

Research Article

ACPEC: A Resource Management Scheme Based on Ant Colony Algorithm for Power Edge Computing

Zhu Liu^{1,2}, Xuesong Qiu¹, and Nan Zhang²

¹State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

²State Grid Information & Telecommunication Group Co., Ltd, Beijing 102211, China

Correspondence should be addressed to Zhu Liu; 13811068052@139.com

Received 4 June 2021; Accepted 22 September 2021; Published 9 December 2021

Academic Editor: Xin Liu

Copyright © 2021 Zhu Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of power IoTs (Internet of Things) technology, more and more intelligent devices access the network. Cloud computing is used to provide the resource storage and task computing services for power network. However, there are many problems with traditional cloud computing such as the long-time delay and resource bottleneck. Therefore, in this paper, a two-level resource management scheme is put forward based on the idea of edge computing. Furthermore, a new task scheduling algorithm is presented based on the ant colony algorithm, which realized the resource sharing and dynamic scheduling. The data of simulation show that this algorithm has a good effect on the performance of task execution time, power consumption, and so on.

1. Background

With the rapid development of smart grid, plenty of power equipment accesses into the Internet. The demand for power services shows exponential growth trend. For cloud computing technology, it not only can provide computing service but also has many advantages such as scalability, flexibility, and security [1].

Cloud computing can provide technical support and theoretical support for power design, data storage, disaster recovery, intelligent power consumption, and power grid simulation analysis system. Furthermore, cloud data center can integrate and manage the data and resources coming from different business systems in power grid uniformly, and these data and resources are the basic of resource sharing for smart grid. However, the total cost of a cloud will increase when the scale decreases, as shown in a survey of Microsoft [2]. Though the computational resource in the edge node is generally limited and the resource in the remote cloud is abundant [3], the resource features between edge cloud and the remote are different.

Edge computing can reduce the distance between user demand and computing servers [4, 5]. With edge computing, not only the computation latency of these tasks can be shortened but also the requirements of users on computation capability and power supply can be reduced [6]. Therefore, the computation tasks especially the latency-sensitive ones can be uploaded to edge nodes. The computation task can be received and executed locally, and the computation result can be sent to the users directly [7].

Heterogeneous cloud is suggested to better serve users with different Quality of Service (QoS) requirements [8]. Furthermore, to some computation-intensive tasks, tasks need to be divided into some subtasks and be executed by resource cooperation. In this paper, a new two-level resource management scheme is proposed based on edge computing. For the new resource management scheme, the computing tasks can be executed in local nodes, and the time delay can be decreased compared with cloud computing. Furthermore, according to the characteristic of edge computing, we present a task allocation algorithm based on the ant colony algorithm in this paper. The resources of the local edge nodes

can be used for providing computation-intensive tasks by resource sharing and cooperation.

2. Related Work

A rational resource management and task assignment scheme is very important to ensure the efficiency and stability of the power cloud computing system. There are many researchers who studied the resource management of cloud computing [9, 10]. For the resource management schemes on saving energy consumption, the authors in [11] proposed a new method to reduce the power consumption by decreasing the processor speed. The balance of load is considered in [12] and realized by effective virtual machine migration strategy; Furthermore, the authors proposed a new scheme for intelligent load migration and resource allocation by machine learning. For the resource management schemes of increasing resource utilization, the authors in [13] designed a two-level resource scheduling scheme and realized a new resource allocation algorithm by the heuristic algorithm. Furthermore, the authors in [14] proposed a new resource scheduling procedure which can realize a real-time vehicle cloud service. The authors in [15] proposed a QoS-based mobile cloud resource scheduling algorithm, which can minimize the communication and computing consumption. In addition, the genetic algorithm has also been used for saving energy consumption resource scheduling in [16, 17]. Furthermore, based on NOMA technology, the authors in [18, 19] introduced the resource allocation methods for cluster-based cognitive industrial and multi-beam satellite industrial.

For edge computing, one main service requirement is to have a low service delay, which would be the output of its request, i.e., the time it takes for the user to receive its results [20]. The authors of [21] proposed that the problem of time delay can be addressed by executing task applications on resource providers external to the edge device. In addition, to control the data transmission of computation tasks effectively according to the channel information, a delay-optimal problem is studied through adopting a Markov decision process approach [22]. The authors in [23] proposed an effective computation offloading strategy and investigated a green edge computing system with rechargeable devices. Furthermore, the authors in [24, 25] considered the balance between the cost of the system and the mean offloading delay, which studied the workload sharing between the edge and the remote cloud. For communication elements, one existing technique called access point scheduling is considered, in [26, 27], which can associate the user and the closest cloudlet in order to minimize transmission distance and improve signal quality and propagation time. In addition, in [28, 29], transmission power control is proposed where the transmission power of the cloudlets is carefully set with the final goal of lowering the latency. The authors in [30, 31] considered to provide fair services by allocating the available resources, such as channel bandwidth.

For the optimization of the QoS in the edge computing system, there are many research studies. The authors in [32]

considered the joint optimization of radio and computational resources to minimize the overall energy consumption of mobile users in edge computing. To minimize the total energy consumption of users, the authors proposed an algorithm for uplink and downlink beamforming and computational resource allocation [33]. In [34], the authors derived the optimal radio and computational resource allocation policy considering the scenario of TDMA networks. The authors in [35] designed a task offloading algorithm to study the energy-delay tradeoff in edge computing.

