



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 14, Issue 4, April 2025**

# **Optimizing Resource Allocation in IoT systems: Comparative Analysis on Optimization and Scheduling Algorithms**

**Abdullah Mohammed Zaid, Nitheeshwar B R, Sauban Imtiaz Khan, Shashank M**

Dept. of CSE, Reva University, Bengaluru, Karnataka, India

**Dr. Annoja Ali**

Assistant Professor, Dept. of CSE, Reva University, Bengaluru, Karnataka, India

**ABSTRACT:** Managing resources efficiently is essential for the success of IoT systems, where challenges like dynamic environments and diverse constraints make optimization difficult. In this paper, we review ten widely-used algorithms, including Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Simulated Annealing, Differential Evolution (DE), Genetic Algorithms (GAs), Grey Wolf Optimization (GWO), Whale Optimization Algorithm (WOA), Q-Learning, Game-Theory-Based Preference-based Stable Matching (PBSM), and Fuzzy-LogicBased Algorithms. We explore their methodologies, compare their strengths and weaknesses across five key parameters—scalability, convergence speed, resource utilization efficiency, Quality of Service (QoS), complexity— and discuss how they address IoT-specific challenges. The performance evaluation of these five algorithms shows Game-Theory PBSM outperforms in terms of low latency of 105ms and high throughput of 55 task per second with an optimal CPU utilization of 91%. By highlighting their limitations and potential improvements, this survey offers insights to guide researchers and practitioners in optimizing IoT resource management effectively.

**KEYWORDS:** ACO-Ant Colony Optimization, Game Theory based PBSM-Preference based Stable Matching, IoT-Internet of Things, IoT Resource Allocation, QoS-Quality of Service

## **I. INTRODUCTION**

The rapid growth of the IoT has revolutionized various industries, from healthcare to smart cities, by enabling seamless connectivity between devices. However, as IoT systems become increasingly complex, the efficient allocation of limited resources—such as computational power, bandwidth, and energy—has emerged as a significant challenge. To address this, researchers have proposed various algorithms, each designed to optimize resource management while meeting the dynamic and diverse demands of IoT environments. This paper presents a comprehensive study on resource allocation and optimization in IoT systems, with a focus on the allocation of critical resources—CPU, memory, and bandwidth. In the literature survey section, we reviewed several recent research papers on resource management algorithms for IoT, examining the strengths and limitations of various techniques, such as Q-Learning, DE algorithm, WOA, FuzzyBased methods, and Game Theory-based algorithms. Each of these approaches has shown promising results in specific use cases, but their applicability in real-time IoT scenarios remains an ongoing challenge. We identify gaps in existing research, particularly in the dynamic allocation of resources under stringent constraints, which motivate the experimentation carried out in this study. Building on the insights from the literature, our experiment involved implementing five algorithms—Q-Learning, DE Algorithm, WOA, Fuzzy-Based Algorithm, and Game Theory-based PBSM Algorithm—to allocate and optimize CPU, memory, and bandwidth in a simulated IoT environment. The central processor in our experiment was a Raspberry Pi, while two Arduino boards served as slave devices. We analyzed the performance of each algorithm under varying conditions, focusing on resource utilization, latency, and task throughput. In the comparative analysis section, we discussed the results in detail, providing a thorough evaluation of each algorithm's strengths and weaknesses. Finally, in the Technical Limitations and Solutions section, we identified the challenges faced by these algorithms and proposed potential solutions to enhance their performance in real-world IoT systems.

