constructs solutions by probabilistically selecting resources based on pheromone trails and heuristics. The pheromone trail represents the quality of each resource allocation solution, and ants deposit and update pheromone values as they move through the solution space. This behavior encourages the exploration of promising solutions and exploits the accumulated knowledge of previous solutions. By utilizing the ACO algorithm, the proposed method can efficiently explore the large solution space and identify potential resource allocation configurations that satisfy the QoS requirements. To further enhance the search process and overcome local optima, the TS technique is integrated into the proposed method. TS is a local search-based metaheuristic that intensifies the search by maintaining a tabu list [32]. The tabu list keeps track of recently visited solutions, preventing them from being revisited in subsequent iterations. This mechanism encourages diversification and exploration of new regions in the solution space, allowing the algorithm to escape local optima and search for better resource allocation solutions. Neighborhood search and diversification strategies are also applied within the TS framework to further improve the search process.

The proposed hybrid method balances exploration and exploitation by fusing the ACO algorithm and TS technique. The ACO algorithm explores the solution space using pheromone trails and heuristics, while the TS technique intensifies the search process by leveraging the tabu list and neighborhood search strategies. This combination enables the algorithm to efficiently converge towards optimal or near-optimal solutions that satisfy the QoS requirements of IoT applications. The ACO algorithm updates the pheromone trails based on the quality of each resource allocation solution. The pheromone update formula is given by:

$$\tau_{ij} = (1-\rho)\tau_{ij}^{old} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{2}$$

where τ_{ij} represents the pheromone level on edge (i, j) in the resource allocation graph, ρ is the evaporation rate that controls the decay of pheromone over time, and $\Delta\tau_{ij}^k$ represents the pheromone increment for ant k on edge (i, j) in the solution construction phase. The ACO algorithm constructs solutions by probabilistically selecting resources based on the pheromone trails and heuristics. The probability of selecting resource j for ant k at node i is calculated using the following equation:

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta} & if\ j \in N_k \\ 0 & otherwise \end{cases} \tag{3}$$

Where α and β control the relative importance of pheromone trails and heuristics, η_{ij} represents the heuristic information, which captures the desirability of allocating resource j at node i based on factors such as energy consumption, network capacity, and QoS requirements. The TS algorithm maintains a tabu list to prevent revisiting previously visited solutions. The tabu list is updated using the following equation:

$$TL = TL_U(i, j) \tag{4}$$

Where TL represents the tabu list, (i, j) represents the resource allocation move (e.g., allocating resource j at node i) that is added to the tabu list. The pseudocode for the proposed hybrid method is shown in Algorithm 1. The proposed algorithm begins by initializing the necessary variables and data structures, including the resource allocation graph, QoS requirements, and algorithm parameters. It then enters a loop that repeats for a specified number of iterations. Within each iteration, a set of ants is created to construct resource allocation solutions. Each ant starts at a random node and selects resources probabilistically according to the pheromone trails and heuristics. The constructed solutions are evaluated against the QoS requirements, and pheromone trails are adjusted accordingly. After the ACO phase, the algorithm proceeds to the TS initialization. In this case, the best solution is set as the current solution, and the tabu list is initialized as empty. The TS loop begins, continuing until a stopping criterion is met. In each iteration of the TS loop, neighboring solutions are generated by making resource allocation moves from the current solution, taking into account the tabu list restrictions. The best non-tabu solution is selected, and if it improves upon the current best solution, it becomes the new best solution. The corresponding resource allocation move is added to the tabu list, and the selected solution becomes the current solution for the next iteration. The algorithm repeats this process for the specified number of iterations, continuously improving the resource allocation solutions. Finally, the best resource allocation solution found is output as the final result.

