Whale Optimization Algorithm for Energy-Efficient Task Allocation in the Internet of Things

Shan YANG[1]*, Renping YU[2], Xin JIN[3]
State Grid Hubei Electric Power Co. Ltd, WuHan 430077, China[1]
Hubei Huazhong Electric Power Technology Development Co., Ltd, WuHan 430077, China[2, 3]

***Abstract*—The Internet of Things (IoT) represents a new paradigm where various physical devices interact and collaborate to achieve common goals. This technology encompasses sensors, mobile phones, actuators, and other smart devices that work together to perform tasks and applications. In order to ensure optimal performance of these tasks and applications, task allocation becomes a critical aspect in IoT networks. Task allocation in IoT networks is a complex problem due to the intricate connections and interactions among devices. The task allocation issue is generally recognized as NP-hard, necessitating the development of effective optimization solutions. This paper proposes a solution using the Whale Optimization Algorithm (WOA) to address the task allocation problem in IoT networks. By leveraging the capabilities of the WOA, our algorithm aims to improve energy efficiency and enhance network stability. The performance of the algorithm was tested comprehensively on the MATLAB simulation platform. Our algorithm outperforms existing algorithms in the literature, especially in terms of energy efficiency, as per the findings.**

Keywords*—Task allocation; internet of things; energy efficiency; optimization; whale optimization algorithm

I. INTRODUCTION

With the rise of the Internet of Things (IoT), a significant technological revolution has occurred over the past few years. The IoT envisions a network where various devices, including actuators, sensors, RFID tags, smartphones, and servers, with different capabilities can interconnect [1, 2]. The objective is to empower these network entities to collaborate efficiently and share their resources to achieve specific goals, such as executing assigned applications [3, 4]. However, the resources available in IoT networks, including electrical energy, memory, processing power, and node capabilities, are often limited. For instance, wireless sensor nodes are typically powered by batteries, resulting in constrained energy levels [5].

Similarly, RFID tags have limited processing capabilities. Considering the size of IoT networks and their heterogeneous nature, resource allocation becomes a non-trivial task to improve network performance [6]. Moreover, IoT networks exhibit dynamic behavior due to widespread deployment in smart settings like smart homes and cities [7]. These applications often rely on opportunistic sensing and networks. Opportunistic networks are characterized by the dynamic creation of connections among nodes without relying on a fixed infrastructure [8]. When forwarding a message, nodes opportunistically choose the next hop based on the probability of bringing the message closer to the recipient. Opportunistic sensing involves deploying sensors to sporadically capture data from the environment whenever their state aligns with the requirements of the application [9]. As dynamic scenarios are characterized by continuous and rapid adjustments, it is not feasible to allocate resources in a centralized manner [10]. Hence, in this context, a decentralized approach to resource allocation is necessary to deal with the dynamic nature of IoT networks [11]. Such an approach should enable nodes to autonomously manage and allocate resources based on local observations and interactions with neighboring nodes. Decentralized resource allocation methods can provide flexibility and adaptability to changing network conditions, making them suitable for IoT networks operating in dynamic and opportunistic environments [12].

IoT task allocation is a critical challenge since tasks must be assigned and fulfilled within specific deadlines. IoT environments present many challenges due to their unique properties and complexities. Task allocation belongs to the class of NP-hard problems, which makes it difficult to find optimal solutions using traditional approaches. Consequently, meta-heuristic algorithms, such as evolutionary algorithms, have emerged as effective tools for addressing this problem. Improving customer service quality is a significant concern in the IoT, and task allocation plays a crucial role in achieving this objective. Tasks represent the requirements of customers, and making appropriate decisions on how to allocate and execute these tasks can greatly enhance network performance and response quality. Various evolutionary algorithm-based strategies have been proposed to solve the task allocation problem in the IoT. These strategies aim to optimize task execution time and improve overall system efficiency compared to traditional methods.

Meta-heuristic algorithms and machine learning offer valuable solutions to address the inherent complexity of task allocation in IoT networks. Meta-heuristic algorithms, such as genetic algorithms, simulated annealing, imperialists competitive algorithm, and particle swarm optimization, provide efficient optimization strategies to tackle the NP-hard nature of task allocation problems [13, 14]. These algorithms iteratively explore the solution space, iteratively refining task assignments based on predefined objectives like energy efficiency, latency, or resource usage. Their adaptability and ability to find near-optimal solutions make them well-suited for the dynamic and diverse IoT environment. Machine learning, on the other hand, leverages data-driven models and predictive analytics to enhance task allocation decisions. These algorithms can analyze historical task allocation patterns, device capabilities, network conditions, and application requirements to predict the most efficient task-device assignments [15, 16]. By learning and adapting from data, machine learning enables IoT systems to dynamically allocate tasks in real-time, optimizing resource utilization and response times. Furthermore, reinforcement learning approaches can allow devices to autonomously learn and improve their task allocation strategies over time based on feedback from the environment [17].

