Journal of Computer Technology and Software

ISSN: 2998-2383

Vol. 5, No. 01, 2026

Intelligent Task Scheduling Framework for Cloud Platforms Based on Improved Particle Swarm Optimization

Dorothea Bradeen[1], Danyang Tong[1]
[1]New York University, New York, USA
*Corresponding author: Danyang Tong; dyt4456@nyu.edu

Abstract: This paper analyzes the characteristics and goals of cloud platform task schedul- ing. Starting from task scheduling algorithms, it proposes an artificial intelligence method based on an improved particle swarm optimization (PSO) algorithm for the electric power scheduling automation system, and develops a cloud computing operation model. Based on this algorithm and the physical model, the control oper- ation takes into account the QoS requirements and the environmental load balance of the cloud platform residents, which can effectively improve the task scheduling efficiency of the cloud platform in the power scheduling automation system. Tak- ing the power automation cloud platform as the research object, the architecture is studied and the PSO algorithm is modified and combined with the structure of cloud resource scheduling model. A three-level data node system is established, and a cloud platform scheduling model based on the improved PSO algorithm is proposed to improve cloud resource allocation efficiency and service quality, solving the task scheduling problem of power scheduling automation systems.

Keywords: Cloud platform, task scheduling, improved particle swarm algorithm, scheduling optimization

1. Introduction

Based on the smart grid scheduling system interconnected via the internal power grid network, the cloud computing platform possesses strong capabilities in data collection, analysis, and computing power [1][2]. Through sustained and efficient utilization of data centers, the hardware and software resources of the electronic control system operation platform are stable and efficient, and the center effectively manages the communication and execution of the entire power grid.

The scheduling automation system of the power system is a cloud platform that sup- ports users in obtaining computing resources through network access to a shared pool. This resource pool manages and optimizes resources, enabling rapid configuration and delivery within a commercial model [3][4][5].

Cloud platforms leverage Internet technology and provide services to a large number of users using a scalable and flexible IT infrastructure. Cloud users do not purchase services based on what they are told they need, but rather according to their actual individual demands. The resources and user needs on the cloud platform are both dynamic, and therefore, pre-allocation and distribution of resources can no longer meet the service requirements of power users.

In the power automation cloud platform, task scheduling must meet user demands, with the goal of achieving optimal scheduling schemes and maximizing overall efficiency. This requires tasks to be scheduled quickly, in real-time, and efficiently, allowing idle resources in the cloud to be fully and effectively utilized to meet user demands [6][7][8][9].

However, traditional task allocation methods fail to satisfy the performance demands of cloud platform users related to power automation, which may make it difficult to ensure accuracy and structural balance. Therefore, it is necessary to use efficient and scientific techniques to perform power flow control on the cloud platform, and to apply dynamic resource allocation algorithms and techniques to improve both user and platform efficiency.

This paper proposes a task scheduling model for power scheduling automation cloud platforms based on an improved particle swarm optimization (PSO) algorithm. It aims to enhance the efficiency of cloud computing resource allocation, improve service quality, and solve task scheduling problems in power scheduling automation systems.

2. The Connotation and Objectives of Cloud Plat- form Task Scheduling

Task scheduling has long been a fundamental issue in distributed and multiprocessor systems. With the continuous expansion of cloud platform scale, the heterogeneity of resources, the dynamic nature of task variations, and the diversity of service requirements have become increasingly prominent. Traditional rule-based or greedy scheduling algorithms are no longer sufficient to meet the demands for adaptivity and efficiency in modern cloud environments. In response to these challenges, this paper integrates recent advances in serverless scheduling, anomaly detection, and federated resource coordination to propose an intelligent scheduling model based on an improved Particle Swarm Optimization (PSO) algorithm, designed for efficient task management on power automation cloud platforms.

Within this framework, task scheduling is modeled as a dynamic mapping process between resources and tasks. By leveraging virtualization technology, underlying heterogeneous computing and storage resources are abstracted into a unified resource pool, while user requests are modeled as multi-dimensional tasks according to their priority and Quality of Service (QoS) requirements, thus enabling highly flexible task-resource adaptation.