## II. LITERATURE SURVEY

Zannou et al. [1] presents a solution to the critical problem of task allocation in IoT systems, which face challenges due to heterogeneous devices and limited resources. The proposed methodology builds on the ACO algorithm, a metaheuristic approach inspired by the foraging behavior of ants. The main goal of is to optimize task allocation by identifying nodes with the most suitable capabilities and minimizing the path length required for task execution. This also focuses on two primary factors for optimization: the shortest path distance between nodes and the specific capabilities required by each task. Algorithm used identifies the most capable nodes along an optimal path which ensures that resource consumption is minimized and the network remains stable. From the simulations carried out, its methodology is effective in improving resource utilization and task execution efficiency, making it adaptable to various IoT scenarios where resource constraints and dynamic network conditions are critical concerns. [1] Yousif et al. [2] focuses on optimizing task clustering and scheduling in IoT edge computing systems using a DE algorithm. IoT edge computing processes data closer to the source, minimizing latency compared to cloud solutions. The authors identified that clustering and scheduling tasks on edge devices face significant challenges due to diverse task characteristics and resource constraints. The proposed methodology employs an evolutionary optimization approach, where IoT tasks are grouped based on their similarity and assigned to appropriate resources. This method was benchmarked against other optimization algorithms such as the FA and PSO. Extensive simulations were conducted, replicating real-world IoT edge environments, to validate the efficacy of the DE mechanism under both lightweight and heavyweight workloads [2]. Pandian et al. [3] emphasizes the application of Simulated Annealing Technique (SAT) as a robust optimization algorithm for IoT systems. This technique is particularly beneficial for handling large-scale data generated by IoT devices and addressing the complexities of resource allocation. The methodology begins with data preprocessing using a contrast-limited adaptive histogram equalization (CLAHE) model, which enhances the image or data contrast. Following this, the segmented data is processed through a Multilayer Perceptron (MLP) for classification. This hybrid approach combines the L-BFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) algorithm with the Modified Water Wave Optimization (MWO), making it well-suited for real-time data management in IoT environments. The system leverages edge and cloud computing platforms to enhance data processing efficiency and energy management in smart city applications, including healthcare and traffic systems [3]. Chauhan et al. [4] explores resource allocation challenges in IoT environments, focusing on PSO as a solution. It reviews various optimization methods, including GAs, ACO, and Artificial Bee Colony (ABC), highlighting their comparative effectiveness. PSO, chosen for its ability to solve complex, multi-dimensional problems, is applied to optimize task scheduling and resource allocation in IoT systems. By simulating the social behavior of particles, PSO explores the solution space efficiently, achieving near-optimal results. The author demonstrates its use in methodologies like multipurpose task assignment, cost optimization, and balancing computational loads in cloud and fog computing environments. PSO reduces task execution time, latency, and costs while enhancing throughput and service provisioning. By integrating with cloud and fog computing, PSO dynamically manages resources, addressing the scalability and complexity challenges of IoT systems, and improving overall performance in distributed environments [4]. Wu et al. [5] proposes BRAD-GWA, an enhanced GWO algorithm for IoT resource allocation in the HIT-IHC healthcare system. GWO, known for solving complex optimization problems, models resources as prey and services as wolves. BRAD-GWA improves GWO with density-aware initialization, a refined social hierarchy to avoid local optima, and a lifetime elimination mechanism for underperforming wolves. This variant efficiently balances profit and cost, outperforming existing methods and optimizing resource use across heterogeneous IoT scenarios [5]. Liu et al. [6] applied Q-learning to optimize resource allocation in IoT edge computing, enabling devices to decide between local execution or task offloading. The problem was modeled as a Markov Decision Process (MDP), considering dynamic factors like channel conditions and resource availability. An -greedy Q-learning approach minimized long-term costs, balancing immediate resource usage with future impact. Simulations showed it outperformed traditional methods, achieving improved power efficiency and reduced latency, optimizing IoT network performance in unpredictable environments [6]. Sangaiah et al. [7] proposes using the WOA to solve the resource allocation problem in IoT systems. The algorithm optimizes resource distribution across gateways by minimizing communication costs and balancing loads. It initializes a population of candidate solutions (whales) that iteratively evolve through exploration, exploitation, and convergence, simulating the whales' social hunting behavior. The proposed WOA is evaluated through simulations and is shown to outperform other heuristic methods in reducing communication costs and improving scalability [7]. Chen et al. [8] presents a biobjective optimization methodology using a genetic algorithm tailored for heterogeneous cloud environments. It focuses on microservice placement to optimize resource utilization and reduce network overhead. The algorithm employs empirical parameter settings, enabling efficient workload distribution in real-world IoT systems while addressing