This paper proposes a novel solution using the Whale Optimization Algorithm (WOA) to address the challenging task and resource allocation problems in IoT networks. By leveraging the capabilities of the WOA, our algorithm aims to improve energy efficiency, enhance network stability, and optimize resource allocation. The performance of the algorithm is tested comprehensively on the MATLAB simulation platform, and our results demonstrate its superiority over existing algorithms in the literature, particularly in terms of energy efficiency. Our approach also involves considering two criteria, the optimal path length and the compatibility between the features of nodes and the task requirements, to determine the best nodes to execute tasks, as well as utilizing the WOA to evaluate and determine the optimal path nodes based on the defined criteria and assigning task allocation abilities to the nodes along the determined optimal path, ensuring efficient and effective task execution. In this paper, we highlight the following objectives:

- Considering two criteria, namely the optimal path length and the compatibility between the features of nodes and the task requirements, to determine the best nodes to execute tasks;
- Utilizing the WOA to evaluate and determine the optimal path nodes based on the defined criteria;
- Assigning task allocation abilities to the nodes along the determined optimal path, ensuring efficient and effective task execution.

The paper is structured as follows: Section II discusses related work in the field, Section III outlines our proposed approach in detail, Section IV presents the results of simulations conducted on the MATLAB platform, and finally, Section V summarizes the key findings and contributions.

II. Related Work

Multi-cloud computing, which combines multiple cloud providers, has emerged as a high-performance and secure platform for data processing, offering enhanced security measures against potential threats. In this context, Cai, et al. [18] developed a distributed scheduling model that integrates task scheduling and multi-cloud in IoT. The model encompasses multiple criteria, including load balancing, resource utilization, energy consumption, throughput, and cost. To address this complex scheduling problem, a multi-objective adaptive strategy based on the sine function was proposed. The algorithm leverages the sine function's ability to capture variation tendencies in the population's diversity strategy. The outcomes from experiments prove the effectiveness of the proposed approach in achieving efficient scheduling and enhancing security measures. This work presents a novel solution for tackling the challenging task of data processing in the IoT environment.

Due to limitations in network infrastructure, efficiently transmitting the sheer volume of sensory data generated by IoT to distant data centers poses challenges in the rapidly growing era of big data. The emergence of fog computing, which uses local mini data centers near sensors, is a promising solution for real-time applications. This approach eases the main data center's burden, and cloud-based IoT is fully utilized. In this context, Boveiri, et al. [19] suggested a high-performance mechanism based on the Max-Min Ant System (MMAS), a variant of the ant colony optimization algorithm. The focus of their approach lies in the scheduling of static task graphs in homogeneous multiprocessor environments, which are commonly incorporated in fog computing systems. The key objective of the developed strategy is to effectively adjust task priority values to achieve optimal task order. By incorporating heuristic values derived from previous experience, the proposed approach demonstrates robustness and efficiency. Experimental evaluations using statistical task graphs of varying dimensions validate the effectiveness and reliability of the proposed approach compared to its traditional competitors, highlighting its performance advantages in fog computing environments.

Javanmardi, et al. [20] propose a fog task scheduler called FPFTS that combines PSO and fuzzy theory. The scheduler takes into account network utilization and application loop delay. To assess the performance of FPFTS, simulations are conducted using an IoT-based scenario in iFogSim. The experiments involve varying user movement, data rate, and response time. The findings demonstrate that FPFTS outperforms the delay-priority and first-come, first-served approaches. Specifically, FPFTS achieves an average reduction of 85.7% in delay-tolerant application loop delay compared to first-come, first-served. For delay-sensitive application loop delay, FPFTS achieves an average reduction of 87.1%. Furthermore, FPFTS achieves an average increase of 80.3% in network utilization. These findings highlight the effectiveness of FPFTS in optimizing performance metrics for fog-based IoT applications.

Bu [21] combined PSO and Genetic Algorithms for load balancing in IoT. The algorithm aims to achieve resource balance and alleviate server overload by considering various occupancy rates such as CPU, memory, and network bandwidth. These occupancy rates are used to develop the resource balance model, and a fitness function is used to estimate the influence and adjust the weights accordingly. The PSO algorithm, incorporating a disturbance factor and contraction operator, optimizes fitness functions to arrive at an optimal weight solution. The proposed hybrid algorithm is then implemented, validated, and evaluated against other approaches. The test results indicate that the proposed algorithm outperforms other ones in terms of resource utilization, request error rate, throughput, and response time.

IoT networks frequently experience node failures due to various factors such as signal disruptions, battery drain, or hardware malfunctions. In extensive IoT networks with complex tasks, the prospect of node failures amplifies in particular regions of the network. Weikert, et al. [22] have conducted a study on node failures and proposed a new task allocation algorithm based on multi-objective optimization to address this challenge. Their algorithm, labeled as the multi-objective task allocation algorithm (MOTA), involves a specialized archive-selection mechanism to increase diversity in the search space. The reliable selection of alternative task assignments in the IoT network is facilitated by maintaining a diverse archive in case of node failures. To examine the effectiveness of their approach, they use a network simulation model and compare the findings with the baseline MOTA and the dynamic task allocation scheduler (DTAS). The performance metrics considered include network lifetime, latency, and availability. The findings of their research indicate that the suggested method leads to substantial enhancements in performance as compared to the current algorithms, especially in situations with a high ratio of tasks to nodes. This highlights the effectiveness of their approach in mitigating the impact of node failures and enhancing the overall performance of the IoT network.