Algorithm 1. Pseudocode of the proposed algorithm

Initialize:
- Initialize pheromone levels τ_{ij} for all edges (i, j) in the resource allocation graph
- Initialize an empty tabu list TL

Repeat for a specified number of iterations:
  Create a set of ants:
  - For each ant k:
    - Start at a random node i
    - Initialize an empty resource allocation solution S_k

  Construct solutions:
  - For each ant k:
    - While not all nodes are visited:
      - Calculate the selection probability P_{ij}^k for unvisited neighboring nodes
      - Select a resource j based on the selection probability
      - Add resource j to the solution S_k
      - Update the visited and unvisited node lists

Update pheromone trails:
- For each ant k:
  - Calculate the solution quality metric Q_k based on QoS requirements
  - Update the pheromone trails:
    - For each edge (i, j) in the solution S_k:
      - Calculate the pheromone increment $\Delta\tau_{ij}^{k}$
      - Update pheromone level τ_{ij} using the pheromone update formula

Evaporate pheromone trails:
- For each edge (i, j) in the resource allocation graph:
  - Evaporate pheromone level τ_{ij} using the evaporation rate ρ

Tabu Search:
- Initialize the best solution S_{best} with the best solution found so far
- Set the current solution S_{curr} as S_{best}
- Set the tabu list *TL* as empty
- Repeat until a stopping criterion is met:
  - Generate a set of neighboring solutions by making resource allocation moves on S_{curr}
  - Select the best non-tabu solution S_{next} from the neighboring solutions
  - If S_{next} is better than S_{best}, update S_{best}
  - Add the resource allocation move corresponding to S_{next} to the tabu list *TL*
  - Set S_{curr} as S_{next}

The overall time complexity of the combined ACO and TS algorithms can be expressed as a combination of the complexities of both algorithms. Assume the number of ants as N_a, the number of iterations as N_i, the size of the solution space as N_s, the number of resources available for allocation as M, the size of the neighborhood as N_n, and the diversification parameter as N_d. The time complexity of the ACO algorithm can be approximated as $O(N_a * N_i * f(N_s))$, where $f(N_s)$ represents the complexity of constructing solutions. The ACO algorithm updates the pheromone trails based on the quality of each resource allocation solution, which has a complexity of $O(M)$ since the pheromone values for each resource need to be updated. The time complexity of the TS algorithm primarily depends on the neighborhood search operation and the diversification strategies employed. Assuming the complexity of the neighborhood search operation is $O(g(N_n))$ and the complexity of diversification strategies is $O(h(N_d))$, the overall time complexity of the TS algorithm can be approximated as $O(N_i * (g(N_n) + h(N_d))$. Consequently, the time complexity of the combined ACO and TS algorithm can be represented as:

$$O\Big(N_a \times \big(N_i \times O\big(f(N_s)\big) + M\big) + O\big(g(N_n)\big) + O\big(h(N_d)\big)\Big) \quad (5)$$

IV. Results and Discussion

In this section, we present the simulation and evaluation of the suggested algorithm. The simulations were performed using MATLAB software on a desktop computer with a core i5 CPU and 4GB of RAM. The performance of the proposed resource allocation technique is assessed by comparing it with previous methods in two scenarios. The first scenario involves four experiments with varying numbers of gateway and source nodes. The details of these experiments are summarized in Table II. Each experiment is characterized by the number of gateway and resource nodes. These parameters allow for the evaluation and comparison of the proposed method under different network configurations. The overall fitness of each mechanism is depicted in Fig. 2 to 5. From these figures, it can be observed that the proposed method consistently outperforms genetic, PSO, and ACO algorithms. Scalability is another important aspect considered in the proposed technique. Fig. 3 proves that as the network size increases, the convergence diagram of the proposed technique (blue line) diverges from the convergence diagrams of the ACO and genetic algorithms (red and black lines), indicating its scalability. The optimality of the convergence graph is determined by the communication costs in the network. The fitness function used in the evaluation takes into account execution time and energy consumption, which are crucial factors in determining the sub-function of fitness.

