Proceedings of the 4th International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2026.32478

Resource Scheduling Strategy and Algorithm Design Based on QoS Constraints in Heterogeneous Edge Environments

Xiazi Zhao

School of Software, Shanxi Agricultural University, Jinzhong, China
2907866139@qq.com

Abstract. Heterogeneous edge computing suffers from severe load imbalance due to node resource differences, and traditional scheduling algorithms fail to synergistically optimize QoS guarantee and load balancing. To address this issue, this paper constructs a multi-dimensional system model integrating QoS constraints and load balancing rules, and designs a two-layer scheduling algorithm: the Location-Aware Heterogeneous QoS Prediction Algorithm (LAQP-H) and the Improved Artificial Bee Colony Load Balancing Scheduling Algorithm (IABC-LB). Simulation experiments with synthetic datasets verify that the proposed strategy outperforms contrast algorithms in average resource utilization, system load imbalance degree, maximum makespan and QoS prediction accuracy (MAE/RMSE). Limitations include ignoring dynamic task access and node failure, which will be optimized with fault-tolerant and adaptive modules in future research.

Keywords: Heterogeneous Edge Computing, QoS Constraints, Resource Scheduling, Load Balancing, QoS Prediction, Artificial Bee Colony Algorithm

1. Introduction

Heterogeneous edge computing deploy edge nodes with different computing power and storage capacity near data sources, it effectively solve the high latency trouble of cloud computing and become the core technology to support new businesses such as autonomous driving and Internet of Things [1]. But as mobile devices and computing tasks grow sharply, the resource difference of heterogeneous edge nodes easily cause serious load imbalance problems. Some nodes is overloaded with massive tasks while others keep idle all the time, and existing resource scheduling algorithm do not fully fit the features of heterogeneous edge environments. These old algorithms ignore the location link between tasks and heterogeneous edge nodes when predicting QoS, and they is hard to realize the collaborative optimization of QoS guarantee and load balancing [2]. Such defects make users' service quality cannot be protected effectively, which restrict the development of edge computing applications. To solve these troubles, this paper focus on the resource scheduling demands in heterogeneous edge environments [3]. We build a multi-dimensional system model that integrate QoS constraints and load balancing rules, and design targeted QoS prediction algorithm and load balancing scheduling algorithm adapting to heterogeneous features [4]. We embed node heterogeneous resource features and location similarity features to boost QoS prediction accuracy, use improved intelligent optimization algorithm to solve the built scheduling model. Finally, we

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.32478

conduct simulation experiments to verify the effectiveness of the proposed strategy and algorithm in heterogeneous edge environments, and check its performance in practical edge computing application scenarios [5].

2. System model

There are many edge nodes with different computing power and storage capacity in heterogeneous edge environments, and these nodes provide computing resources for tasks with diverse QoS demands. Load balancing scheduling under QoS constraints aim to assign suitable resources to tasks, it realize global load balance of nodes on the premise of meeting QoS needs and cut down task transmission and processing delays. The resource allocation scenario of this system are shown in Figure 1, and the overall architecture of the designed scheduling algorithm is shown in Figure 2. Firstly, the QoS prediction algorithm generate a resource pre-allocation queue, then the load balancing module is introduced to adjust the pre-allocation results to fit the resource differences of heterogeneous edge nodes, and finally the optimal resource allocation queue is obtained. We define the set of heterogeneous edge nodes as $S = s_1, s_2, \ldots, s_N$, and each node have heterogeneous CPU, memory and disk resources; the set of tasks generated by mobile devices is $T = t_1, t_2, \ldots, t_M$, each task have unique computing needs and QoS threshold, and the task subset needing node s_j is $T_j = t_{j1}, t_{j2}, \ldots, t_{jK}$. We predict the task-resource QoS matrix via historical interaction data, as shown in Equation (1).