Bali, et al. [23] propose a meta-heuristic algorithm called the Hybrid Artificial Bee Colony Whales Optimization (HAWO) algorithm to optimize energy consumption and time delay in task allocation for edge-based processing devices in the IoT. The HAWO algorithm is formed by combining the ABC algorithm with WOA. This integration overcomes the premature convergence issue observed in the Artificial Bee Colony algorithm, which arises from the local search in the Employee Bee phase and Onlooker Bee phase, leading to looping problems. Through simulation experiments conducted in MATLAB, the researchers observed promising results with the integrated HAWO method. Additionally, compared to benchmark works, the HAWO algorithm exhibits significant enhancements of 60% in the optimum cost metric, 50% in energy consumption, and 25% in time delay.

The rapid advancement of IoT technology has led to the development of various applications and services, including cognitive computing systems and AI-powered chatbots. Nevertheless, due to the restricted computational capacity of IoT devices, cloud services are frequently required to manage data-intensive tasks. This approach, while offloading the processing burden, results in high latency, traffic congestion, and increased energy use. To address these challenges, Fog Computing (FC) is proposed as a solution to enable faster and more efficient data processing in time-sensitive IoT applications. Nematollahi, et al. [24] present a novel architecture for offloading jobs and allocating resources in the IoT environment. This architecture consists of sensors, controllers, and Fog Computing (FC) servers as part of an enhanced system. The later layer of the architecture operates a subtask pool approach for job offloading and employs a combination of the Opposition-based Learning (OBL) and Moth-Flame Optimization (MFO) algorithm for resource allocation, called OBLMFO. To avoid system load imbalance, a stack cache method is utilized for resource allocation in the subsequent layer.

Additionally, the architecture employs blockchain technology to ensure the precision of transaction data, enhancing resource allocation in the IoT. The evaluation of the OBLMFO model was conducted using the Python 3.9 environment, which involved a variety of different jobs. The outcomes indicate that the OBLMFO model accomplished a 12.18% drop in the delay factor and a 6.22% decline in energy usage. These findings highlight the effectiveness of the proposed architecture and OBLMFO model in enhancing the performance of resource allocation in the IoT, leading to reduced delays and improved energy efficiency.

Aghapour, et al. [25] present a method that utilizes deep reinforcement learning to address the challenges of offloading and resource allocation in the IoT. The proposed algorithm divides the problem into two sub-problems: offloading policy optimization and resource allocation optimization. The strategy for sending tasks to external devices is updated based on information obtained from the environment, and the resource allocation is optimized using the Salp Swarm Algorithm (SSA). By combining deep reinforcement learning with SSA, the proposed method aims to achieve efficient offloading and resource allocation in IoT infrastructures. The performance of the proposed algorithm is evaluated through simulations involving different deep-learning tasks executed on IoT devices with varying capacities of cloudlet devices. The results demonstrate that the proposed algorithm achieves the lowest cost in terms of energy consumption and latency. On average, it outperforms other methods, such as complete domestic processing, complete decentralization, and Jointly Resource allocation and computation Offloading PSO (JROPSO), by 92%, 17%, and 12%, respectively.

Our proposed method, based on the WOA, stands out for several reasons. In contrast to previous approaches that often struggled to optimize energy usage in IoT networks, our WOA-based method explicitly focuses on enhancing energy efficiency. Through rigorous simulations, we demonstrate that our approach significantly reduces energy consumption, making it well-suited to resource-constrained IoT devices. IoT networks are inherently dynamic and face network stability challenges. Our approach leverages the decentralized nature of the WOA to adapt dynamically to changing network conditions. This adaptability results in improved network stability, even in opportunistic and dynamic environments. Previous approaches often lacked the capability to efficiently allocate resources. Our method considers both optimal path length and compatibility between node features and task requirements, ensuring that resources are allocated to nodes best suited to execute tasks efficiently.

III. Proposed Approach

This section introduces an algorithm aimed at addressing the task allocation problem in IoT networks. The distribution of tasks is determined by two key parameters: the ability of a particular node to perform subtasks and the distance required to execute those subtasks. The section is divided into three subsections: the problem statement, the network model, and the explanation of the proposed solution.

A. Problem Definition

The task allocation problem is formulated using a complete graph $G = (V, E)$, where V refers to the nodes representing tasks, and E stands for the edges indicating the routes between the tasks. The user application is divided into multiple tasks that have specific sequential dependencies, and these dependencies are represented using a Directed Acyclic Graph (DAG). A DAG is a directed graph consisting of a collection of nodes and a collection of edges such that all the edges are in the form of $i \to j$, where i and j stand for nodes. Each node in the DAG represents a task, and an edge from node i to node j represents the dependency between task i and task j. Task j cannot be executed until task i has been completed, and task j becomes a ready node once all its parent tasks have been completed. In the context of the DAG, node i is considered a parent node of node j, and node j is a child of node i. To model the costs associated with computation and communication, weights are assigned to each node and edge in the DAG. The weight assigned to a node represents the cost of computation for that task, while the weight assigned to an edge represents the cost of communication between the corresponding tasks. Fig. 1 provides an example of a DAG, illustrating the task dependencies and associated weights. This example demonstrates how the DAG representation captures the sequential relationships between tasks and the corresponding computation and communication costs.