Inspired by the predictive autoscaling mechanism proposed in relevant literature [10], this work designs a task prediction and classification module. By analyzing historical task data and resource usage trends, this module can forecast the types and loads of upcoming tasks, enabling the scheduler to dynamically pre-allocate resources before tasks actually arrive. Such proactive scheduling significantly reduces system response latency and improves overall throughput, especially in power automation scenarios characterized by high concurrency and frequent fluctuations in resource demand.

To enhance the intelligence and robustness of the scheduling system, this study incorporates structure-aware and semantically enhanced graph modeling methods [11], embedding task structural attributes and semantic features into the PSO fitness function. By integrating task dependencies and behavioral patterns, the scheduling system can detect abnormal task allocation in real time, thereby improving its adaptability and anomaly detection capabilities in complex and dynamic environments.

For coordination mechanisms, a dual-strategy is adopted: on the one hand, node abstraction and hierarchical structures are used to ensure the transparency and controllability of the scheduling process; on the other hand, a three-level scheduling architecture is constructed based on a hierarchical reputation evaluation mechanism [12], which supports tiered scheduling and dynamic load balancing for different types and priorities of tasks, thus enhancing resource utilization and service fairness in multi-tenant environments.

To address issues of resource fragmentation and low utilization in heterogeneous clusters, this study introduces a fragmentation-aware task placement strategy [13], which intelligently allocates tasks to the optimal resource partitions according to task granularity and resource pool fragmentation characteristics. This effectively improves the utilization of computational and storage resources in the cloud platform while reducing resource waste and scheduling conflicts.

In multi-tenant and complex environments, to further enhance system stability and security, this work integrates federated modeling and personalized anomaly detection mechanisms [14]. Through collaborative analysis of distributed models and local monitoring data, the scheduler can dynamically detect and correct performance drift and abnormal behavior during scheduling, strengthening the system's ability to handle resource anomalies and task imbalances in multi-tenant scenarios.

Additionally, to meet the needs of generalized scheduling across clouds and multiple business types, this study adopts transfer learning and self-supervised feature encoding methods [15], designing a scheduling model capable of adaptive generalization across multiple task types and execution contexts. This mechanism significantly enhances the system's coordination and adaptability in different application domains and workloads.

In summary, the improved PSO-based intelligent scheduling framework proposed in this paper organically integrates advanced methods such as task prediction, structural modeling, multi-level coordination, anomaly detection, and migration optimization, achieving comprehensive improvements in task scheduling efficiency, resource utilization, and service robustness for power automation cloud platforms. The overall system architecture is shown in Figure 1:

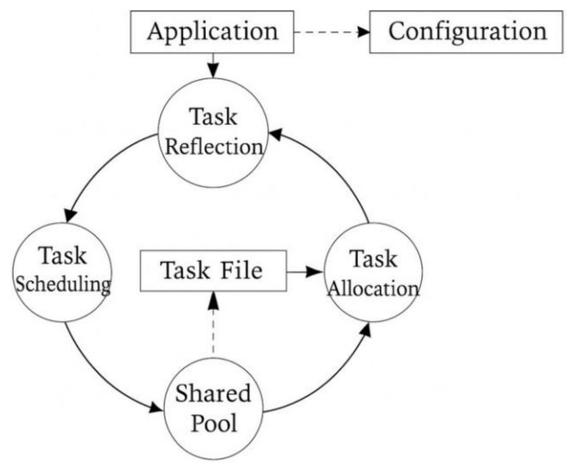

Figure 1. Cloud platform task scheduling model

This typically refers to the usage of cloud platforms, including:

a. Parsing configuration files of application programs to analyze scheduling schemes.

b. Selecting from resource pools and configuration files.

c. Predicting execution and required resources for tasks.

d. Allocating resources to end users based on estimated availability and usage.