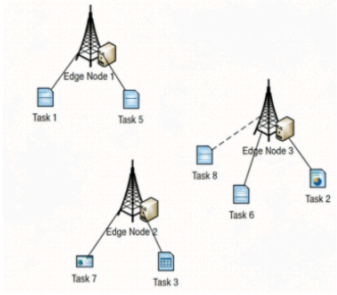


Figure 1. QoS-oriented resource allocation scenario

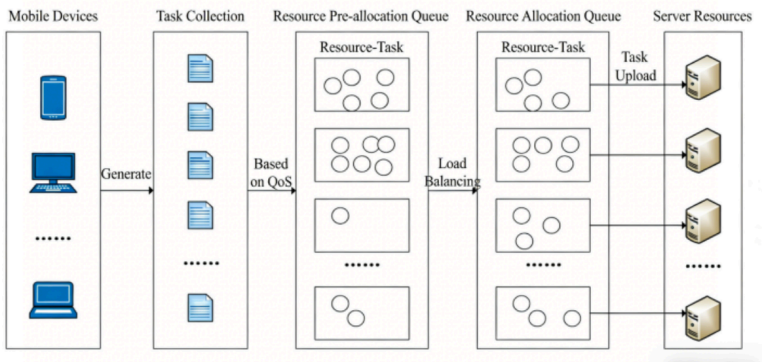


Figure 2. Algorithm model architecture

DOI: 10.54254/2755-2721/2026.32478

$$\widehat{Q} = \begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1N} \\ q_{21} & q_{22} & \cdots & q_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ q_{M1} & q_{M2} & \cdots & q_{MN} \end{pmatrix} \tag{1}$$

Each task t_i have a QoS threshold q_i^{max}, QoS loss occur when allocated nodes cannot meet this threshold, and single-task QoS loss is shown in Equation (2). The total QoS loss of node s_j is the sum of single-task loss, as shown in Equation (3), and the overall system QoS loss is the sum of all nodes' loss, as shown in Equation (4).

$$lq_i = q_i^{\max} - q_{ij} \tag{2}$$

$$Loss_j^{qos} = \sum_{i=1}^{K} lq_i \tag{3}$$

$$Loss_{qos} = \sum_{j=1}^{N} Loss_j^{qos} \tag{4}$$

We evaluate node load from three dimensions: CPU, memory and disk. Set the total CPU, memory and disk of node s_j as SC_j, SM_j and SD_j, the corresponding resources required by task subset T_j are $cpu_j = c_{j1}, c_{j2}, \ldots, c_{jK}$, $memory_j = m_{j1}, m_{j2}, \ldots, m_{jK}$ and $disk_j = d_{j1}, d_{j2}, \ldots, d_{jK}$. The CPU load, memory load and disk load of node s_j are shown in Equation (5), Equation (6) and Equation (7) respectively.

$$LC_j = \frac{\sum_{i=1}^{K} c_{ji}}{SC_j} \tag{5}$$

$$LM_j = \frac{\sum_{i=1}^{K} m_{ji}}{SM_j} \tag{6}$$

$$LD_j = \frac{\sum_{i=1}^{K} d_{ji}}{SD_j} \tag{7}$$

The CPU processing speed of heterogeneous nodes are different, set the speed of node s_j as $CPU(s_j)$ and the size of task t_{ji} as $TaskSize(t_{ji})$. The total processing time of node s_j is shown in Equation (8), and the system makespan is the maximum processing time of all nodes, as shown in Equation (9), which reflect the overall task processing efficiency of the system.

DOI: 10.54254/2755-2721/2026.32478

$$ET\left(s_j\right) = \frac{\sum_{t_{ji} \in T_j} TaskSize\left(t_{ji}\right)}{CPU(s_j)} \tag{8}$$