challenges in resource heterogeneity and communication reliability [8]. Hatti et al. [9] presents a novel methodology using a Fuzzy Inference System (FIS) to classify and allocate resources for jobs in IoT systems. The methodology is built around the classification of jobs into "emergent" and "non-emergent" categories based on their resource requirements. Devices gather information periodically and transmit it to a local processor (LP) for initial classification. This is followed by a global processor (GP) calculating resource demands and communicating with the cloud to determine resource availability. By prioritizing highcritical jobs using fuzzy logic, the proposed system achieves efficient resource allocation [9].

### III.METHODS

**A. Q-Learning** In our experiment, we applied a Q-Learning algorithm to optimize the allocation of CPU, memory, and bandwidth resources across IoT devices. The system architecture comprised a Raspberry Pi as the central processor and two Arduino boards as slave devices, each with limited processing capabilities. The Raspberry Pi acted as the decision-making agent, learning to allocate tasks efficiently to the Arduinos to minimize execution time and maximize resource utilization. The QLearning algorithm maintained a Q-table  $Q(s, a)$ , where each entry represented the expected cumulative reward for assigning a task with specific resource requirements to an Arduino in a given state  $s$ . The state  $s$  encoded the current resource availability (CPU, memory, bandwidth) on each Arduino, and the action  $a$  represented the assignment of a task to one of the Arduinos. At each decision step, the Raspberry Pi selected an action using an  $\epsilon$ -greedy policy: with probability  $\epsilon$ , the system explored by choosing a random task-device pair, and with probability  $1 - \epsilon$ , it exploited the action with the highest Qvalue. The Q-value was updated using the formula:  $Q(st, at) \leftarrow (1-\alpha) \cdot Q(st, at) + \alpha \cdot (Rt + \gamma \cdot \max_{a'} Q(st+1, a') - Q(st, at))$  (1) where  $\alpha$  is the learning rate,  $\gamma$  is the discount factor, and  $Rt$  is the immediate reward. The reward  $Rt$  was defined based on task execution success and resource utilization efficiency:  $Rt = \frac{\text{Total Allocated Resources}}{\text{Execution Time} + \epsilon}$  (2) where  $\epsilon$  is a small constant to prevent division by zero. The results demonstrated a significant improvement in resource allocation and task execution efficiency. The QLearning algorithm dynamically distributed tasks between the two Arduino boards, balancing their workloads and preventing resource bottlenecks. As the system iteratively updated the Q-values, it converged to an optimal policy that minimized task execution delays and maximized resource utilization. By the end of the experiment, the system effectively reduced execution times and increased the number of tasks successfully completed within the resource constraints of the Arduinos. This demonstrated the capability of reinforcement learning to adaptively manage resource allocation in real-time IoT environments, making it suitable for applications such as smart city traffic management and other IoT-based resource optimization scenarios.