The power dispatching problem is an NP problem, mainly because resources and tasks are dynamic, making scheduling difficult. Therefore, solving the power scheduling problem requires finding a relatively optimal solution using scheduling algorithms, with QoS as the evaluation criterion.

Cloud tasks mainly involve the following aspects:

a. Minimum latency. This refers to the shortest time to complete a task, from the submission of the first task to the completion of the last one. The length of this interval directly affects user experience. The shorter the time, the lower the cost for users.

b. Service quality. This refers to the system capability of the cloud platform to provide service guarantees. QoS levels directly affect whether power users will con- tinue using the cloud platform. The evolution of the platform depends largely on how well it manages tasks and virtual resource allocation. In diverse applications, effective resource management is critical to service quality.

c. Load balancing. This is a key metric in cloud systems, referring to the balanced migration of resources across cloud servers to avoid overloading any single node. To achieve this, a good strategy is needed to assign tasks properly and fully utilize virtual resources.

d. Economic efficiency. Cost reduction is a key concern for both providers and users. Users seek lower operational costs, and service providers aim to reduce costs while ensuring overall profitability.

3. Cloud Platform Architecture of Power Dispatch Automation System

The intelligent communication system and automation system of the network adopt cloud technologies to integrate data and information available from distributed data services into structured systems. This enables the network to achieve high reliability, high efficiency, and high accuracy.

The cloud platform includes distributed distribution systems and master host alloc- ation platforms. These components form a virtual layer, and through distributed data services, data exchange, transmission, and integration are achieved. Meanwhile, unified management and deployment of physical hardware at the bottom layer ensures effective and stable access.

To facilitate real-time monitoring, editing, and supervision of system components used and operated by the public, the platform provides unified management and supervision technologies. The network's regional deployment consists of three functional modules. The control center has been integrated into the network platform through master host control cards. Different functions of the traffic control center can be mutually supported, enhancing system safety and reliability.

3.1 Distributed Data Service Backbone

The power dispatch platform adopts a large-scale and flexible network architecture, which is a decentralized network. In geographically distributed areas, multiple active transport platforms allow administrators to integrate distributed servers, allocate the least business resources, and achieve low-latency data transmission over long distances, ensuring high availability.

It also supports distributed parameter expansion, dynamic application location, and automatic recovery from system faults. Efficient remote control and strong logging capab- ilities allow operators to adjust parameters and monitor network status, providing reliable support for power system search and management.

3.2 Data Storage System

The distributed system is a decentralized software system that integrates data storage devices with network nodes to process and transmit data in real time. For long-term business development, massive data storage and management should utilize distributed systems without requiring new hardware or changes to physical device types.

Spatial data storage is encapsulated within the file system, and any node appearing in a local file system accesses dynamic file system data. This also supports storage and retrieval between systems and file data. By establishing auxiliary nodes, issues in data access are resolved and data processing is improved.

3.3 Dynamic Load Balancing and Resource Allocation System

To efficiently manage the driver system, the platform integrates load capacity balan- cing and dynamic resource allocation mechanisms. Starting from the smallest computing nodes, the animation computer system transmits computing speeds and signal execution processes to the action center. Basic nodes and cross-system servers integrate pressure into a dynamic system.

The central control system includes four main functions: monitoring, authentication, strategic management, and decision centers. These provide a unified access platform for the entire power grid. The authentication and strategy center is responsible for authentic-ation and management of the entire power plant network, while the decision center makes critical judgments on the power and traffic management systems.

With assistance from systems cooperating with specific nodes, the platform detects and eliminates faults. Tasks from faulty nodes can be reassigned to less burdened nodes to achieve load shifting.

3.4 Integrated Computing Engine

The integrated computing engine combines various computing resources to solve massive data problems in big data scenarios, improving real-time processing and analysis. This engine reduces dependency on external reports by allowing direct access to distributed node data, as shown in Figure 2.