$$Makespan = Max(ET(s_j)) \tag{9}$$

3. Algorithm design

Aimed at the features of heterogeneous edge environments, this paper design a double-layer scheduling algorithm to solve resource scheduling problems. We first propose the Location-Aware Heterogeneous QoS Prediction Algorithm (LAQP-H), which integrate resource features of heterogeneous edge nodes and location features between tasks and nodes to improve prediction accuracy. Then we design the Improved Artificial Bee Colony Load Balancing Scheduling Algorithm (IABC-LB), that combine chaos algorithm and Q-Learning to modify the traditional artificial bee colony algorithm. This improved algorithm solve the load balancing scheduling model with QoS constraints in heterogeneous scenarios, and realize the optimal allocation of edge resources. The LAQP-H algorithm rely on multi-layer feedforward neural networks, it add resource feature embedding of heterogeneous nodes in the input layer and bring in location similarity adaptive corrector. This structure fix the defect that traditional QoS prediction ignore location factors, and the algorithm model are shown in Figure 3. We input task ID, task location features, heterogeneous node ID and node resource features into the embedding layer to map high-dimensional sparse features to low-dimensional dense features. The embedding mapping of heterogeneous resource features are shown in Equation (10).

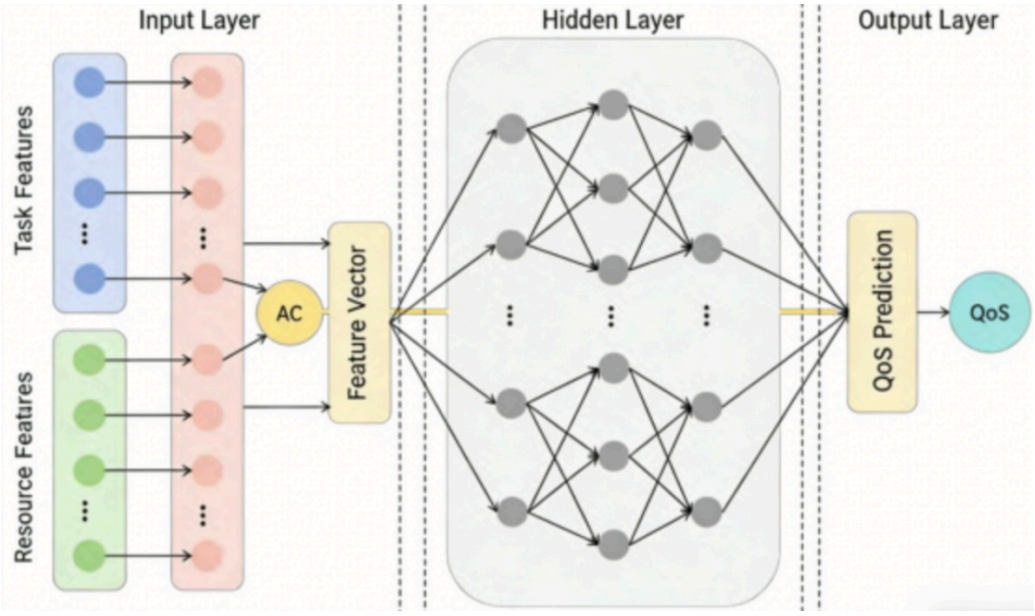


Figure 3. Location-aware QoS prediction model architecture

$$G_s^h = f_1\left(W_s^{hT} g_s^h + b_1\right) \tag{10}$$

DOI: 10.54254/2755-2721/2026.32478

We use Euclidean distance to calculate the location similarity between tasks and heterogeneous edge nodes, as shown in Equation (11), and the result help correct the predicted QoS value. To measure the error between predicted and real QoS values, we design a robust loss function as Equation (12), which switch calculation rules based on the error threshold δ .

$$o^{AC} = \sqrt{\sum_{i=1}^{n} \left(g_{t_i}^{f} - g_{s_i}^{f} \right)^2} \tag{11}$$

$$L_{\delta}\left(Q, \widehat{Q}\right) = \begin{cases} \frac{1}{2}\left(Q - \widehat{Q}\right)^2 & \left|Q - \widehat{Q}\right| \le \delta \\ \delta\left|Q - \widehat{Q}\right| - \frac{1}{2}\delta^2 & others \end{cases} \tag{12}$$