**B. Differential Evolution Algorithm** In this experiment, the DE algorithm was applied to optimize the allocation of CPU, memory, and bandwidth across a system consisting of a Raspberry Pi as the central processor and two Arduino boards as slave devices. Each candidate solution represented a possible allocation of tasks to the arduinos, with resource requirements mapped as vectors in a multidimensional space. The DE process began by initializing a population of candidate solutions. Mutation generated a donor vector  $v_i$  by perturbing a base vector  $xr1$  with the scaled difference between two other randomly selected vectors  $xr2$  and  $xr3$ , as given by:  $v_i = xr1 + F \cdot (xr2 - xr3)$  (3) where  $F$  is the scaling factor controlling the amplification of the differential variation. Crossover combined the donor vector  $v_i$  with the current solution  $x_i$  to produce a trial vector  $u_i$ , defined as:  $u_{ij} = (v_{ij} \text{ if } \text{rand}(0, 1) \leq CR \text{ or } j = \text{rand}(1, N) \text{ otherwise } x_{ij})$  (4) where  $CR$  is the crossover probability, ensuring diversity in the population. Selection then evaluated the fitness of the trial vector based on the objective function, which minimized the total execution time and resource usage. The DE algorithm iteratively refined resource allocation, converging toward an optimal solution by balancing CPU, memory, and bandwidth usage across the Arduino boards. The fitness function  $Z$  was defined as:  $Z = \sum_{i=1}^N \sum_{j=1}^M \sum_{r=1}^K X[i][j][r] \cdot E[i][j][r]$  (5) where  $E[i][j][r]$  represents the execution time of task  $i$  in cluster  $j$  on resource  $r$ , and  $X[i][j][r]$  is a binary variable indicating task-resource assignment. The results demonstrated that the DE algorithm effectively distributed tasks, reducing execution time and improving resource utilization compared to static methods. This dynamic, adaptive approach proved well-suited for resource-constrained IoT environments, highlighting its potential for real-time applications like edge computing and smart city traffic management.

**C. Whale Optimization Algorithm** In the experiment, the WOA was employed to allocate CPU, memory, and bandwidth resources in a distributed IoT system with a Raspberry Pi as the central processor and two Arduino boards as slave devices. The algorithm mimics the social hunting behavior of humpback whales to find optimal resource allocations. Initially, a population of candidate solutions, represented as whales, is generated, where each solution corresponds to a resource allocation strategy. The whales' positions are updated in each iteration based on their distance from the best-known solution,  $X^*$ , using the formula:  $X(t+1) = X^*(t) - A \cdot |C \cdot X^*(t) - X(t)|$  (6) where  $A$  and  $C$  are coefficient vectors controlling the exploration-exploitation balance. In the bubble-net attack phase, whales either shrink the search space or

spiral around the best solution to refine resource allocation further. The fitness function, which minimizes execution time and optimizes resource usage, guides the search. The results showed that the WOA effectively balanced the resource distribution, improving task execution efficiency and minimizing system latency.

**D. Fuzzy-Based Allocation Algorithm** In this experiment, a Fuzzy-Based Resource Allocation Algorithm was utilized to optimize the distribution of CPU, memory, and bandwidth in a distributed IoT system, with a Raspberry Pi acting as the central processor and two Arduino boards as slave devices. The algorithm employed a Fuzzy Inference System (FIS) to classify tasks based on their criticality, assigning each task an urgency value between 0 and 1. The urgency was determined by evaluating the task's computational demand and communication cost. The total resource requirement  $C_{total}$  for each task was calculated as the sum of communication cost  $C_{comm}$  and computational cost  $C_{comp}$ :  $C_{total} = C_{comm} + C_{comp}$  (7) where  $C_{comm}$  accounted for bandwidth, latency, and energy consumption, and  $C_{comp}$  considered CPU speed, memory, and buffer capacity. The Raspberry Pi, acting as the global processor, received task requests from IoT devices and computed their resource requirements. High-priority tasks, identified by higher urgency values, were allocated resources first, ensuring critical operations were executed promptly. The Arduino boards were queried for available resources, and tasks were distributed to balance load and minimize system latency. This fuzzybased approach demonstrated improved resource utilization, ensuring that high-criticality tasks were completed efficiently while maintaining system stability and minimizing response times.