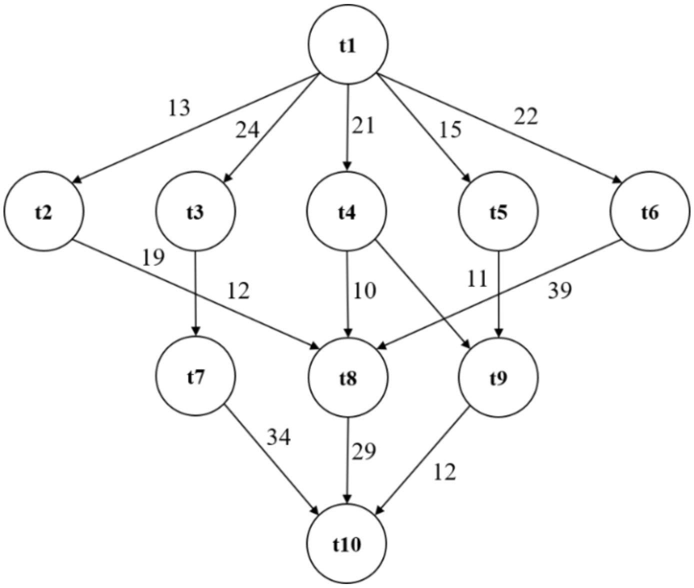

Fig. 1. A streamlined DAG for allocating IoT tasks.

B. Network Model

The IoT architecture adopted in this study consists of four layers: perception, network, service management, and application, illustrated in Fig. 2. RFID tags and barcodes make up the perception layer. This layer is responsible for collecting and processing real-time information using sensors and smart devices. These devices play a crucial role in capturing data from the environment. The network layer utilizes various technologies to provide networking capabilities. It enables connectivity and communication between sensors, smart devices, and other components in the IoT ecosystem. This layer handles the high volume of data generated by sensors and smart devices, ensuring efficient data transmission and management. The service management layer focuses on analyzing information, handling security, modeling processes, and managing devices. It plays a critical role in handling the collected sensor data, analyzing it for insights, and ensuring the security and integrity of the data. Task allocation, which involves assigning tasks to appropriate resources, is also a responsibility of this layer. Finally, the application layer encompasses programs and applications classified into network type, convergence, business models, and real-time requirements. This layer provides the interface for users to interact with the IoT system and offers various services and functionalities tailored to specific application domains.

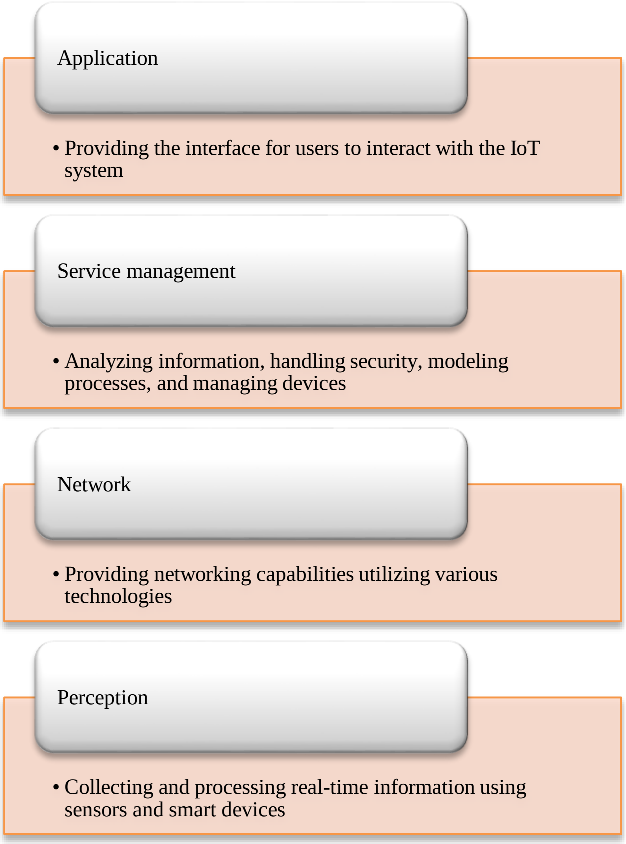
Application
• Providing the interface for users to interact with the IoT system
Service management
• Analyzing information, handling security, modeling processes, and managing devices
Network
• Providing networking capabilities utilizing various technologies
Perception
• Collecting and processing real-time information using sensors and smart devices

Fig. 2. IoT architecture.

C. WOA for IoT Task Allocation

The WOA draws inspiration from the hunting behavior of humpback whales searching for food in a multidimensional space. The locations of the whales serve as decision variables, and the objective costs are represented by the distances between the whales and their food source. The algorithm consists of three main operational processes: searching, encircling, and attacking the prey. These processes determine the time-dependent location of each whale [26]. The primary presentation of the WOA is depicted in Fig. 3. Each of the operational processes is described and mathematically expressed to guide the optimization process. The search for prey involves updating the position of each whale based on its current position and velocity. The encircling method simulates the collaboration between whales to encircle the prey and optimize the search process. The shrinking encircling prey process further refines the search space by adjusting the positions and velocities of the whales.