The IABC-LB algorithm optimize population initialization, fitness function and search strategies of traditional artificial bee colony algorithm. It introduce chaos algorithm to avoid local optimum and use Q-Learning to optimize food source selection, which boost search efficiency and solution quality. We generate chaotic population via Logistic mapping to expand search range, as shown in Equation (13). A hybrid fitness function are built to fuse QoS loss, system load and makespan, as shown in Equation (14). We also update Q-value to optimize search strategies, as shown in Equation (15), which make the algorithm fit heterogeneous edge environments better.

$$p_{i+1,j} = \mu p_{i,j} \left(1 - p_{i,j}\right) \tag{13}$$

$$Fit = \lambda_1 F_{QoS} + \lambda_2 F_{load} + \lambda_3 F_{ms} \tag{14}$$

$$Q_{t+1}(st_t, a_t) = Q_t(st_t, a_t) + \alpha \left[r_t + \gamma \max \left[Q_t(\delta(st_t, a_t), a_t\,') \right] - Q_t(st_t, a_t) \right] \tag{15}$$

4. Experimental results and analysis

To verifys the performance of proposed scheduling strategy and algorithm in heterogeneous edge environments, we carry out simulation experiments with synthetic dataset that simulate heterogeneous edge node features. This dataset contain 10000 interaction data between IoT devices and edge nodes, covering task size, computing demands, node resource features and location coordinates, and we build QoS metrics with delay and cost. We select Max_Min, MOCS, HABC as contrast algorithms, and test performance from four dimensions: average resource utilization, system load imbalance, maximum makespan and QoS prediction accuracy. QoS prediction accuracy are measured by Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), and the calculation formulas of MAE and RMSE are shown as Equation (16) and Equation (17). We divide the dataset into small, medium and large parts by task scale, and the average resource utilization of different algorithms are shown in Figure 4. The proposed IABC-LB algorithm have higher resource

DOI: 10.54254/2755-2721/2026.32478

utilization than contrast algorithms under all scales, and it reach the optimal value in large-scale tasks, which increase 7.8% compared with HABC algorithm. The system load imbalance under different task numbers are shown in Table 1, the imbalance value of IABC-LB algorithm drop to 0.097 when task number is 1000, which is much lower than other algorithms. The maximum makespan is shown in Figure 5, the proposed algorithm shorten the makespan obviously with the increase of tasks. The MAE and RMSE of LAQP-H algorithm are shown in Table 2 and Table 3, its prediction error is the lowest among all contrast algorithms, which prove the effectiveness of embedded heterogeneous and location features. All experimental results verifys that the proposed strategy have better performance in load balancing and QoS guarantee.

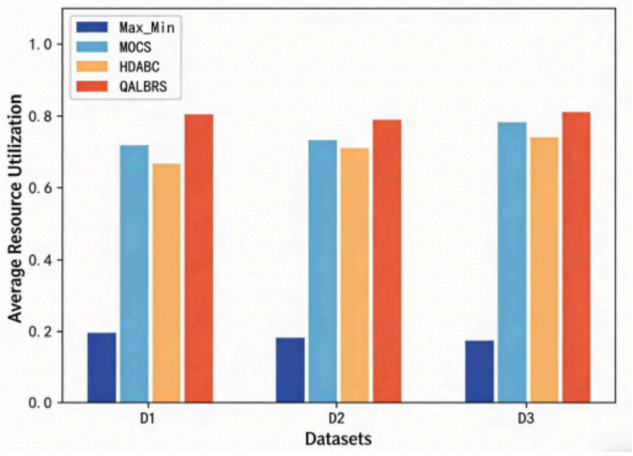


Figure 4. Average resource utilization of algorithms on different datasets

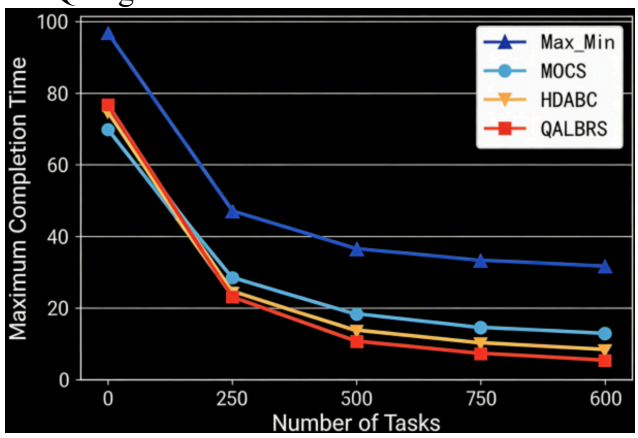


Figure 5. Maximum completion time under different task quantities

Table 1. System load imbalance degree of different algorithms under different numbers of tasks