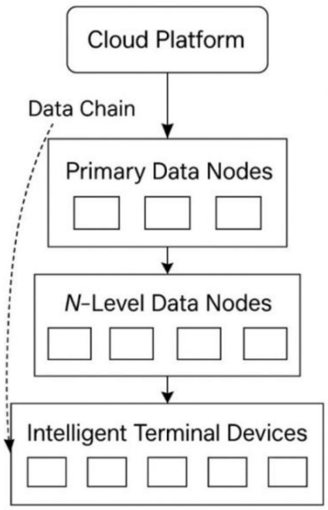

Figure 2. Integrated computing engine

Data analysis and processing are conducted across different functional nodes until final data collection is completed.

4. Cloud Platform Task Scheduling Model of Power Dispatch Automation System Based on Artificial Intelligence Algorithms

The task scheduling strategy of the cloud platform based on artificial intelligence can be described as follows: the cloud user sends a request to the cloud service data center, and the computing data center allocates virtual computing resources based on system strategy requirements. Then, using the algorithm proposed in this paper, the best available phys- ical resources are allocated, and users receive appropriate treatment accordingly.

A component of the model is based on user needs and specific contractual virtual resource invocation strategies. Another component involves physical mapping, where improved optimization algorithms are used to adjust the ratio between virtual and physical resource allocation appropriately.

An improved Particle Swarm Optimization (PSO) algorithm is used as the intelligent algorithm to build the platform operation model. This enhanced PSO model considers the connection between network nodes in the cloud computing model and views physical computing resources as individual PSO particles. The entire physical resource pool is considered as a single particle to optimize the best shared platform.

Each computing PSO particle represents the computing data of a unit resource, and dynamic particles are optimized accordingly. The communication capabilities of devices correspond to particle speed, and separated physical resources are assigned to the best matching particle or subset.

Task assignment in the cloud is calculated based on the particle optimization prin- ciple. Task constraints and assignments are aligned with the optimization strategy of the particles. In the cloud environment, tasks are assigned to specific resources in a loop until optimal resource allocation is achieved using the improved algorithm.

In this context, the QoS function and a utility function are combined to evaluate the optimization value, while penalization functions are used as auxiliary objectives.

4.1 Model Construction

To apply the PSO algorithm proposed in this paper to cloud platform task scheduling, we first build a mathematical model for scheduling. Based on the specifics of cloud- based power dispatching, an objective function is established to define basic constraints and parameters of the system, which form the basis for manipulating cloud computing particles.

In normal operations, a balance between power user demand and virtual power re- sources needs to be maintained. Cloud users expect low-cost and reliable service, while providers aim to offer stable services under cost-effective and low-power conditions. There- fore, the amount of virtual resource usage and service quality generated in the online environment together represent this balance.

Evidence shows that giving users more power and choice enables access to newer and more reliable services. With increased user demand, the complexity of the system grows, making stability harder to maintain. Thus, an optimal control model is developed to balance user needs and expectations:

$$F = Q(p) + G(p)$$

where $Q(p)$ represents the relationship between the quantity of power resources and power users, and $G(p)$ represents the relationship between power resources and load bal- ancing. The goal is to find the minimum value of F, where the values of $Q(p)$ and $G(p)$ reach a balance, indicating optimal resource usage. This corresponds to the intersection point in the figure.

The mathematical model is illustrated in Figure 3.

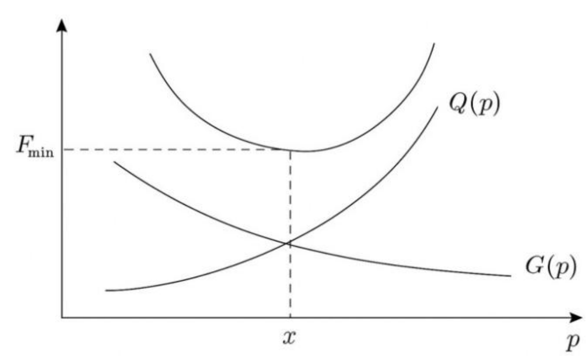

Figure 3. Task scheduling model

4.2 Objective Function

Under the condition of load balancing, the optimal task scheduling problem is solved using artificial intelligence algorithms. This is a power dispatching process under user service demand constraints. The optimal power scheduling corresponds to power resource data, which serves as the solution to the objective function. Therefore, the problem is transformed into one of finding the optimal task scheduling.

