

Retraction

Retracted: Optimization and Simulation of a Reasonable Scheduling Model under Multiple Tasks in Company Management

Complexity

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

Copyright © 2024 Complexity. 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.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] Z. Wang, "Optimization and Simulation of a Reasonable Scheduling Model under Multiple Tasks in Company Management," *Complexity*, vol. 2020, Article ID 1204815, 10 pages, 2020.

Research Article

Optimization and Simulation of a Reasonable Scheduling Model under Multiple Tasks in Company Management

Zhuo Wang ^{1,2}

¹School of Economics and Management, Yanshan University, Qinhuangdao 066000, Hebei, China

²School of Management, Jilin Normal University, Siping, Jilin 136000, China

Correspondence should be addressed to Zhuo Wang; wangzhuo@jlnu.edu.cn

Received 20 July 2020; Revised 29 August 2020; Accepted 12 September 2020; Published 25 September 2020

Academic Editor: Zhihan Lv

Copyright © 2020 Zhuo Wang. 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 rapid development of market economy, the task scheduling model has become the core problem in the field of corporate management. In order to solve the problem that the stability of the model will decline due to the interference of human factors in the process of multitask scheduling in the traditional algorithm, a reasonable scheduling model based on the priority principle is proposed in this paper. This paper expounds the principle of multitask scheduling in company management, constructs a multitask scheduling network model based on priority, performs virtual scheduling on the model, searches the optimal solution in the solution space of the scheduling problem, and obtains the reasonable multitask scheduling method in the company management. Through the analysis of relevant simulation experiments, it can be concluded that the task scheduling algorithm proposed in this paper can not only allocate the corresponding resources for high-priority tasks effectively, but also can save the cost, so as to show better execution efficiency.

1. Introduction

With the rapid development of market economy, the competition between enterprises is becoming more and more fierce. As a more important factor affecting the competitiveness of enterprises, the method of task scheduling in company management has been paid more and more attention by company managers [1, 2]. Therefore, the method of task scheduling in company management has become a hot topic in the field of enterprise, which has been paid attention by many experts. At present, the main task scheduling models in company management mainly include the task scheduling method based on the fuzzy clustering algorithm, the task scheduling method based on the optimization genetic algorithm, and the task scheduling method based on the ant colony algorithm [3, 4]. Because of the wide development space of the task scheduling method, it has become the focus of many experts.

Task scheduling model has become the core problem in the field of company management. In the management of the company, the pipeline work task management mode is

adopted, and the multithread and multitask mode is adopted for the work task under this management [5]. In the process of multitask scheduling, the correlation between tasks is very complex, which is greatly affected by the scheduling order. The traditional task scheduling model is based on single linear order optimization [6, 7]. Once there are too many tasks, the contradiction between different task scheduling will increase, the stability of the model will decline, and the efficiency of scheduling will be low. However, it is difficult to assign a schedulable priority to each task in the design model due to the limitation of multiple times. At the same time, it is difficult to avoid the undesirable priority inversion. Therefore, the design model cannot be used to analyze the fixed priority scheduling theory.

Aiming at the defects of the traditional algorithm, a reasonable scheduling method based on the multiconstraint scheduling model is proposed. In this paper, the principle of multitask scheduling in company management is described in detail [8, 9]. A multitask scheduling network model based on priority is constructed by using this principle. Virtual scheduling is carried out for the network model, and the

optimal solution is searched in the solution space, and the reasonable multitask scheduling method in company management is obtained [10]. The experimental results show that the improved algorithm can improve the resource utilization rate and scheduling efficiency and avoid the defect of reduced stability of the scheduling model caused by multiwork doping.

2. Principle of Multitask Scheduling in Company Management

2.1. Analysis of Task Scheduling Model. Task scheduling is simply a problem of allocating resources according to time to complete tasks. The accuracy and rationality of scheduling is directly related to the economic effect of the whole production activity. The goal of task scheduling software is to use advanced computer technology to achieve efficient control of production scheduling [11]. The objectives of the task scheduling software are as follows:

- (1) *Convenient Interaction.* The software should have a friendly man-machine interface, which is convenient, intuitive, and easy to learn. The user can easily realize the man-machine dialogue.
- (2) *Network Management.* As a part of the enterprise internal management system, the task scheduling software should have a good information interface to realize the information interaction with other subsystems. Through the interface, it can realize the functions of task allocation, release, and processing.
- (3) *Controllability of Scheduling Process.* It has the monitoring function for the whole process of task scheduling and timely reflects the problems to the system administrator.
- (4) *Security.* Most of the internal resources involved in software are confidential, so security is very important. The software should have the ability of user identification and authority control.