TABLE II. SIMULATION PARAMETER VALUES

Parameters	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Number of gateway nodes	10	25	60	120
Number of resource nodes	30	100	500	800

We compare the proposed method with ACO and genetic algorithms in the second scenario. Similar to the first scenario, four experiments are conducted, following the setup described in Table II. Fig. 6 to 9 illustrate the performance of the algorithms in these experiments. From the figures, it can be observed that the genetic algorithm exhibits the highest execution time among the three algorithms. This prolonged decision-making process can lead to traffic congestion on the server, causing delays and inefficiencies. On the other hand, the ACO algorithm performs better than the genetic algorithm, as it takes less time to select the best resource. In comparison to existing benchmark algorithms, our proposed method demonstrates a faster decision-making process in selecting the best resource. This reduced time is beneficial in achieving efficient resource allocation and improving overall system performance.

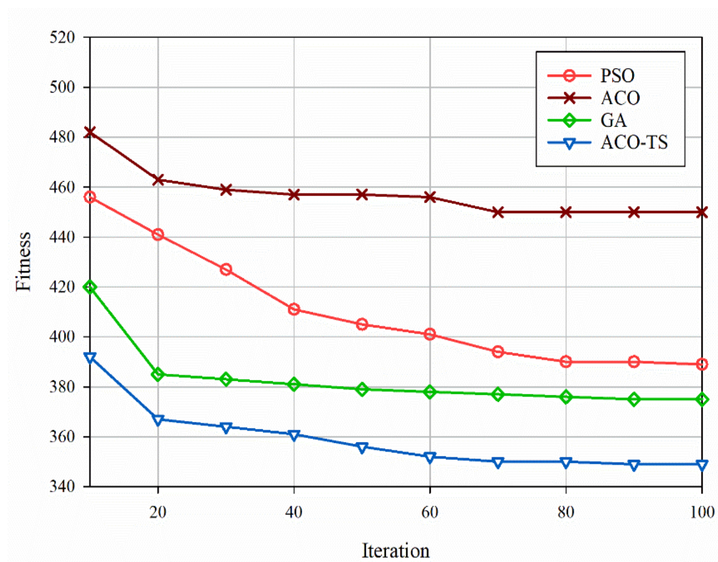

Fig. 2. Fitness comparison (First experiment).

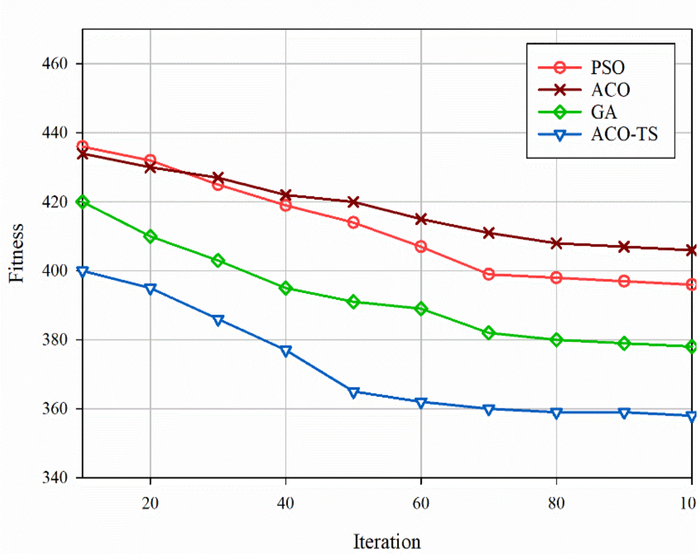

Fig. 3. Fitness comparison (Second experiment).

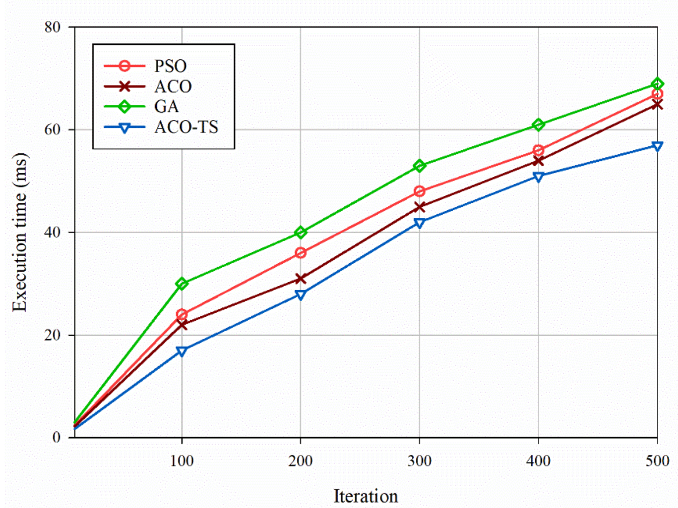

Fig. 6. Execution time comparison (First experiment).