Such computing paradigm, called edge computing, has drawn extensive attention from both academy and industry [36, 37]. With the help of edge computing, application tasks running on the devices can be offloaded to the edge servers in edge computing to get better service. Generally speaking, the computing resources at the edge computing system are still limited compared with remote cloud data centers. Therefore, an edge computing is customarily backed by a remote back-end cloud via the Internet [38].

To a certain extent, above research studies solve some resource management problems. However, for the problem of the resource management power information system, there is not much research in the existing literature. Therefore, in this paper, a power resource management scheme based on edge computing is studied.

3. The Power Resource Management Scheme Based on Edge Computing

With the rapid development of cloud computing, more and more intelligent terminals access into the network. For traditional cloud computing, computing resource and storage resource are in the cloud data center and all tasks will be transferred to the cloud. For the power network, the power equipment includes a large amount of data information. Therefore, in power network, resource bottleneck of traditional cloud computing cannot be avoided, and the real-time performance cannot be ensured.

In the edge of power network, there are a lot of devices which include considerable resources. If the resources of these devices can be used for providing computing services, the computing energy of data center can be effectively supplemented [39]. In this paper, a new power cloud resource management scheme based on edge computing is proposed. In this scheme, some computing tasks in cloud servers will be sank to the edge devices, and the local computing tasks will also be performed in local edge devices.

In the new resource management scheme, as shown in Figure 1, there are two-level computing nodes: the first level includes the analysis control nodes which are in the cloud servers and the second level includes computing nodes which are in the edge devices. The first-level nodes are used to perform computation-intensive tasks and manage the second-level nodes, and the second-level nodes are used to perform local computational tasks. The resources in second-level nodes should be subjected to the scheduling of the first-level nodes. The application scenarios of this new resource management scheme can be the power supply service command system and big data platform, the hot spot

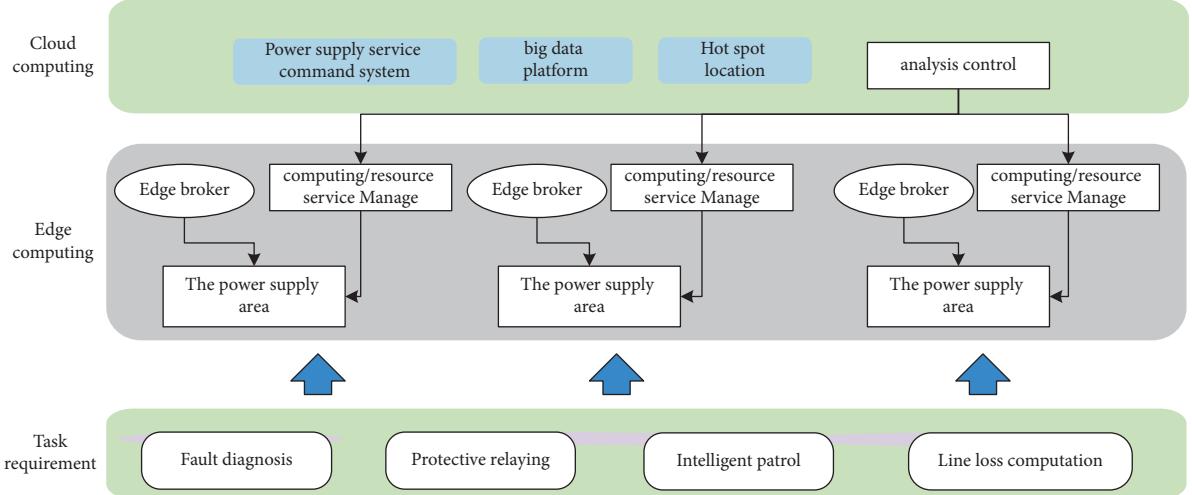


FIGURE 1: The resource management scheme based on edge computing for smart grid.

location, and so on. And the task requirement of this new resource management scheme can be fault diagnosis, protective relaying, intelligent patrol, line loss computation, and so on.

In this resource management scheme, the resource in the second-level nodes is limited and some tasks will be divided into many subtasks and be assigned to different edge nodes. The tasks will be executed by resources sharing and node cooperation.

In this paper, we consider an actual application scenario. As shown in Figure 2, this area represents a distribution power supply area, which includes many transformers, some intelligent terminals, and some users. In this situation, the intelligent terminals are set as the edge nodes to provide task computing services.

4. The Task Scheduling Method Based on Ant Colony Algorithm

The ant colony algorithm is an intelligent optimization algorithm, which is an optimization process for solving complex problems. The ant colony algorithm can be used for resolving resource allocation problem. In the paper, the edge nodes are always resource-limited to execute a computation-intensive task alone. Therefore, the computation-intensive tasks will be divided into many subtasks and be allocated on many different edge nodes. In this paper, we presented a new tasks allocation algorithm based on the ant colony algorithm to allocate appropriate resources for subtasks.

4.1. Introduction of Ant Colony Algorithm. The ant colony algorithm is a novel simulated evolvement algorithm solving complicated combinatorial optimization problem, and its typical feature is swarm intelligence. For the ant colony algorithm, a colony consisting of many ants is considered to perform a highly complex task that an ant alone cannot accomplish. The ant regulates their behavior through collaboration and information exchange. For the ant colony algorithm, pheromones play a very important role, which

can help ants judge the next direction of transfer by sensing the presence and concentration of pheromones.