**E. Game-theory based PBSM Algorithm** In this experiment, resource allocation was carried out using the Game-Theory-based PBSM algorithm, with a Raspberry Pi serving as the central processor and two Arduino boards as slave devices. Each IoT task was modeled as a player in a game where tasks compete for limited edge resources (CPU, memory, and bandwidth). The primary objective was to minimize operational and communication costs while ensuring efficient resource utilization and adherence to resource constraints. The operational cost  $O_c(e)$  for each edge node  $e$  was calculated as:  $O_c(e) = RAME + CPU_e$  (8) where  $RAME$  and  $CPU_e$  represent the memory and processing resources used by the edge node. The communication cost  $T_c(e)$  was modeled as:  $T_c(e) = \sum_{a \in A, e \in E} W_{ae} \cdot B_{ae} + \sum_{e' \in E} W_{e'e} \cdot N_{e'a}$  (9) where  $W_{ae}$  and  $B_{ae}$  are the weight and bandwidth between task  $a$  and edge node  $e$ , and  $N_{e'a}$  is the latency between them. The total objective function aimed to minimize the combined operational and communication costs:  $\min M(O, T) = O_c(e) + T_c(e)$  (10) subject to constraints such as resource usage not exceeding the capacity of the edge nodes:  $\sum_{a \in A} s_{ra} \cdot B_{ae} \leq s_{re}, \forall e \in E, \forall r \in R$  (11) and ensuring that each IoT task is assigned to exactly one edge node:  $\sum_{e \in E} B_{ae} = 1, \forall a \in A$  (12) The PBSM algorithm iteratively updated preference lists for IoT tasks and edge nodes, aiming for a stable match that minimized costs while balancing resource utilization. Results from the experiment demonstrated that PBSM efficiently allocated CPU, memory, and bandwidth across the Raspberry Pi and Arduino boards, leading to reduced latency, improved system throughput, and optimized resource utilization.

#### IV. COMPARATIVE ANALYSIS OF ALGORITHMS

The comparative analysis of five resource allocation algorithms—Q-Learning, DE, WOA, Fuzzy-Based Algorithm, and Game-Theory-Based PBSM—as shown in Table 1 highlights their relative effectiveness in optimizing CPU, memory, and bandwidth across a Raspberry Pi central processor and two Arduino boards as slave devices. Each algorithm was evaluated for resource utilization, latency, and task throughput, providing insights into their suitability for dynamic IoT environments. The PBSM algorithm emerged as the most efficient, demonstrating superior resource utilization and minimal latency, making it an ideal candidate for realtime IoT applications where resource constraints are critical. The Game-Theory-Based PBSM algorithm achieved the highest resource utilization, with 91.8% CPU, 84.6% memory, and 82.7% bandwidth usage, significantly outperforming the other algorithms. This high utilization is attributed to the algorithm's ability to model resource allocation as a preference-based matching problem, ensuring stable matches between IoT tasks and available resources. With a latency of just 105 ms and a throughput of 53 tasks per second, PBSM effectively balances resource allocation while minimizing response time. Its capacity to reduce operational and traffic costs simultaneously underscores its effectiveness in largescale IoT systems with stringent performance requirements. The DE algorithm also performed well, offering a resource utilization of 89.5% for CPU, 81.3% for memory, and 78.2% for bandwidth. DE's iterative mutation, crossover, and selection operations enabled it to explore a wide range of solutions and converge towards optimal task-resource mappings. With a latency of 110 ms and a throughput of 50 tasks per second, DE provided a competitive alternative to PBSM, particularly in environments where dynamic resource . Similarly, the WOA showed comparable performance, though slightly less efficient, with a latency of 115 ms and a throughput of 48 tasks per second, indicating its suitability for moderately time-sensitive IoT applications. In contrast, the Q-Learning and Fuzzy-Based algorithms displayed lower resource utilization and throughput. Q-Learning achieved 83.2% CPU, 77.1% memory, and 73.5% bandwidth utilization, with a latency of 120

ms and a throughput of 45 tasks per second. While it excelled in environments requiring adaptive learning, its higher latency limits its use in realtime scenarios. The Fuzzy-Based algorithm, with resource utilization below 80% and a latency of 125 ms, demonstrated its effectiveness in prioritizing tasks based on criticality but was less suited for high-throughput environments. Overall, PBSM and DE stood out as the most efficient algorithms for resource allocation in IoT systems, while Q-Learning and Fuzzy-Based methods offered balanced performance for less time-sensitive applications.