According to the requirement analysis of task scheduling software, the task scheduling is divided into six functions: plan preprocessing, task scheduling, basic information management, material resource management, job management, and work process tracking. Thus, the domain model of task scheduling is obtained, as shown in Figure 1. The task scheduling model is mainly composed of two modules: task request scheduling and optimal task selection. The classifier classifies tasks into several classes according to their priority and sends them to the corresponding queue buffer and gives each queue a weight. The difference between virtual scheduling and actual scheduling is that the virtual scheduling task allocates a set of resources, while the actual scheduling task allocates a specific resource. Virtual scheduling is developed based on relaxation theory, so virtual scheduling is an intermediate process to solve the scheduling scheme.

After the plan preprocessing function searches the plans, it classifies the plans according to the retrieved results and compiles the dispatch list, which is used to

drive the task scheduling [12]. After receiving the dispatch list issued by the plan, the task scheduling first calls the residual production capacity information in the basic information management, the process information of the product, and the task decomposition, assignment, and sorting based on the example-based contract net scheduling method. Then, the raw materials and semifinished products in the resource management are called to schedule and process the tasks [13, 14]. After the scheduling is completed, the production, man hour quota, and cost are statistically managed. The whole scheduling process is accompanied by work process tracking. Work process tracking tracks the execution status of the plan before the execution of the task scheduling, tracks the logistics and cost information generated in the scheduling process, and tracks the output and quality in the operation management after the scheduling is completed.

2.2. Design of Task Scheduling Architecture. Class model is the core of object-oriented modeling and an important part of UML. Its main function is to define the classes in the system and the relations between them. Class structure in the class model is formed by abstracting meaningful concepts from domain requirements and then encapsulating certain behaviors with them [15, 16]. The initial class model is a concept-based class model, which focuses on the expression of the relationship between classes, while ignoring the expression of the attributes and behaviors in the internal structure of the class. According to the domain requirements and basic framework of the system, the initial class model is constructed abstractly. The following is an analysis and summary of the abstract modeling methods of the initial class model [17, 18]:

- (1) The initial class is extracted from the underlying use case, activity diagram, and information carrier analysis, and the initial class framework is established
- (2) From the use case diagram and the activity diagram to the relationship between information flow and data flow abstract classes in the class diagram
- (3) The mapping from the use case diagram and the activity diagram to workflow in the class diagram is used to reflect business processing logic
- (4) Accurate identification of association types (association, aggregation, composition, and generalization)
- (5) An interface definition or description that expresses the services (operations) provided to the user using the interface
- (6) Proxy object is used to refine association and provide access control; action object is used to provide preprocessing to simplify interaction

The initial class architecture of task scheduling software is shown in Figure 2. The initial class framework of task scheduling is developed according to the business main line of domain application, which embodies the functional abstraction and data processing flow of IPO control.

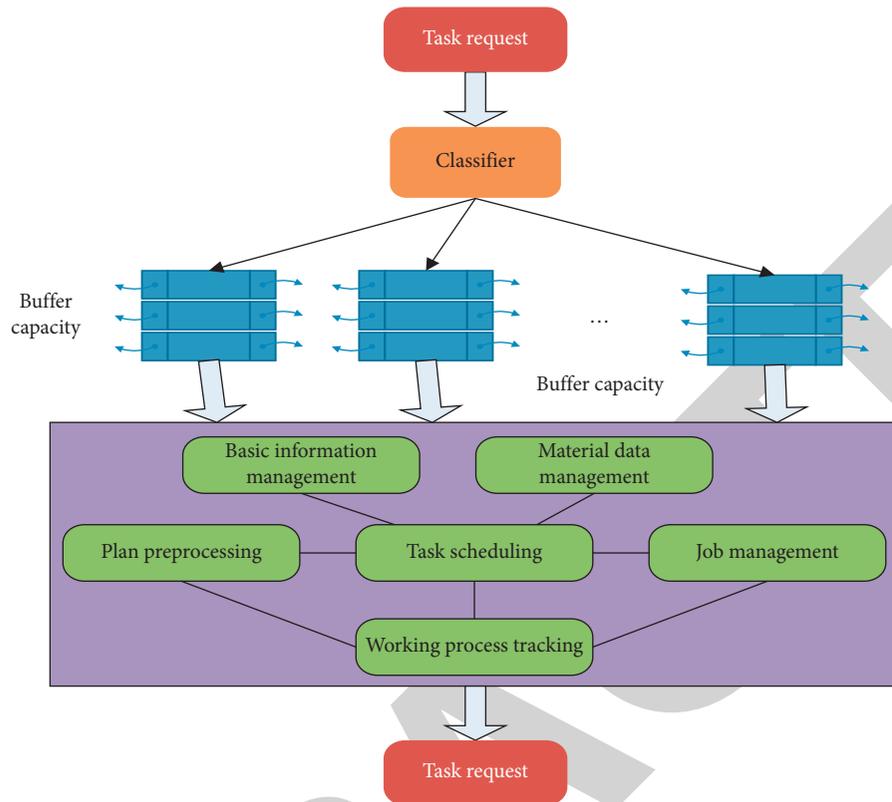


FIGURE 1: Domain model of production task scheduling.