The ant colony algorithm is a random search algorithm evolved from the behavior of ants foraging. In the process of simulating ant foraging, positive feedback and distributed cooperation are mainly used to find the optimal path. There are some characteristics in the behaviors of ant foraging shown as follows: firstly, the pheromone concentration along the path will change over time and the ant will choose the path according to the pheromone concentration of different paths. Secondly, to avoid the process of path choosing getting trapped in local optimality, the paths that have already been chosen before will not be allowed to be selected again. Thirdly, the pheromone of paths will decrease over time, and the length of the path will affect pheromone concentration.

4.2. The Detailed Design of Task Scheduling Algorithm. Recently, the task scale in power network increases rapidly, so the issue of task resource scheduling is increasingly complex. However, the rational scheduling among resources and tasks is very important for increasing the resource utilization and task completion rate.

For the resource management scheme proposed in this paper, the resources in the edge nodes are limited to complete some computation-intensive tasks along. The computation-intensive tasks will be often divided into some subtasks and be assigned to different edge nodes. Therefore, it is a very key issue that how to assign the tasks among the different edge nodes, which will be solved in this section.

The ant colony algorithm is an intelligent optimization algorithm, which is used to solve complicated combinatorial optimization problem. For the ant colony algorithm, distributed computing is relatively easy to implement and it is easy to merge with other algorithms to become a new algorithm. Because of these advantages, the ant colony algorithm will usually be applied to resolve resource-task scheduling and optimization problem.

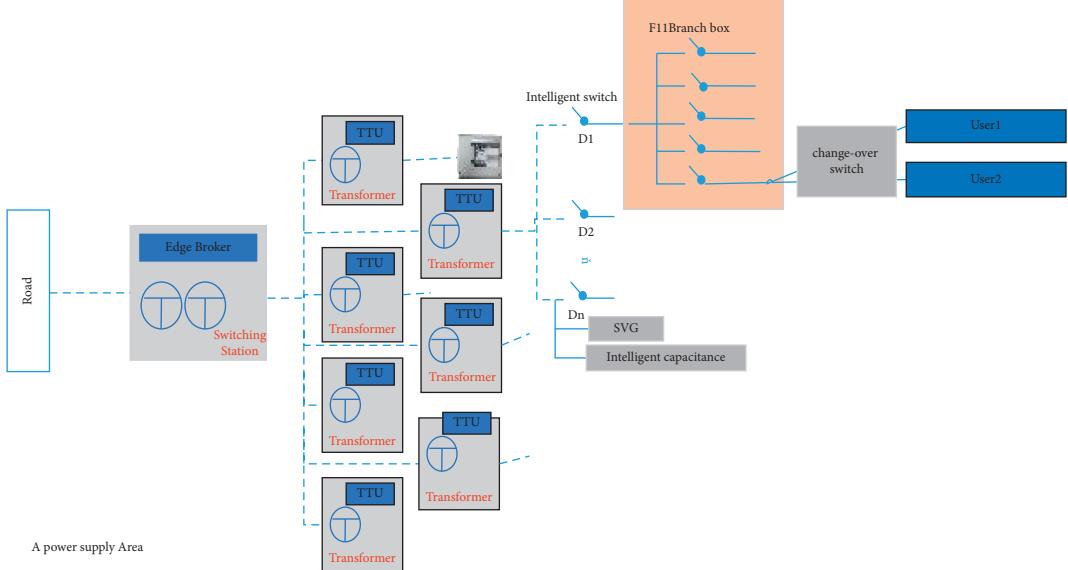


FIGURE 2: Edge computing application of the power supply area.

For example, in [40], a task scheduling algorithm is proposed, named “DSFACO,” which is assigned to decrease the execution time for several tasks on the same scheduling queue. This algorithm improves the running efficiency of virtual machine resources.

In the second-level nodes, the computation-intensive tasks will be usually divided into many subtasks and the edge nodes will complete the tasks by cooperation with each other. Therefore, in this paper, the ant colony algorithm is considered to realize the resource sharing among the edge nodes.

4.2.1. Mathematical Model. For the ant colony algorithm, some parameters are introduced in Table 1. Furthermore, in this section, we introduced the computational model for pheromones, the update model for pheromones, and the state transition probability model of nodes.

(1) *Computational Model for Pheromones.* In this model, the pheromone of resource node can be calculated by its attribute values. Some attributes are considered: CPU utilization, power consumption, memory utilization, external storage utilization, and node security. In this paper, the pheromone of a resource node can be calculated as follows:

$$\text{Ph}_i(t) = \partial_1 * \text{Ph}_i^{\text{cu}}(t) + \partial_2 * \text{Ph}_i^{\text{pc}}(t) + \partial_3 * \text{Ph}_i^{\text{mu}}(t) + \partial_4 * \text{Ph}_i^{\text{su}}(t) + \partial_5 * \text{Ph}_i^{\text{ns}}(t), \quad (1)$$

where $\partial_1, \partial_2, \partial_3, \partial_4$, and ∂_5 represent the important degree of resource attributes on node pheromones. In addition, $\text{Ph}_i^{\text{cu}}(t)$, $\text{Ph}_i^{\text{pc}}(t)$, $\text{Ph}_i^{\text{mu}}(t)$, $\text{Ph}_i^{\text{su}}(t)$, and $\text{Ph}_i^{\text{ns}}(t)$ represent the CPU utilization, power consumption, memory utilization, external storage utilization, and node security grade of node i after normalization. At the beginning, $\tau_i(0) = C$. Furthermore, the power consumption of edge node i can be expressed as follows:

$$\text{Ph}_i^{\text{pc}}(t) = \text{PC}_{\text{no}}(i) + \left(\frac{\sum_{j=1}^n f_{i,j}}{f_i^{\max}} \right)^2 (\text{PC}_{\max}(i) - \text{PC}_{\text{no}}(i)), \quad (2)$$

where $\text{PC}_{\text{no}}(i)$ is the power consumption when there is no task assigned on this edge node. $\text{PC}_{\max}(i)$ is the power consumption when this edge node is full load. $f_{i,j}$ is the processing speed of task i on edge node j . f_i^{\max} is the max value of processing speed.