## V. TECHNICAL LIMITATIONS AND ITS SOLUTIONS

A. Q-Learning Q-Learning faces several challenges when applied to IoT systems. One major issue is the curse of dimensionality, where the state-action space grows exponentially with the number of devices and resources, making the algorithm computationally expensive and slower to converge. Additionally, the cold start problem occurs because the algorithm requires extensive exploration before reaching optimal policies, leading to inefficiencies in the initial stages. Another challenge is its inability to adapt quickly to dynamic environments, such as changes in resource availability or traffic patterns. These challenges can be addressed by integrating Deep Q-Learning (DQL), which uses neural networks to manage high-dimensional spaces effectively. The cold start problem can be alleviated using transfer learning, where pre-trained models from similar environments are reused to jumpstart the learning process. Moreover, implementing adaptive exploration strategies, such as dynamic epsilon-greedy, can improve performance by balancing exploration and exploitation in real-time scenarios.

B. DE Algorithm DE often struggles with premature convergence, where the algorithm settles on suboptimal solutions, especially in complex IoT environments. It also lacks fine-tuning capabilities for meeting strict QoS requirements. Additionally, the algorithm's performance is highly sensitive to its parameter settings, such as mutation and crossover rates, requiring careful tuning. These challenges can be mitigated by adopting adaptive parameter control, where the algorithm dynamically adjusts its mutation and crossover rates during execution. To improve solution quality and avoid premature convergence, DE can be combined with local search techniques, creating a hybrid approach that balances exploration and exploitation. Furthermore, constraint-handling methods can enhance the algorithm's ability to satisfy QoS requirements, ensuring better resource allocation and task scheduling.

C. WOA The WOA faces challenges with local optima, where it may get trapped, especially in non-linear IoT resource allocation problems. Its static behavior can also limit adaptability in dynamic IoT environments where resource demands and availability frequently change. Additionally, the algorithm's scalability can be an issue in larger networks, as it struggles to efficiently explore the growing search space. These limitations can be addressed by introducing randomization techniques, such as chaotic maps, to diversify the search and avoid local optima. Enhancing WOA with dynamic behaviors, such as task migration or adaptive strategies, can improve its performance in dynamic settings. Moreover, leveraging parallel computing can help WOA handle larger search spaces efficiently, improving its scalability for complex IoT networks.

D. Fuzzy-based Algorithm Fuzzy-Based algorithms face the challenge of rule complexity, as the number of input parameters increases, leading to an exponential growth in the rule base. Additionally, traditional fuzzy systems have limited learning ability, relying on manually defined rules and membership functions, which can be labor-intensive and prone to errors. Real-time adaptation is another issue, as predefined rules may not respond effectively to rapidly changing conditions in IoT environments. These challenges can be addressed by integrating neuro-fuzzy systems, which combine fuzzy logic with neural networks to learn and optimize rules automatically. The use of hierarchical fuzzy systems can reduce rule complexity by dividing the problem into smaller, more manageable subsystems. Finally, implementing dynamic fuzzy rules that update based on realtime feedback can enhance the algorithm's adaptability and effectiveness in dynamic IoT scenarios.