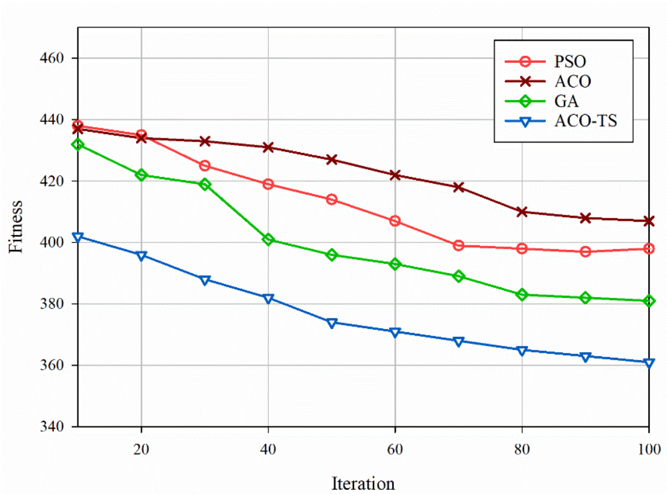

Fig. 4. Fitness comparison (Third experiment).

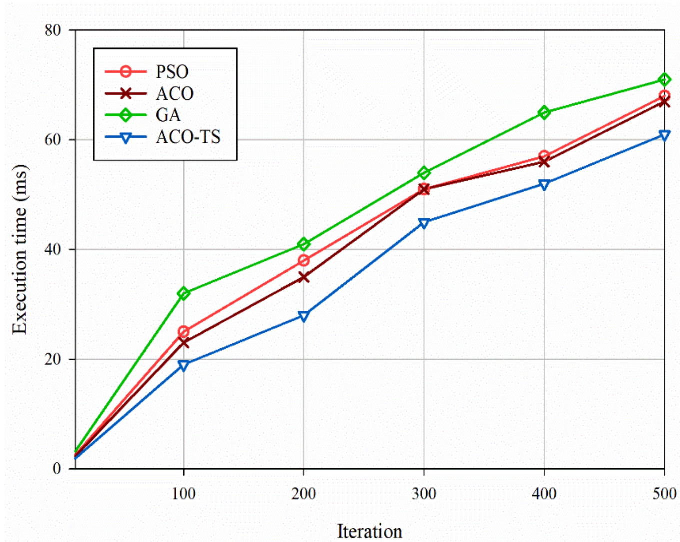

Fig. 7. Execution time comparison (Second experiment).

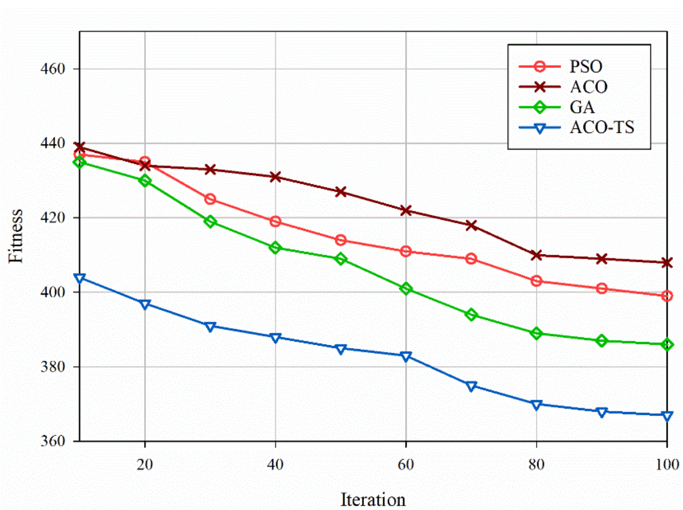

Fig. 5. Fitness comparison (Fourth experiment)