To resign a unified computing method of node pheromone, the values of attribute are normalized before being integrated. Therefore, the SAW (simple additive weighting) technology is used in this paper [41]. The attributes are divided into two categories: the first kind of attribute values is positively correlated with node pheromones, which means that the value of node pheromones will increase with the value of attribute, such as node security grade; the second kind of attribute values is negatively correlated with node pheromones, which means that the value of node pheromones will decrease with the value of attribute, such as CPU utilization, power consumption, memory utilization, and external storage utilization. The normalization process of the first kind of attribute can be expressed as formula (2), and the normalization process of the second kind of attribute can be expressed as formula (3). Furthermore, $l_x^+(\text{Ph}_i^x)$ and $l_x^-(\text{Ph}_i^x)$ represent the normalized value of x -th attribute. On the contrary, $p_x(\text{Ph}_i^x)$ represents the initial value before

TABLE 1: Parameters.

Name	Introduction
n	Number of computing nodes
m	Number of task
n	Number of resource nodes
Tabu_k	Tabu table of ant k
ΔPh_i	Pheromone increment in node i
ΔPh_i^k	Pheromone increment when ant k assigns task to node i
$P_{ij}^k(t)$	Transfer probability when ant k transfers from node i to node j at time t
d_{ij}	Distance between node i and node j
θ_{ij}	Heuristic factor, the inspired degree from i to j
δ	Pheromone persistence (the residual coefficient), $0 < \delta < 1$
τ_i	Pheromones in node i

normalization. In addition, $\max p_x$ and $\min p_x$ represent the maximum value and the minimum value of x -th attribute.

$$l_x^+(\text{Ph}_i^x) = \begin{cases} \frac{p_x(\text{Ph}_i^x) - \min p_x}{\max p_x - \min p_x}, & \max p_x - \min p_x \neq 0, \\ 1, & \max p_x - \min p_x = 0, \end{cases} \quad (3)$$

$$l_x^-(\text{Ph}_i^x) = \begin{cases} \frac{\max p_x - p_x(\text{Ph}_i^x)}{\max p_x - \min p_x}, & \max p_x - \min p_x \neq 0, \\ 1, & \max p_x - \min p_x = 0. \end{cases} \quad (4)$$

(2) *Update Model for Pheromones.* In the ant colony algorithm, effective nodes are those useful nodes searched by the ant, and the effective nodes are used to perform computational tasks. In this paper, the effective node is defined as a

node whose pheromone is not less than a certain threshold after being assigned tasks. When a node is assigned tasks, its CPU utilization, memory utilization, and external storage utilization will increase. Therefore, to ensure the load balance of each resource node, the node pheromone should be decreased whose resource utilizations are relatively high. In that case, the nodes whose resource utilizations are high will have lower probability to be assigned tasks. Therefore, the pheromone update model can be expressed as follows:

$$\text{Ph}_i(t + n) = (1 - \delta)\text{Ph}_i(t) - \Delta\text{Ph}_i, \quad (5)$$

where δ is the pheromone persistence coefficient, that is, the affect degree of existing pheromone to pheromone update process. In addition, when node i is assigned task, the probability it will be assigned tasks next time can be changed, so ΔPh_i^k is the pheromone variation to node i when it is assigned tasks by ant k , and $\Delta\text{Ph}_i = \sum_{k=1}^m \Delta\text{Ph}_i^k$ is the sum of pheromone affect. The pheromone incremental effect of task i when it is assigned task k can be expressed as follows:

$$\Delta\text{Ph}_i^k = \Delta\text{Ph}_i^{k,\text{cu}}(t) + \Delta\text{Ph}_i^{k,\text{pc}}(t) + \Delta\text{Ph}_i^{k,\text{mu}}(t) + \Delta\text{Ph}_i^{k,\text{su}}(t) + \Delta\text{Ph}_i^{k,\text{ns}}(t). \quad (6)$$

(3) *Probability Model for State Transition.* When selecting the next transition node, ants always tend to choose the neighbor node with the highest transfer probability. The transfer probability calculation formula can be expressed by

$$p_{ij}^k(t) = \begin{cases} \frac{[\text{Ph}_{ij}(t)]^\lambda [\theta_{ij}(t)]^\chi}{\sum_{S \in j_k(i)} [\text{Ph}_{ij}(t)]^\lambda [\theta_{ij}(t)]^\chi}, & \text{if } j \in j_k(i), \\ 0, & \text{else,} \end{cases} \quad (7)$$

wherer λ is the importance degree of pheromones and χ is the importance degree of heuristic factor. In addition, $j_k(i)$ is the node set which can be selected in next step for ant k . Heuristic factor θ_{ij} can be calculated as follows:

$$\theta_{ij}(t) = \frac{1}{d_{ij}}, \quad (8)$$

where d_{ij} means the distance of node i and node j . When ant k selects the next transfer node, the smaller d_{ij} is, the less the communication cost between node i and node j is, and the bigger probability that node j is selected as the next transfer node. In addition, $\theta_{ij}(t)$ is expectation level of ant k moving from node i to node j .