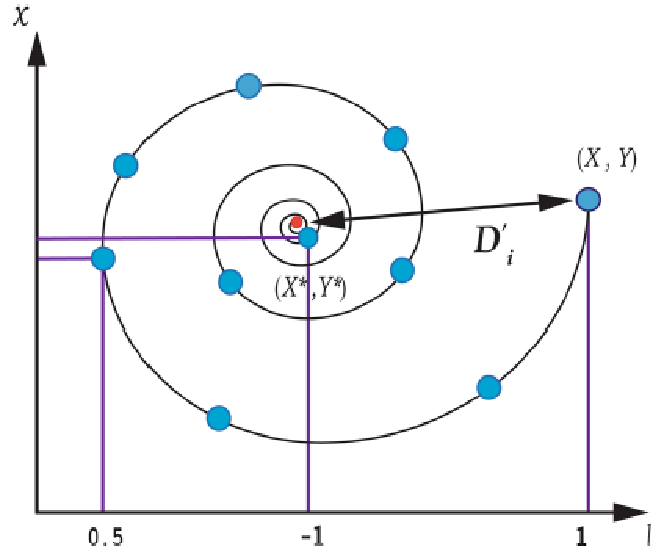

Fig. 3. Position update in a spiral.

The encircling behavior of whales during optimization is mathematically modeled using Eq. (1) and Eq. (2). These equations describe how whales update their positions to imitate the encircling behavior found around the best search agent at the moment.

$$\vec{D} = \left|\vec{C}.\overrightarrow{X^*}(t) - \vec{X}(t)\right| \quad (1)$$

$$\vec{X}(G+1) = \overrightarrow{X^*}(t) - \vec{A}.\vec{D} \quad (2)$$

X stands for the position vector, C and A refers to the coefficient vectors, and t is the current iteration. $X*$ denotes the position vector of the most optimal solution found thus far. $X*$ will be updated if there is a better solution for each iteration. Eq. (3) and Eq. (4) are used to calculate the coefficient vectors A and C.

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \quad (3)$$

$$\vec{C} = 2\vec{a} \quad (4)$$

Eq. (3) and Eq. (4) contain a coefficient vector a that gradually diminishes from 2 to 0 during the iterations and an unpredictable vector r that change from 0 to 1.

The bubble net attacking strategy serves as the basis for modeling the attacking behavior. To simulate the bubble-net behavior of humpback whales, two main approaches are employed. These approaches are the shrinking encircling mechanism and the spiral updating position mechanism, as depicted in Fig. 3. Humpback whales can utilize either of these mechanisms to capture their prey, with each mechanism occurring randomly with a 50% probability. To introduce randomness, a variable "p" is introduced that ranges between 0 and 1. In the shrinking encircling mechanism, the value of "a" in Eq. (3) is progressively decreased. The value of "A" is determined randomly within the range of [-a, a], while "a" itself decreases gradually from two to zero over a specified number of iterations.

On the other hand, the spiral updating position mechanism mimics the helix-like maneuvering pattern of humpback whales. During the prey attack phase, the optimization algorithm focuses on exploiting potential solutions. Exploitation involves searching within a restricted yet promising region of the search space, with the aim of improving the quality of solutions in the vicinity of a solution denoted as 'S'. This operation intensifies and refines the search around 'S'. The value of "A" ranges between -1 and 1. When |A| is less than 1, exploitation is induced, and all search agents converge to obtain the best possible solution. The updating model can be described using Eq. (5).

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A}.\vec{D} & if\ p < 0.5 \\ \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(t) & if\ p \geq 0.5 \end{cases} \quad (5)$$

During the exploration phase, the focus shifts towards exploring new areas of the search space in order to discover potentially better solutions resembling a global search. This phase relies on the variation of the vector "A" to mobilize the search agents. By setting the absolute value of "A" greater than 1, the search agent is forced to diverge significantly in the search space. This stands in stark contrast to the exploitation phase. In the exploration phase, the search agents update their positions by referencing a randomly selected search agent rather than the best search agent, as in the exploitation phase. This introduces a stochastic element into the search process, enabling the exploration of different regions. The searching mechanism can be mathematically represented by Eq. (6) and Eq. (7).

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_{rand}} - \vec{X}\right| \quad (6)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D} \quad (7)$$