Number of Tasks	Max_Min	MOCS	HDABC	QALBRS
200	1.347	0.283	0.465	0.214
400	0.989	0.181	0.247	0.168
600	1.112	0.214	0.382	0.117
800	1.169	0.178	0.158	0.114
1000	1.353	0.161	0.176	0.097

Table 2. MAE of each algorithm under different data densities

Algorithm	5%	10%	15%	20%	25%	30%
NFM	0.812	0.776	0.683	0.648	0.627	0.612
LFM	0.609	0.589	0.571	0.556	0.538	0.523
NCF	0.527	0.496	0.477	0.463	0.445	0.429
LAQP	0.497	0.474	0.458	0.439	0.428	0.398

DOI: 10.54254/2755-2721/2026.32478

Table 3. RMSE of each algorithm under different data densities

Algorithm	5%	10%	15%	20%	25%	30%
NFM	1.761	1.705	1.646	1.615	1.583	1.579
LFM	1.643	1.565	1.511	1.494	1.473	1.452
NCF	1.488	1.461	1.427	1.403	1.375	1.355
LAQP	1.459	1.393	1.348	1.314	1.283	1.247

$$MAE = \frac{1}{M}\sum_{i=1}^{M}\left|Q_i - \widehat{Q}_i\right| \tag{16}$$

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(Q_i - \widehat{Q}_i\right)^2} \tag{17}$$

5. Conclusion

This paper focus on the collaborative scheduling problem of QoS guarantee and load balancing in heterogeneous edge environments, and design a complete resource scheduling strategy and algorithm. We build a multi-dimensional system model and propose two targeted algorithms, which effectively improve QoS prediction accuracy and realize load balancing optimization. Simulation results show that the proposed algorithm have obvious advantages in resource utilization, task efficiency and load balance, and it fit the scheduling demands of heterogeneous edge environments well. But this study do not consider dynamic task access and node failure, which are limitations of the research. In the future, we will optimize real-time scheduling algorithms for dynamic edge environments, add fault-tolerant mechanisms and adaptive adjustment modules to boost the robustness and practicability of the whole scheduling system.

References

[1] Yang J, Zhao L, Peng B . Edge-Intelligence-Driven Cooperative Control Framework for Heterogeneous Unmanned Aerial and Surface Vehicles in Complex Maritime Environments [J].Drones, 2025, 9(11): 755-755.DOI: 10.3390/DRONES9110755.

[2] Kazemi F, Barzegar B, Motameni H, et al. An energy-aware scheduling in DVFS-enabled heterogeneous edge computing environments [J].The Journal of Supercomputing, 2025, 81(9): 1078-1078.DOI: 10.1007/S11227-025-07432-2.

[3] Huang F, Wang W, Guo J, et al. Price-aware resource management for multi-modal DNN inference in collaborative heterogeneous edge environments [J].Journal of Parallel and Distributed Computing, 2025, 201105080-105080.DOI: 10.1016/J.JPDC.2025.105080.

[4] Ximing C, Xilong H, Du C, et al. FedMEM: Adaptive Personalized Federated Learning Framework for Heterogeneous Mobile Edge Environments [J].International Journal of Computational Intelligence Systems, 2025, 18(1): 84-84.DOI: 10.1007/S44196-025-00814-7.

[5] Wei Z, Wang J, Zhao Z, et al. Toward data efficient anomaly detection in heterogeneous edge–cloud environments using clustered federated learning [J].Future Generation Computer Systems, 2025, 164107559-107559.DOI: 10.1016/J.FUTURE.2024.107559.