4.2.2. *Algorithm Flow.* In this section, we will introduce the specific flow of the proposed resource management algorithm. The specific flow is shown in Figure 3.

- (1) The node pheromone is initialized.
- (2) The ants are placed, and the timer is started.
- (3) The ant will select the next transfer node according to the state transition probability and determine whether that node is a valid node. If the selected node is a valid node, the node information will be recorded and returned; if the selected node is not the valid

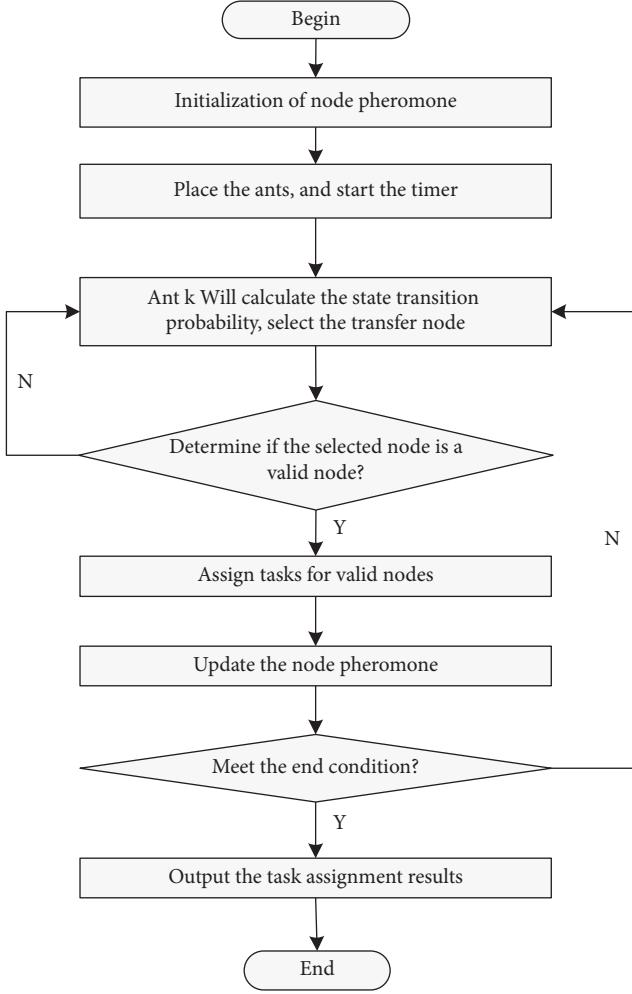


FIGURE 3: The task scheduling algorithm based on the ant colony algorithm.

node, the ant will continue to choose the next transfer node and determine if it is a valid node.

- (4) The tasks are assigned for the valid nodes, and the node pheromone is updated.
- (5) The timer ends.

5. Simulations

In the new resource management scheme based on the idea of edge computing, we propose a task scheduling algorithm based on the ant colony algorithm. In this section, we will simulate the performance of the new algorithm. Without loss of generality, all the data are the average values of ten experiments. In the experiments, $\partial_1, \partial_2, \partial_3, \partial_4$, and ∂_5 are set as 0.25. The simulation parameters are expressed in Table 2, and we set some contrast algorithms to verify the effectiveness of the new algorithm. In this section, two comparing algorithms are set. One is the modified round robin algorithm, and the other is the generalized priority algorithm.

- (i) *Modified Round Robin Algorithm (RRA, for Short).* This algorithm is proposed in [42], which can randomly assign free resources for task execution

TABLE 2: Parameters introduction.

Parameters	Values
Number of edge nodes	[20, 500]
Node security grade of node	[0.5–1)
Number of tasks	30–350
Processing speed	100–1000 MIPS
Length of the task	1000–10000 MI

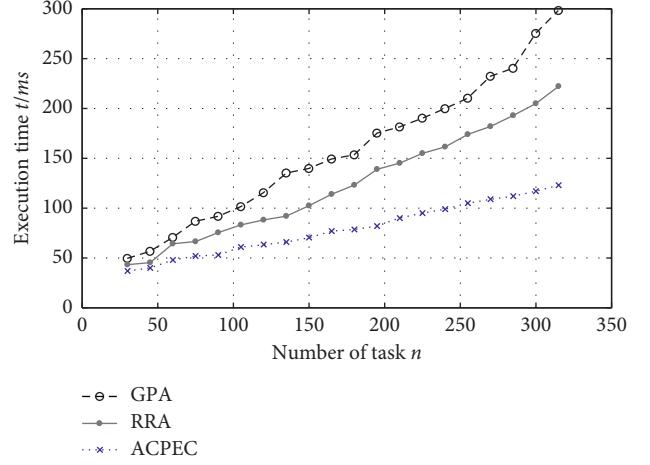


FIGURE 4: The comparison of task completion time.