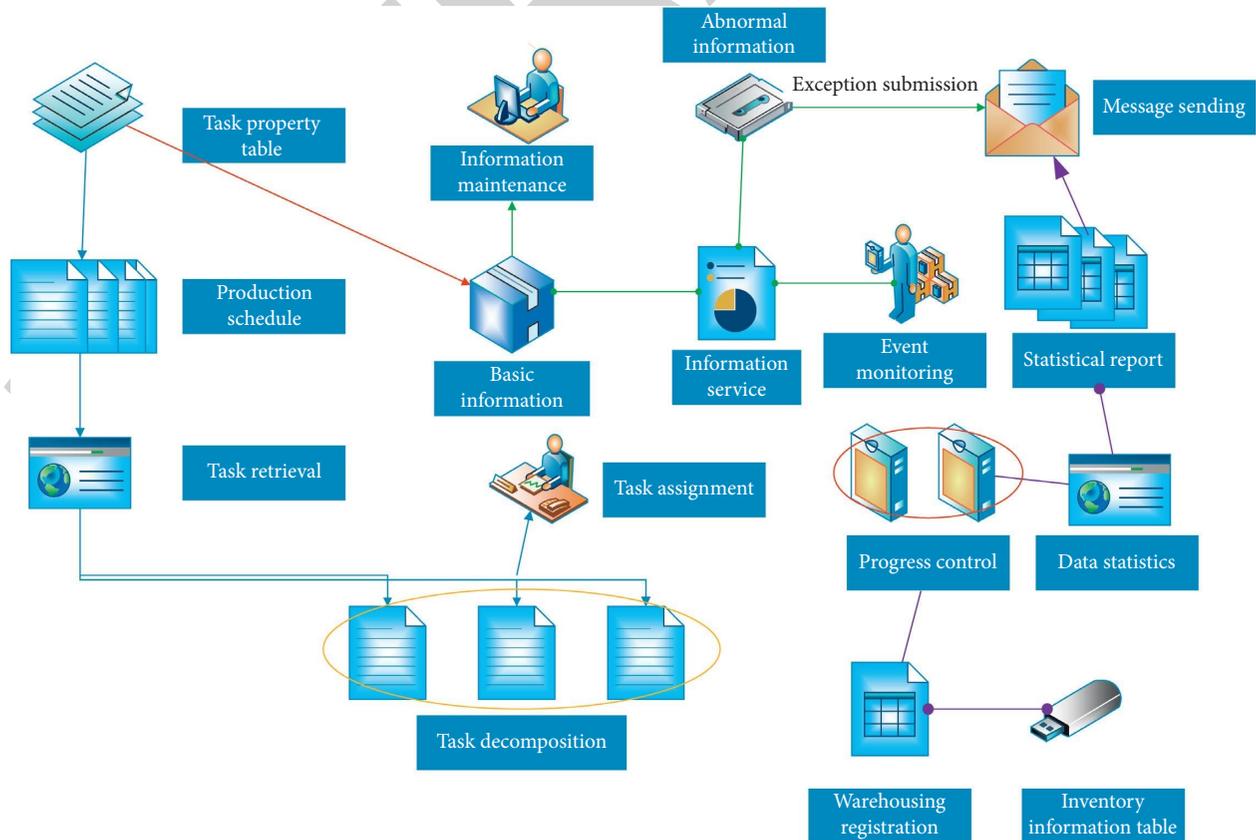


FIGURE 2: Initial class architecture of production task scheduling.

In addition, the initial architecture of the system only describes the status of this class in the system and the relationship between this class and other classes in the system, but fails to describe its detailed design and optimization. Therefore, it is necessary to further analyze and design the middle classification of the initial class framework according to the situation. The optimization method is used to optimize the structure [19–21]. Task assignment class is the core process of collaborative task scheduling. According to the domain analysis, the task assignment class is further analyzed and designed. The structure of the task assignment class is shown in Figure 3.