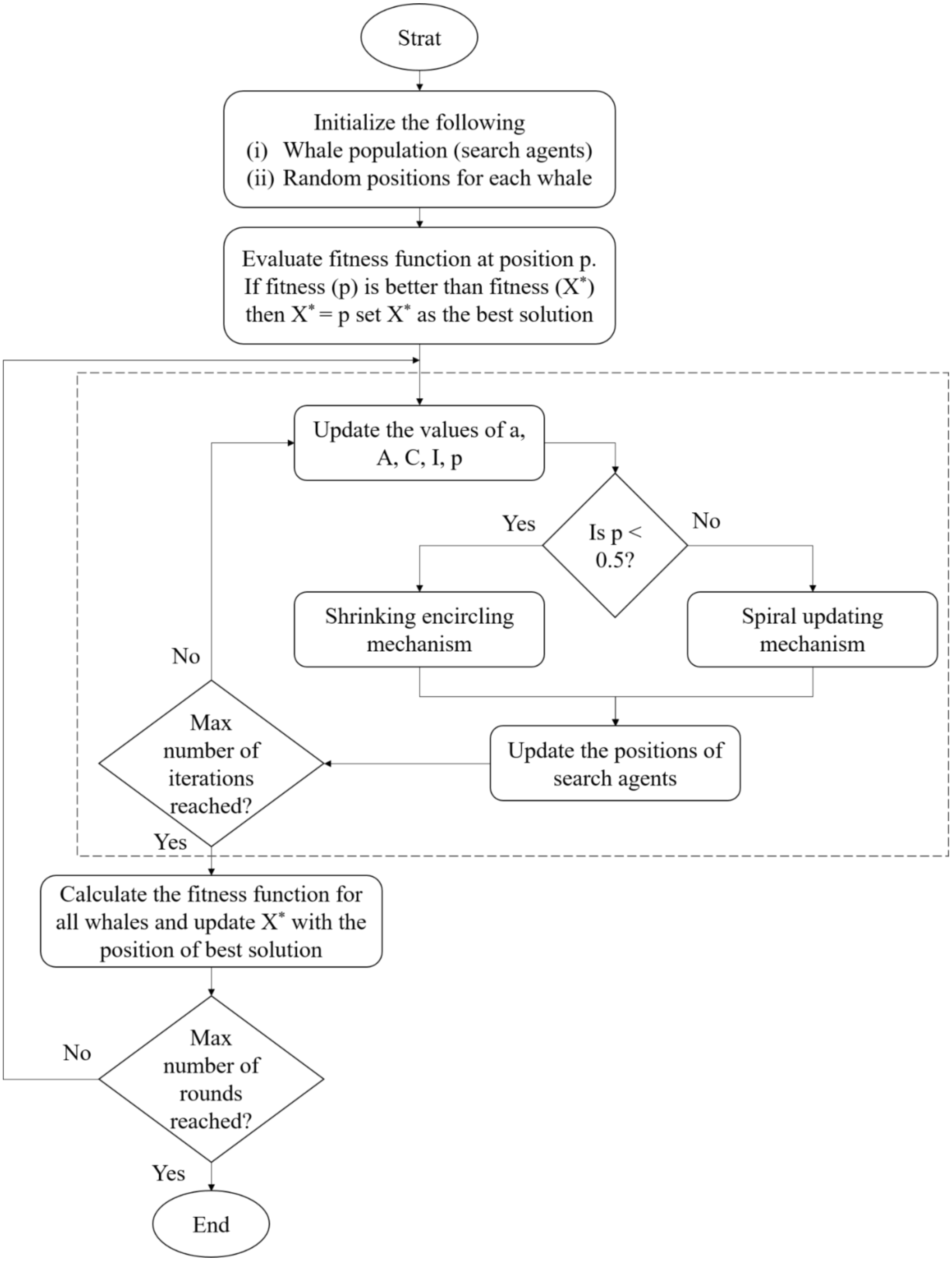
Strat
Initialize the following
(i) Whale population (search agents)
(ii) Random positions for each whale
Evaluate fitness function at position p.
If fitness (p) is better than fitness (X*)
then X* = p set X* as the best solution
Update the values of a,
A, C, I, p
Yes
Is p < 0.5?
No
Shrinking encircling mechanism
Spiral updating mechanism
No
Max number of iterations reached?
Update the positions of search agents
Yes
Calculate the fitness function for all whales and update X* with the position of best solution
No
Max number of rounds reached?
Yes
End

Fig. 4. Flowchart of the proposed algorithm.

The IoT task allocation problem is addressed using the WOA, where the solution is mapped to the positions of whales. The allocation results are represented by a matrix $x = [x_{ij}]$, where i and j represent the indices of tasks and resources, respectively. If task T_i is allocated to resource R_j, then $x_{ij} = 1$. Conversely, if $x_{ij} = 0$, it indicates that the corresponding whale group's position can be used for task allocation. The WOA simulates the movement and feeding behaviors of whales to find the optimal positions for whale groups, which correspond to the optimal allocation of IoT tasks. The flowchart of the proposed algorithm is shown in Fig. 4. The process of IoT task allocation using the WOA typically involves the following steps:

- Initialization: Initialize the whale population and set the parameters for the optimization process, such as the maximum number of iterations and the convergence threshold.
- Fitness evaluation: Evaluate the fitness of each whale group based on the task allocation solution represented by the matrix x. The fitness function considers various factors, such as energy efficiency, stability, and resource utilization.
- Updating the best solution: Keep track of the best task allocation solution found so far, and update it whenever a better solution is discovered during the optimization process.
- Movement and feeding: Apply the movement and feeding behaviors of whales to update the positions of the whale groups. This step involves exploring the search space to find better task allocation solutions.
- Exploration and exploitation: Balance the exploration and exploitation of the search space by adjusting the parameters of the WOA, such as the search range and the degree of randomness in the movement and feeding behaviors.
- Convergence check: Check if the optimization process has converged, either by reaching the maximum number of iterations or by achieving a satisfactory level of task allocation performance.
- Termination: End the optimization process and output the best task allocation solution obtained.