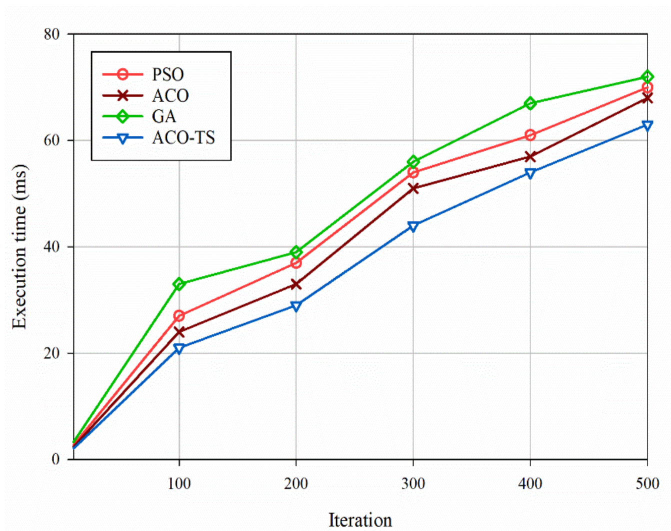

Fig. 8. Execution time comparison (Third experiment)

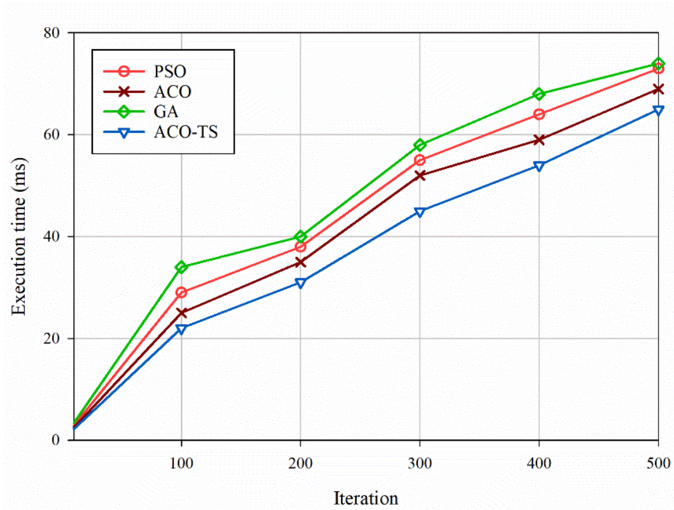

Fig. 9. Execution time comparison (Fourth experiment)

In our evaluation of the proposed algorithms, we utilized multiple datasets that reflect diverse IoT scenarios and characteristics. These datasets differ in terms of size, complexity, and the specific application domains they represent. Such variations inherently contribute to differences in algorithm performance. For instance, in datasets characterized by large-scale IoT networks with numerous interconnected devices, the proposed hybrid algorithm may excel in optimizing resource allocation by leveraging ACO's global exploration capabilities. On the other hand, in smaller-scale networks or datasets representing resource-intensive IoT applications, the local search enhancements offered by the TS algorithm might yield more pronounced benefits. Moreover, the nuances of each dataset, including the nature of IoT devices, their energy constraints, and communication patterns, can significantly impact the suitability of the proposed algorithms. The variations observed in comparative results across these datasets suggest that our algorithms exhibit adaptability to different IoT contexts. These findings open avenues for future research in fine-tuning the proposed algorithms to specific IoT scenarios, as well as exploring adaptive resource allocation strategies that can dynamically adjust to the unique requirements of diverse IoT data types and application domains. In essence, the dataset variations shed light on the algorithmic flexibility of our approach, indicating its potential to cater to the evolving and multifaceted landscape of IoT resource allocation.