- (ii) *Generalized Priority Algorithm (GPA, for Short).* This algorithm is proposed in [43], which is an efficient optimal algorithm of task scheduling in cloud computing environment
- (iii) *Ant Colony Algorithm in Power Edge Computing (ACPEC, for Short).* This algorithm is the resource scheduling algorithm proposed in this paper

In this section, we simulate the performance of RRA, GPA, and ACPEC on the task execution time with different number of tasks. As shown in Figure 4, we can find that the task execution times of ACPEC are always shorter than those of GPA and RRA. With the increase in task scale, the advantages become more and more obvious. When the number of task reaches 320, the advantage of ACPEC is very obvious, as shown in Figure 4.

The task wait delay is a key factor affecting the real-time response of the system. As shown in Figure 5, the task waiting time always increases with the increase in task scale. Compared with GPA and RRA, the ACPEC algorithm is more complex, so the required time for task assigning is always longer slightly. However, compared with the total amount of time for task execution, a small increase in task wait time is acceptable for users.

The performances on power consumption cost are shown in Figure 6. For the ACPEC algorithm, the tasks can be assigned to different resource nodes according to the purpose of reducing power consumption. Therefore, when the task scale is the same, the power consumption with ACPEC is always less than the power consumption with FSFC and RRA. And as the task scale increases, the advantage tends to be obvious.

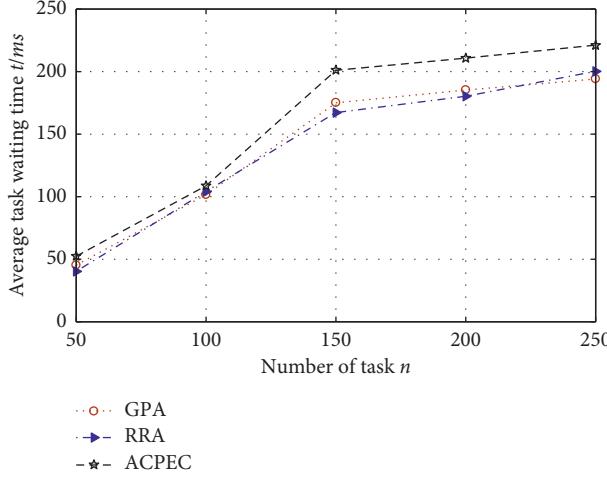


FIGURE 5: The comparison of average waiting delays.

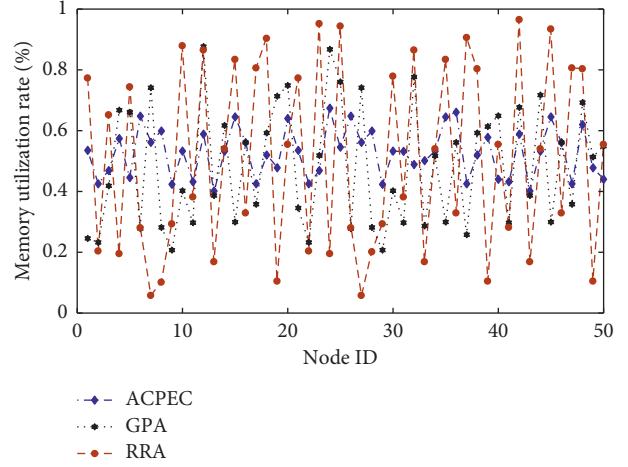


FIGURE 8: The comparison of memory utilization.

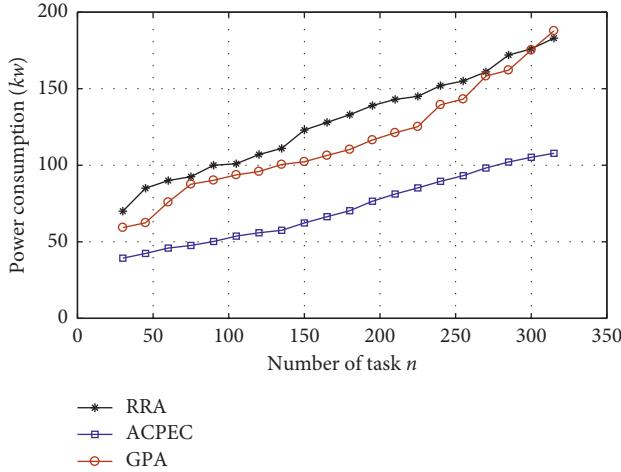


FIGURE 6: The comparison of power cost.

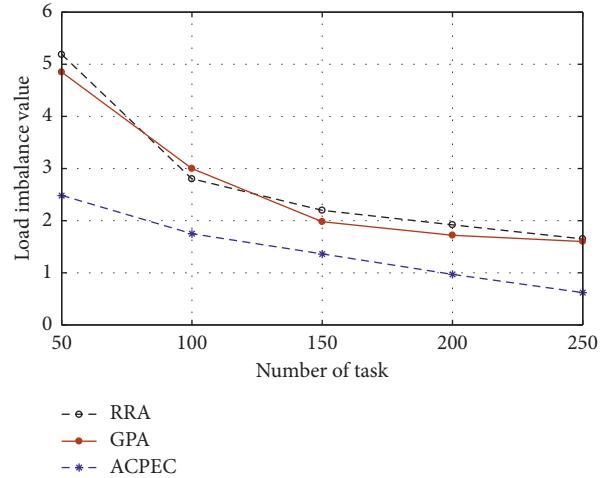


FIGURE 9: The comparison of unbalanced load value.