When selecting between physical and virtual machines, the load limit of virtual ma- chines must be considered from the outset to ensure power load balancing. The objective function is therefore constructed as:

$$F = \min\left(h_1 \times \frac{f_{\mathrm{ir}}(U) + \mathrm{De}(U)}{t_{ip}} + h_2 \times \frac{t_{oc}(U)}{co_p} + h_3 \times \frac{tr_p}{\mathrm{Tr}(U)}\right)$$

where t_{ip}, co_p, and tr_p represent the expected power service time, cost, and reliability requirements from the cloud user respectively.

4.3 Constraints

Before scheduling tasks, the suitability of computing tasks is calculated. Some tasks are filtered based on suitability, and the remaining tasks are optimized and allocated under constraints. The constraint set of the objective function is:

$$\begin{aligned} \text{s.t.}\quad & \mathrm{De}(VM_i) \leq \mathrm{DL} \\ & \mathrm{Ti}_c(VM_i) \leq \mathrm{TL} \\ & \mathrm{Ba}(VM_i) \geq \mathrm{BL} \\ & \mathrm{Tr}(VM_i) \geq \mathrm{TrL} \\ & \mathrm{fit}(VM_i) \geq \varepsilon \\ & f_{\mathrm{ir}}(U) + \mathrm{De}(U) \leq t_{ip} \\ & t_{oc}(U) \leq co_p \\ & \mathrm{Tr}(U) \geq tr_p \end{aligned}$$

where -Ti$_c$($V\ M_i$): the time cost of processing the task on virtual machine $V\ M_i$, upper bound TL. - De($V\ M_i$): the task delay on $V\ M_i$, upper bound DL. - Ba($V\ M_i$): allocated network bandwidth to $V\ M_i$, lower bound BL. - Tr($V\ M_i$): trust value of $V\ M_i$, lower bound TrL. - fit($V\ M_i$): task suitability for $V\ M_i$, ε is the minimum required suitability.

4.4 Parameter Settings

According to the improved Particle Swarm Optimization (PSO) algorithm applied to the cloud platform task scheduling model, the required parameter settings are shown in Table 1.

Table 1: Parameter settings of the improved PSO algorithm

Parameter	Range / Value	Description
η	(1.5, 2.5)	Inertia coefficient
α	(0, 4)	Individual learning factor
β	(0, 4)	Group learning factor
λ, μ	[0, 1]	Free coefficient
Vmax	100	Maximum particle velocity
δ	2	Maximum constraint function value
Imax	1000	Maximum number of iterations
C1	1	Prediction speed constant
C2	[0.85, 0.95]	Inertia coefficient weight
e	[0.7, 0.8]	Upper limit of physical load ratio

4.5 Processing Procedure

Based on the objective function and task constraints of the power scheduling model, the improved PSO algorithm is used to optimize particles in the cloud platform. The process is as follows:

a. The objective function calculates user service demand, clarifying the goal for mean- ingful operations.

b. According to the problem scale, determine the algorithm parameters and initialize particle positions and velocities to form an optimal starting swarm.

c. Compute initial fitness values of particles based on the fitness function.

d. Update particle information using the evolution rules, i.e., each particle updates its personal best (Pbest) and global best (Gbest) values. Increment the iteration counter.

e. Determine if early convergence occurs. If yes, process early convergence and return to step d; otherwise, proceed.

f. Check if the maximum iteration number is reached. If yes, proceed to step i; other- wise, continue.

g. Check for possible convergence. If so, go to the next step; else return to d.

h. Verify each particle's information and compare with Gbest.

i. From the value of Gbest, map to the corresponding VIRTUAL machine.

j. Execute the task on the selected VM, register task-to-resource mapping.

k. The VM executes the pre-scheduled task, completes execution, and releases re- sources.

l. End and output results.