V. Conclusion

This paper suggested a novel QoS-aware resource allocation method for IoT systems using a hybrid approach of the ACO and TS algorithms, called ACO-TS. It efficiently allocates limited resources in IoT systems in accordance with the QoS requirements of individual devices. Through the integration of the ACO algorithm and TS technique, we have demonstrated the effectiveness and efficiency of our approach in optimizing resource allocation decisions. The ACO algorithm leverages the behavior of ant colonies to explore the solution space, while the TS technique intensifies the search process to overcome local optima. By combining these two techniques, we achieve a balance between exploration and exploitation, resulting in improved convergence speed and solution quality. Our experimental evaluations in realistic IoT scenarios have suggested the merits of the ACO-TS approach. In comparison with existing resource allocation methods, ACO-TS achieves significant improvements in energy consumption reduction, network capacity maximization, and satisfaction of QoS requirements.

References

[1] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research," Journal of Network and Computer Applications, vol. 97, pp. 23-34, 2017.

[2] F. Kamalov, B. Pourghebleh, M. Gheisari, Y. Liu, and S. Moussa, "Internet of Medical Things Privacy and Security: Challenges, Solutions, and Future Trends from a New Perspective," Sustainability, vol. 15, no. 4, p. 3317, 2023.

[3] T. Taami, S. Azizi, and R. Yarinezhad, "Unequal sized cells based on cross shapes for data collection in green Internet of Things (IoT) networks," Wireless Networks, pp. 1-18, 2023.

[4] B. Pourghebleh and V. Hayyolalam, "A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things," Cluster Computing, pp. 1-21, 2019.

[5] A. Thakur and M. S. Goraya, "RAFL: A hybrid metaheuristic based resource allocation framework for load balancing in cloud computing environment," Simulation Modelling Practice and Theory, vol. 116, p. 102485, 2022.

[6] P. He, N. Almasifar, A. Mehbodniya, D. Javaheri, and J. L. Webber, "Towards green smart cities using Internet of Things and optimization algorithms: A systematic and bibliometric review," Sustainable Computing: Informatics and Systems, vol. 36, p. 100822, 2022, doi: https://doi.org/10.1016/j.suscom.2022.100822.

[7] V. Srinadh and P. N. Rao, "Implementation of Dynamic Resource Allocation using Adaptive Fuzzy Multi-Objective Genetic Algorithm for IoT based Cloud System," in 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2022: IEEE, pp. 111-118.

[8] A. K. Sangaiah, A. A. R. Hosseinabadi, M. B. Shareh, S. Y. Bozorgi Rad, A. Zolfagharian, and N. Chilamkurti, "IoT resource allocation and optimization based on heuristic algorithm," Sensors, vol. 20, no. 2, p. 539, 2020.

[9] M. Mohseni, F. Amirghafouri, and B. Pourghebleh, "CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic," Peer-to-Peer Networking and Applications, pp. 1-21, 2022.

[10] S. Li, Q. Ni, Y. Sun, G. Min, and S. Al-Rubaye, "Energy-efficient resource allocation for industrial cyber-physical IoT systems in 5G era," IEEE Transactions on Industrial Informatics, vol. 14, no. 6, pp. 2618-2628, 2018.

[11] R. Singh et al., "Analysis of Network Slicing for Management of 5G Networks Using Machine Learning Techniques," Wireless Communications and Mobile Computing, vol. 2022, 2022.

[12] B. M. Jafari, M. Zhao, and A. Jafari, "Rumi: An Intelligent Agent Enhancing Learning Management Systems Using Machine Learning Techniques," Journal of Software Engineering and Applications, vol. 15, no. 9, pp. 325-343, 2022.

[13] C. Han and X. Fu, "Challenge and Opportunity: Deep Learning-Based Stock Price Prediction by Using Bi-Directional LSTM Model," Frontiers in Business, Economics and Management, vol. 8, no. 2, pp. 51-54, 2023.

[14] M. Bagheri et al., "Data conditioning and forecasting methodology using machine learning on production data for a well pad," in Offshore Technology Conference, 2020: OTC, p. D031S037R002.

[15] S. N. H. Bukhari, J. Webber, and A. Mehbodniya, "Decision tree based ensemble machine learning model for the prediction of Zika virus T-cell epitopes as potential vaccine candidates," Scientific Reports, vol. 12, no. 1, p. 7810, 2022.