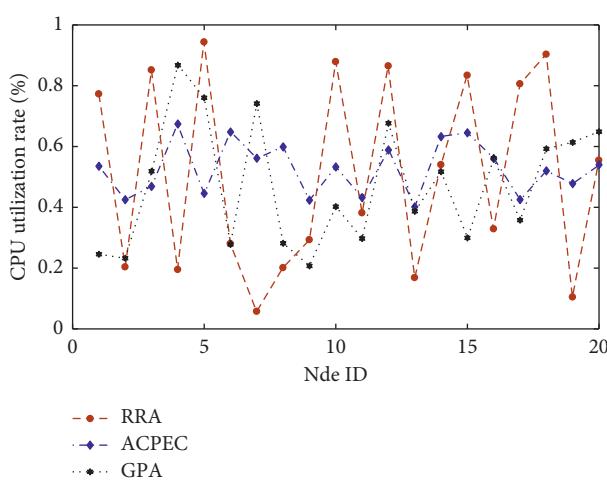


FIGURE 7: The comparison of CPU utilization.

When the number of tasks is 100, the CPU utilization with 20 edge nodes and the memory utilization with 50 nodes are simulated as shown in Figures 7 and 8. We can find that the CPU utilization and memory utilization of some nodes fluctuate greatly when using GPA and RRA. Furthermore, some nodes are under low load state when using GPA and RRA. When using the ACPEC algorithm, the CPU utilization and memory utilization among different nodes have become average. Compared with RRA, the average CPU utilization increased from 38.8% to 69.9%, and the average memory utilization increased from 38% to 68%.

In this section, we will evaluate the performance of these task scheduling algorithms on load imbalance. In this section, M_{DI} (degree of imbalance) represents the level of load imbalance, which can be calculated as follows:

$$M_{DI} = \frac{M_{\max} - M_{\min}}{M_{\text{avg}}}, \quad (9)$$

where M_{\max} , M_{\min} , and M_{avg} represent the longest time, the shortest time, and the average time for task performing. The load imbalance levels of different algorithms are shown in Figure 9. We can find that the load imbalance levels decrease gradually with the increase in task scale. This is because the assigning of tasks is gradually balanced with the increase in task scale. Compared with GPA and RRA, the values of M_{DI} with ACPEC are lower. The performance of ACPEC on load balance is better than that of GPA and RRA as shown in Figure 9.

6. Conclusion

To ensure the efficient and stable operation of the power cloud computing system and make full use of the advantages of edge nodes, in this paper, we propose a new resource management scheme based on edge computing. In this scheme, the computing tasks can be executed in local edge nodes instead of uploading to the cloud. In this new scheme, although the resources of single edge node are limited, the resource among edge nodes can cooperate with each other to complete the computation-intensive task. To realize the optimal allocation between edge nodes and tasks, in this paper, a task allocation algorithm based on the ant colony algorithm is proposed. This algorithm has good performance in task execution time, waiting delay, CPU utilization, memory utilization, and load balancing. The communication overhead is greatly affected by network environment; therefore, the communication overhead of data transmission and the difference of system power consumption should be considered in resource management issue, which may become our future work.

Data Availability

The simulation data used to support the findings of this study are included within the supplementary information file.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Key R&D Program of China (2020YFB2104503).

References

- [1] A. Fox, R. Griffith, A. Joseph, and Armando, “Above the clouds: a berkeley view of cloud computing,” *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, vol. 28, no. 13, pp. 3–5, 2009.
- [2] R. Harms and M. Yamartino, *The Economics of the Cloud*, Microsoft whitepaper, Microsoft Corporation, Redmond, Washington, 2010.
- [3] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, “A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing,” in *Proceedings of the 2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, San Diego, CA, USA, December 2015.
- [4] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, “Joint computation offloading and interference management in wireless cellular networks with mobile edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7432–7445, 2017.
- [5] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.
- [6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: the communication perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [7] S. E. Mahmoodi, R. N. Uma, and K. P. Subbalakshmi, “Optimal joint scheduling and cloud offloading for mobile applications,” *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 301–313, 2019, to be published.
- [8] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, “Heterogeneity in mobile cloud computing: taxonomy and open challenges,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 369–392, 2014.
- [9] N. Zhang, X. Yang, M. Zhang, and Y. Sun, “Crowd-Funding: a new resource cooperation mode for mobile cloud computing,” *PloS One*, vol. 11, no. 12, pp. e0167657–989, 2016.
- [10] Y. Sun and N. Zhang, “A resource-sharing model based on a repeated game in fog computing,” *Saudi Journal of Biological Sciences*, vol. 24, no. 3, pp. 687–694, 2017.
- [11] G. Von Laszewski, L. Wang, A. J. Younge, and X. He, “Power-aware scheduling of virtual machines in dvfs-enabled clusters,” in *Proceedings of the IEEE International Conference on Cluster Computing 2009*, pp. 1–10, New Orleans, LA, USA, September 2009.
- [12] J. L. Berral, I. Goiri, R. Nou, and F. Julià, “Towards energy-aware scheduling in data centers using machine learning,” in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, pp. 215–224, ACM, New York, NY, USA, April 2010.
- [13] R. Jayarani, S. Sadhasivam, and N. Nagaveni, “Design and implementation of an efficient two-level scheduler for cloud computing environment,” in *Proceedings of the 2009 International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 884–886, IEEE, NW Washington, DC, October 2009.
- [14] M. Shojafar, N. Cordeschi, and E. Baccarelli, “Energy-efficient adaptive resource management for real-time vehicular cloud services,” *IEEE Transactions on Cloud computing*, vol. 7, no. 1, pp. 196–209, 2016.
- [15] M. Shojafar, N. Cordeschi, J. H. Abawajy, and E. Baccarelli, “Adaptive energy-efficient Qos-aware scheduling algorithm for TCP/IP mobile cloud,” in *Proceedings of the 2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, IEEE, San Diego, CA, USA, December 2015.
- [16] G. Portaluri, S. Giordano, D. Kliazovich, and B. Dorronsoro, “A power efficient genetic algorithm for resource allocation in cloud computing data centers,” in *Proceedings of the 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pp. 58–63, IEEE, Luxembourg, October 2014.
- [17] N. Zhang, X. Yang, M. Zhang, Y. Sun, and K. Long, “A genetic algorithm-based task scheduling for cloud resource crowd-funding model,” *International Journal of Communication Systems*, vol. 31, no. 1, Article ID e3394, 2018.