E. PBSM Algorithm The PBSM algorithm's effectiveness relies heavily on accurately defining preference metrics, which can be challenging in dynamic IoT environments with fluctuating demands. Another significant challenge is the latency in distributed systems, where communication delays between IoT devices and edge nodes can degrade performance. Additionally, scalability issues arise as the network size increases, leading to computational overhead in managing large preference lists. To overcome these challenges, dynamic preference adjustments can be implemented, allowing realtime feedback to refine preference metrics and adapt to changing conditions. Hybrid models combining

centralized and distributed approaches can reduce latency by leveraging global optimization while enabling localized decisions. Furthermore, adopting parallel processing techniques can handle computational demands efficiently, ensuring that preference lists are managed effectively even in large-scale systems.

## VI.CONCLUSION

In this paper, we analyzed and compared Q-Learning, PBSM, DE, WOA and Fuzzy-Based algorithms for resource allocation in IoT systems. Each algorithm offers unique strengths, such as scalability, efficiency, and adaptability, but also faces challenges like complexity, convergence issues, and dynamic adaptability. By addressing these limitations through hybridization, adaptive techniques, and advanced optimization strategies, these algorithms can better meet IoT demands. The choice of algorithm depends on specific application requirements, such as QoS, resource constraints, and network size. Hence, it is clear that the best optimization algorithm for an IOT based system (in our experiment we've used IoT based traffic signal management) is Game-Theory PBSM with a low latency of 105ms, high throughput of 55 task per second and an optimal CPU usage of 91%. A lower latency ensures that data from sensors is processed and acted upon quickly, allowing for dynamic adjustments to traffic signals based on real-time conditions. High throughput is necessary for handling peak traffic times when data from numerous sources must be analyzed quickly. Optimal CPU utilization ensures that the system can process incoming data efficiently without lagging, which is vital for maintaining real-time operations. Future advancements in AI and hybrid models promise more robust solutions for IoT resource optimization challenges.

## REFERENCES

- [1] A. Zannou, A. Boulaam, and E. H. Nfaoui, "A Task Allocation In IoT Using Ant Colony Optimization," 2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS), pp. 1–6, Dec. 2019
- [2] A. Yousif, Mohammed Bakri Bashir, and A. Ali, "An Evolutionary Algorithm for Task Clustering and Scheduling in IoT Edge Computing," *Mathematics*, vol. 12, no. 2, pp. 281–281, Jan. 2024
- [3] P. Senthil Pandian, P. Sundaravadivel, R. Augustian Isaac, P. Hemavathy, E. Vinothkumar, and P. Ramal, "ANALYSING IOT SMART SYSTEM APPLICATION AND ENVIRONMENT BY USING SIMULATED ANNEALING TECHNIQUES," *Journal of Theoretical and Applied Information Technology*, vol. 102, no. 8, 2024. [4] A. S. Chauhan and S. Saini, "RESOURCE ALLOCATION IN IOT ENVIRONMENT USING SWARM OPTIMIZATION: A REVIEW," Jan. 01, 2022.
- [5] J. Wu, H. Dai, Y. Wang, and Z. Tu, "Multi-Scenario Bimetric-balanced IoT Resource Allocation: An Evolutionary Approach," arXiv (Cornell University), Dec. 2022
- [6] X. Liu, Z. Qin, and Y. Gao, "Resource Allocation for Edge Computing in IoT Networks via Reinforcement Learning," *IEEE Xplore*, May 01, 2019
- [7] 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, Jan. 2020
- [8] Q.-H. Chen and C.-Y. Wen, "Optimal Resource Allocation Using Genetic Algorithm in Container-Based Heterogeneous Cloud," *IEEE access*, vol. 12, pp. 7413–7429, Jan. 2024
- [9] D. I. Hatti and A. V. Sutagundar, "Fuzzy based job classification and resource allocation in IoT," 2017 International Conference on Inventive Systems and Control (ICISC), vol. 1, pp. 1–4, Jan. 2017
- [10] Anjan Bandopadhyay et al., "EdgeMatch: A Smart Approach for Scheduling IoT-Edge Tasks with Multiple Criteria using Game Theory," *IEEE access*, vol. 12, pp. 7609–7623, Jan. 2024



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details