[16] H. Kosarirad, M. Ghasempour Nejati, A. Saffari, M. Khishe, and M. Mohammadi, "Feature Selection and Training Multilayer Perceptron Neural Networks Using Grasshopper Optimization Algorithm for Design Optimal Classifier of Big Data Sonar," Journal of Sensors, vol. 2022, 2022.

[17] S. Saeidi, S. Enjedani, E. Alvandi Behineh, K. Tehranian, and S. Jazayerifar, "Factors Affecting Public Transportation Use during Pandemic: An Integrated Approach of Technology Acceptance Model and Theory of Planned Behavior," Tehnički glasnik, vol. 18, pp. 1-12, 09/01 2023, doi: 10.31803/tg-20230601145322.

[18] P. Zuo, G. Sun, Z. Li, C. Guo, S. Li, and Z. Wei, "Towards Secure Transmission in Fog Internet of Things Using Intelligent Resource Allocation," IEEE Sensors Journal, 2023.

[19] B. Pourghebleh, V. Hayyolalam, and A. A. Anvigh, "Service discovery in the Internet of Things: review of current trends and research challenges," Wireless Networks, vol. 26, no. 7, pp. 5371-5391, 2020.

[20] M. A. Raza, M. Abolhasan, J. Lipman, N. Shariati, W. Ni, and A. Jamalipour, "Statistical Learning-based Adaptive Network Access for the Industrial Internet-of-Things," IEEE Internet of Things Journal, 2023.

[21] X. Liu and X. Zhang, "NOMA-based resource allocation for cluster-based cognitive industrial internet of things," IEEE transactions on industrial informatics, vol. 16, no. 8, pp. 5379-5388, 2019.

[22] B. Pourghebleh, K. Wakil, and N. J. Navimipour, "A comprehensive study on the trust management techniques in the Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 6, pp. 9326-9337, 2019.

[23] B. Pourghebleh, N. Hekmati, Z. Davoudnia, and M. Sadeghi, "A roadmap towards energy‐efficient data fusion methods in the Internet of Things," Concurrency and Computation: Practice and Experience, p. e6959, 2022.

[24] Y. Wang, F. Shang, J. Lei, X. Zhu, H. Qin, and J. Wen, "Dual-attention assisted deep reinforcement learning algorithm for energy-efficient resource allocation in Industrial Internet of Things," Future Generation Computer Systems, vol. 142, pp. 150-164, 2023.

[25] M. Kim and I.-Y. Ko, "An efficient resource allocation approach based on a genetic algorithm for composite services in IoT environments," in 2015 IEEE international conference on web services, 2015: IEEE, pp. 543-550.

[26] C.-W. Tsai, "SEIRA: An effective algorithm for IoT resource allocation problem," Computer Communications, vol. 119, pp. 156-166, 2018.

[27] S. Deng et al., "Dynamical resource allocation in edge for trustable Internet-of-Things systems: A reinforcement learning method," IEEE Transactions on Industrial Informatics, vol. 16, no. 9, pp. 6103-6113, 2020.

[28] M. Nematollahi, A. Ghaffari, and A. Mirzaei, "Task and resource allocation in the internet of things based on an improved version of the moth-flame optimization algorithm," Cluster Computing, pp. 1-23, 2023.

[29] V.-D. Nguyen, S. K. Sharma, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Efficient federated learning algorithm for resource allocation in wireless IoT networks," IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3394-3409, 2020.

[30] J. Liu, "Task offloading and resource allocation algorithm based on mobile edge computing in Internet of Things environment," The Journal of Engineering, vol. 2021, no. 9, pp. 500-509, 2021.

[31] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," IEEE computational intelligence magazine, vol. 1, no. 4, pp. 28-39, 2006.

[32] W. Fan and R. B. Machemehl, "Tabu search strategies for the public transportation network optimizations with variable transit demand," Computer‐Aided Civil and Infrastructure Engineering, vol. 23, no. 7, pp. 502-520, 2008.