- [18] X. Liu, X. B. Zhai, W. Lu, and C. Wu, "QoS-guarantee resource allocation for multibeam satellite industrial internet of things with NOMA," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2052–2061, 2021.
- [19] 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, 2020.
- [20] L. Gkatzikis and I. Koutsopoulos, "Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 24–32, 2013.
- [21] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.
- [22] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proceedings of the IEEE ISIT*, pp. 1451–1455, Barcelona, Spain, July 2016.
- [23] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.
- [24] R. Deng, R. Lu, C. Lai, and T. H. Luan, "Towards power consumptiondelay tradeoff by workload allocation in cloud-fog computing," in *Proceedings of the 2015 IEEE International Conference on Communications (ICC)*, pp. 3909–3914, London, UK, June 2015.
- [25] E. Gelenbe, R. Lent, and M. Douratsos, "Choosing a local or remote cloud," in *Proceedings of the Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on*, pp. 25–30, IEEE, London, UK, December 2012.
- [26] Y. Li and W. Wang, "The unheralded power of cloudlet computing in the vicinity of mobile devices," in *Proceedings of the IEEE Global Commun. Conf.*, pp. 4994–4999, London, UK, December 2013.
- [27] K. Suto, K. Miyanabe, H. Nishiyama, N. Kato, H. Ujikawa, and K.-I. Suzuki, "QoE-Guaranteed and power-efficient network operation for cloud radio access network with power over fiber," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 4, pp. 127–136, 2015.
- [28] G. von Zengen, F. Bsching, W. B. Pttner, and L. Wolf, "Transmission power control for interference minimization in WSNs," in *Proceedings of the Int. Wireless Commun. Mobile Comput. Conf.*, pp. 74–79, New York, NY, USA, August 2014.
- [29] T. Aota and K. Higuchi, "A simple downlink transmission power control method for worst user throughput maximization in heterogeneous networks," in *Proceedings of the 7th Int. Conf. Signal Process. Commun. Syst.*, pp. 1–6, China, December 2013.
- [30] M. Peng, K. Zhang, J. Jiang, J. Wang, and W. Wang, "Energy-efficient resource assignment and power allocation in heterogeneous cloud radio access networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5275–5287, 2015.
- [31] J. Armstrong, "OFDM for optical communications," *Journal of Lightwave Technology*, vol. 27, no. 3, pp. 189–204, 2009.
- [32] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, 2015.
- [33] J. Cheng, Y. Shi, B. Bai, and W. Chen, "Computation offloading in cloud-ran based mobile cloud computing system," in *Proceedings of the 2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kuala Lumpur, Malaysia, May 2016.
- [34] C. You, K. Huang, H. Chae, and B. H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 99, p. 1, 2016.
- [35] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Abu Dhabi, UAE, December 2016.
- [36] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [37] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the edge: a scalable IoT architecture based on transparent computing," *IEEE Network*, vol. 31, no. 5, pp. 96–105, 2017.
- [38] H. Tan, Z. Han, X.-Y. Li, and F. C. M. Lau, "Online job dispatching and scheduling in edge-clouds," in *Proceedings of the IEEE INFOCOM IEEE Conf. Comput. Commun.*, pp. 1–9, Toronto, ON, Canada, May 2017.
- [39] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile edge computing: progress and challenges," in *Proceedings of the 2016 4th IEEE international conference on mobile cloud computing, services, and engineering (MobileCloud)*, pp. 83–84, IEEE, Oxford, UK, April 2016.
- [40] Q. Y. Guo and F. D. Zhu, "Cloud computing resource scheduling algorithm based on ant colony algorithm and leap frog algorithm," *Bulletin of Science and Technology*, vol. 33, no. 5, pp. 167–170, 2017.
- [41] C. Wang and Z. Li, "Parametric analysis for adaptive computation offloading," *ACM SIGPLAN Notices*, vol. 39, no. 6, pp. 119–130, 2004.
- [42] P. Pradhan, P. K. Behera, and B. N. B. Ray, "Modified round robin algorithm for resource allocation in cloud computing," *Procedia Computer Science*, vol. 85, pp. 878–890, 2016.
- [43] A. Agarwal and S. Jain, "Efficient optimal algorithm of task scheduling in cloud computing environment," *International Journal of Computer Trends and Technology*, vol. 9, no. 7, 2014.