IV. Results and Discussion

In this section, a comparative analysis of the proposed task allocation method is presented, taking into account the application DAG and connection costs, as depicted in Fig. 1. MATLAB is selected as the programming platform for its capability in matrix calculations, practical functions, and seamless data exchange with different programming languages. It offers high-level programming capabilities and is widely used for computational methods and mathematical problem-solving in academia. For the simulation of the proposed method, MATLAB R2022 is utilized. The dataset from [27] serves as the foundation for this study. Task allocation policies in the IoT have a direct impact on application performance and efficient resource utilization. Finding the optimal approach for task assignment in the IoT presents a significant challenge. The procedure for task assignment in the IoT environment is as follows: initially, tasks and resources are mapped based on the available task and resource data. This mapping ensures task completion and QoS by assigning appropriate resources to tasks. Finally, the submitting user receives an overview of the results, providing an overall understanding of the task allocation outcomes. The convergence diagram for the proposed method is illustrated in Fig. 5. The results indicate that approximately 350 generations of the proposed algorithm were required to achieve the correct answer, showcasing the effectiveness and convergence of the proposed approach.

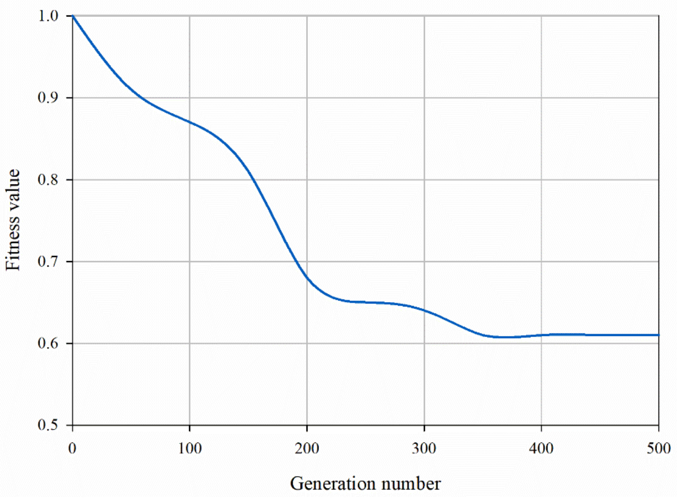

Fig. 5. Convergence diagram.

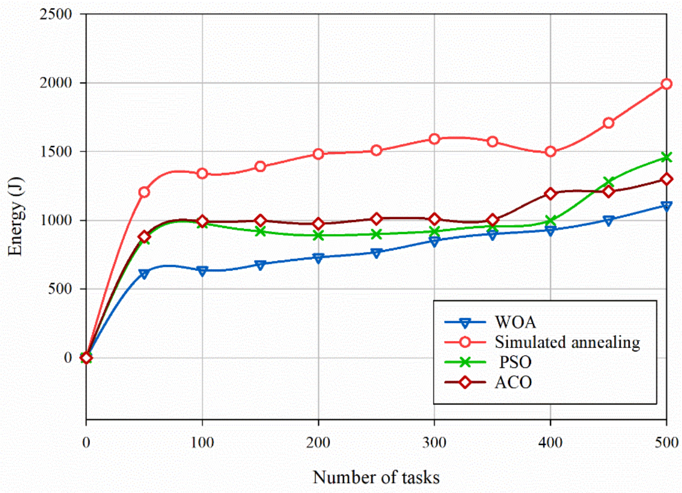

Fig. 6. Energy consumption comparison.

Task allocation in the IoT plays a significant role in energy consumption. In IoT architectures, data centers have the potential to optimize their energy usage by employing strategies such as migrating low-load or idle servers to other servers, shutting down underutilized servers, and transferring idle workloads to different servers. Fig. 6 presents the energy consumption of various algorithms for different task counts. The results demonstrate that as the number of tasks increases, the suggested algorithm exhibits superior energy efficiency compared to other algorithms. This indicates that the proposed algorithm effectively manages energy consumption, making it a favorable choice for IoT task allocation. The IoT offers a favorable environment for users, providing access to software and hardware resources at a reduced cost. In such a network, effective allocation of resources is crucial, benefiting both IoT service providers and ensuring prompt response to user requirements. One of the most critical challenges in the IoT is cost-effective resource allocation. Therefore, it is vital to employ efficient algorithms to address this issue. Fig. 7 illustrates the cost diagram of the proposed method in comparison to other algorithms, highlighting its effectiveness in reducing the overall cost. The diagram illustrates the superior capability of the proposed method in minimizing costs when compared to alternative approaches. This emphasizes the algorithm's power in optimizing resource allocation and achieving cost efficiency in the IoT environment.

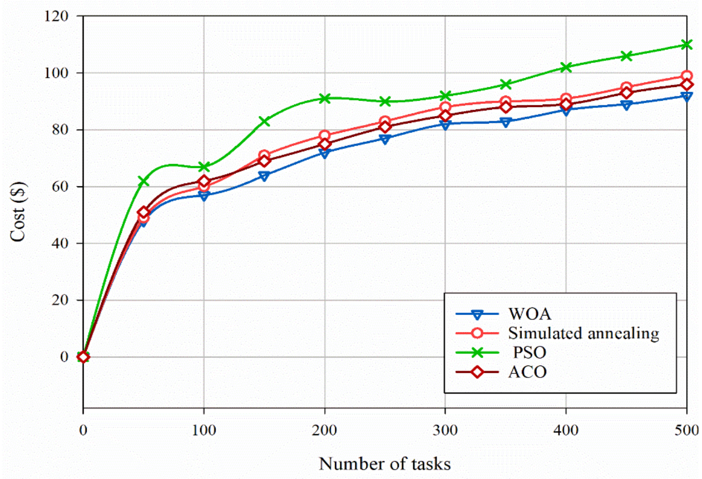

Fig. 7. Cost comparison.