5. Conclusion

This paper analyzed the characteristics and goals of task scheduling on cloud platforms. Starting from task scheduling algorithms, it proposed an artificial intelligence method based on an improved Particle Swarm Optimization (PSO) algorithm for power dispatch automation systems, and developed a cloud computing operation model.

Based on this algorithm and the physical operation model, the runtime control con- siders QoS requirements and environmental load balancing for cloud platform users. This effectively improves the efficiency of cloud platform task scheduling in the power dispatch automation system.

References

[1] Munshi, A. A. and Yasser, A. R. M., "Big data framework for analytics in smart grids", Electric Power Systems Research, vol. 151, pp. 369-380, 2017.

[2] Ntalampiras, S., "Fault diagnosis for smart grids in pragmatic conditions", IEEE Transactions on Smart Grid, vol. 9, no. 3, pp. 1964-1971, 2016.

[3] Buyya, R., Beloglazov, A. and Abawajy, J., "Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges", arXiv preprint arXiv:1006.0308, 2010.

[4] Khazaei, H., Misic, J. and Misic, V. B., "Performance analysis of cloud computing centers using M/G/m/m+R queuing systems", IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 5, pp. 936-943, 2011.

[5] Rodriguez, M. A. and Buyya, R., "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds", IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 222-235, 2014.

[6] Ghemawat, S., Gobioff, H. and Leung, S. T., "The Google file system", Proceedings of the 19th ACM Symposium on Operating Systems Principles, pp. 29-43, 2003.

[7] Chen, W. and Deelman, E., "WorkflowSim: A toolkit for simulating scientific workflows in distributed environments", Proceedings of the 2012 IEEE 8th International Conference on E-Science, pp. 1-8, 2012.

[8] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... and Zaharia, M., "A view of cloud computing", Communications of the ACM, vol. 53, no. 4, pp. 50-58, 2010.

[9] Y. K. Liu, X. Xu, and L. Zhang, “An extensible model for multitask-oriented service composition and scheduling in cloud manufacturing,” Journal of Computing and Information Science in Engineering, vol. 16, no. 4, pp. 11-19, 2016.

[10] Chen, B., "FlashServe: Cost-Efficient Serverless Inference Scheduling for Large Language Models via Tiered Memory Management and Predictive Autoscaling", arXiv preprint arXiv:2512.20218, 2025.

[11] Lyu, N., Jiang, J., Chang, L., Shao, C., Chen, F. and Zhang, C., "Improving Pattern Recognition of Scheduling Anomalies through Structure-Aware and Semantically-Enhanced Graphs", arXiv preprint arXiv:2512.18673, 2025.

[12] Yang, J., Chen, J., Huang, Z., Xu, C., Zhang, C. and Li, S., "Cost-TrustFL: Cost-Aware Hierarchical Federated Learning with Lightweight Reputation Evaluation across Multi-Cloud", arXiv preprint arXiv:2512.20218, 2025.

[13] Ni, Y., Yang, X., Tang, Y., Qiu, Z., Wang, C. and Yuan, T., "Predictive-LoRA: A Proactive and Fragmentation-Aware Serverless Inference System for LLMs", arXiv preprint arXiv:2512.20210, 2025.

[14] Wang, Y., Liu, H., Long, N. and Yao, G., "Federated anomaly detection for multi-tenant cloud platforms with personalized modeling", Proceedings of the 2025 5th International Conference on Intelligent Communications and Computing (ICICC), pp. 555-559, 2025.

[15] Zhou, Y., "Self-supervised transfer learning with shared encoders for cross-domain cloud optimization", Proceedings of the 2025 5th International Conference on Electronic Information Engineering and Computer Science (EIECS), pp. 1435-1439, 2025.