V. Conclusion

With the rapid advancement of IoT technology, the convergence of multiple networks has become the prevailing trend in the IoT landscape. This paper addresses the challenge of optimizing network resource allocation in multiple heterogeneous IoT networks to facilitate seamless information exchange across different services. To achieve this, the proposed approach utilizes a resource allocation algorithm based on the WOA. The primary objectives of this algorithm are to enhance energy efficiency and minimize network latency within IoT networks. Compared to other optimization algorithms such as ACO, simulated annealing, and PSO, our proposed solution demonstrates superiority in selecting optimal assignment decision policies for task allocation and scheduling. Optimizing local and global processes, our approach promises energy efficiency and low-latency communication in the heterogeneous IoT environment. We plan to explore task capability similarity to predict task execution paths in our future work, which could lead to better outcomes. Furthermore, we aim to compare task capabilities to effectively coordinate them, leading to improved outcomes in IoT resource allocation.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] M. K. Hasan et al., "A review on security threats, vulnerabilities, and counter measures of 5G enabled Internet‐of‐Medical‐Things," IET Communications, vol. 16, no. 5, pp. 421-432, 2022.

[8] T. Samizadeh Nikoui, A. M. Rahmani, A. Balador, and H. Haj Seyyed Javadi, "Internet of Things architecture challenges: A systematic review," International Journal of Communication Systems, vol. 34, no. 4, p. e4678, 2021.

[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] A. Peivandizadeh and B. Molavi, "Compatible authentication and key agreement protocol for low power and lossy network in IoT environment," Available at SSRN 4194715, 2022.

[11] J. Jiao, Y. Sun, S. Wu, Y. Wang, and Q. Zhang, "Network utility maximization resource allocation for NOMA in satellite-based Internet of Things," IEEE Internet of Things Journal, vol. 7, no. 4, pp. 3230-3242, 2020.

[12] S. Habib, S. Aghakhani, M. G. Nejati, M. Azimian, Y. Jia, and E. M. Ahmed, "Energy management of an intelligent parking lot equipped with hydrogen storage systems and renewable energy sources using the stochastic p-robust optimization approach," Energy, p. 127844, 2023.

[13] S. Mahmoudinazlou, A. Alizadeh, J. Noble, and S. Eslamdoust, "An improved hybrid ICA-SA metaheuristic for order acceptance and scheduling with time windows and sequence-dependent setup times," Neural Computing and Applications, pp. 1-19, 2023.

[14] V. Hayyolalam, B. Pourghebleh, M. R. Chehrehzad, and A. A. Pourhaji Kazem, "Single‐objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends," Concurrency and Computation: Practice and Experience, vol. 34, no. 5, p. e6698, 2022.

[15] T. Gera, J. Singh, A. Mehbodniya, J. L. Webber, M. Shabaz, and D. Thakur, "Dominant feature selection and machine learning-based hybrid approach to analyze android ransomware," Security and Communication Networks, vol. 2021, pp. 1-22, 2021.

[16] 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.

[17] 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.

[18] X. Cai, S. Geng, D. Wu, J. Cai, and J. Chen, "A multicloud-model-based many-objective intelligent algorithm for efficient task scheduling in internet of things," IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9645-9653, 2020.

[19] H. R. Boveiri, R. Khayami, M. Elhoseny, and M. Gunasekaran, "An efficient Swarm-Intelligence approach for task scheduling in cloud-based internet of things applications," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 9, pp. 3469-3479, 2019.

[20] S. Javanmardi, M. Shojafar, V. Persico, and A. Pescapè, "FPFTS: A joint fuzzy particle swarm optimization mobility‐aware approach to fog task scheduling algorithm for Internet of Things devices," Software: Practice and Experience, 2020.

[21] B. Bu, "Multi-task equilibrium scheduling of Internet of Things: A rough set genetic algorithm," Computer Communications, vol. 184, pp. 42-55, 2022.

[22] D. Weikert, C. Steup, and S. Mostaghim, "Availability-Aware Multiobjective Task Allocation Algorithm for Internet of Things

Networks," IEEE Internet of Things Journal, vol. 9, no. 15, pp. 12945-12953, 2022.

[23] M. S. Bali et al., "An efficient task allocation framework for scheduled data in edge based Internet of Things using hybrid optimization algorithm approach," Physical Communication, vol. 58, p. 102047, 2023.

[24] 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.

[25] Z. Aghapour, S. Sharifian, and H. Taheri, "Task offloading and resource allocation algorithm based on deep reinforcement learning for distributed AI execution tasks in IoT edge computing environments," Computer Networks, vol. 223, p. 109577, 2023.

[26] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in engineering software, vol. 95, pp. 51-67, 2016.

[27] D. M. Abdelkader and F. Omara, "Dynamic task scheduling algorithm with load balancing for heterogeneous computing system," Egyptian Informatics Journal, vol. 13, no. 2, pp. 135-145, 2012.