2.3. Description of Multitask Scheduling Principle. According to the above method, we can solve the utilization ratio of scheduling resources in multitasks. In the process of scheduling tasks, the demand for nonrenewable resources is determined. Therefore, the task scheduling cannot be ignored. Therefore, when solving the multitask scheduling problem, we first simplify the problem [22, 23]:

- (1) Nonupdatable resources are ignored during the operation
- (2) The duration of task completion is only related to the working capacity of the selected renewable resources

The solution of the scheduling problem with multiple tasks is (W, R, S) . W is used to describe the total number of tasks, and ω_i is used to describe i . R is used to describe the resource allocation vector of all tasks, and r_i is used to describe the allocation of resources to task i . S is used to describe the start time of work task, and S_i is used to describe the start time of work task i . S is obtained by calculating W and R . Therefore, the process of solving the scheduling problem is to solve the W and R values. The process of solving the multitask scheduling problem can be described as follows: (1) constructing the priority network of the problem to be scheduled; (2) through virtual scheduling, the solution space is continuously compressed until the end condition is met, and the problem solving process is exited; (3) the solution result (no solution or optimal solution) of the scheduling problem is obtained.

In the management of the company, the pipeline work task management mode is adopted, and the multithread and multitask mode is adopted for the work task under this management. In the process of multitask scheduling, the correlation between tasks is very complex, which is greatly affected by the scheduling order. In the company management, reasonable distribution of work tasks is conducive to improving task efficiency. The principle of scheduling is as follows.

Set $t = \{T_i\} (1 \leq i \leq l)$ as the multitask set in company management, which can be described by the following matrix:

$$t = \begin{bmatrix} T_1^{\text{Type}} & T_1^L & T_1^{\text{In}} & T_1^{\text{Out}} & T_1^E \\ T_2^{\text{Type}} & T_2^L & T_2^{\text{In}} & T_2^{\text{Out}} & T_2^E \\ \dots & \dots & \dots & \dots & \dots \\ T_l^{\text{Type}} & T_l^L & T_l^{\text{In}} & T_l^{\text{Out}} & T_l^E \end{bmatrix}. \quad (1)$$

The row vector of the matrix can be used to describe the attribute set of the corresponding task. Among them, T_i^{Type} is used to describe the types of work tasks; T_1^L is the total length of work tasks; T_i^{In} is the input size of work tasks; T_i^{Out} is the output size of work tasks; T_i^E is the expected value of work tasks by company management. QoS can also be divided into different tasks according to the type of QoS. For example, completion time, labor quantity, and cost.

Set $d = \{D_i\} (1 \leq i \leq m)$ to describe the resource set of a work task in the company management, which can be described by the following matrix:

$$d = \begin{bmatrix} D_1^{\text{rg}} & D_1^{\text{time}} & D_1^{\text{sm}} & D_1^{\text{cb}} & D_1^{\text{jq}} \\ D_2^{\text{rg}} & D_2^{\text{time}} & D_2^{\text{sm}} & D_2^{\text{cb}} & D_2^{\text{jq}} \\ \dots & \dots & \dots & \dots & \dots \\ D_m^{\text{rg}} & D_m^{\text{time}} & D_m^{\text{sm}} & D_m^{\text{cb}} & D_m^{\text{jq}} \end{bmatrix}. \quad (2)$$

The row vector of the matrix is used to describe the attributes of various resources in the corresponding task. Among them, D_i^{rg} is used to describe the working ability of workers; D_i^{time} is used to describe the completion time of tasks; D_i^{sm} is used to describe the number of workers required to complete tasks; D_i^{cb} is used to describe the use cost of a single worker; D_i^{jq} is used to describe the use cost of machines. In order to comprehensively consider the performance of various resources in multitask scheduling, it is necessary to normalize the performance attribute values of resources.

$$V = \begin{cases} \frac{V_i^{\text{rg}} - \min(\text{rg})}{\max(\text{rg}) - \min(\text{rg})}, & \max(\text{rg}) \neq \min(\text{rg}), \quad 1 \leq i \leq m, \\ 1, & \max(\text{rg}) = \min(\text{rg}), \end{cases} \quad (3)$$

where V_i^{rg} is the working ability attribute value of the i th worker, $\min(\text{rg})$ is the minimum value of the working ability performance attribute value of m workers in the scheduling resource set, and $\max(\text{rg})$ is the maximum value of the working ability attribute value of m workers. Obviously, formula (1) normalizes the attribute value of workers' working ability to the interval of $[0, 1]$.

The utilization rate of scheduling the resource set under multitasks can be described by a vector, where $D_{i,\text{uti}} (1 \leq i \leq m)$ is used to describe the corresponding scheduling resource utilization. For example, for labor-intensive tasks, the utilization rate of scheduling resources is mainly reflected by worker resources, and its calculation formula can be described as follows:

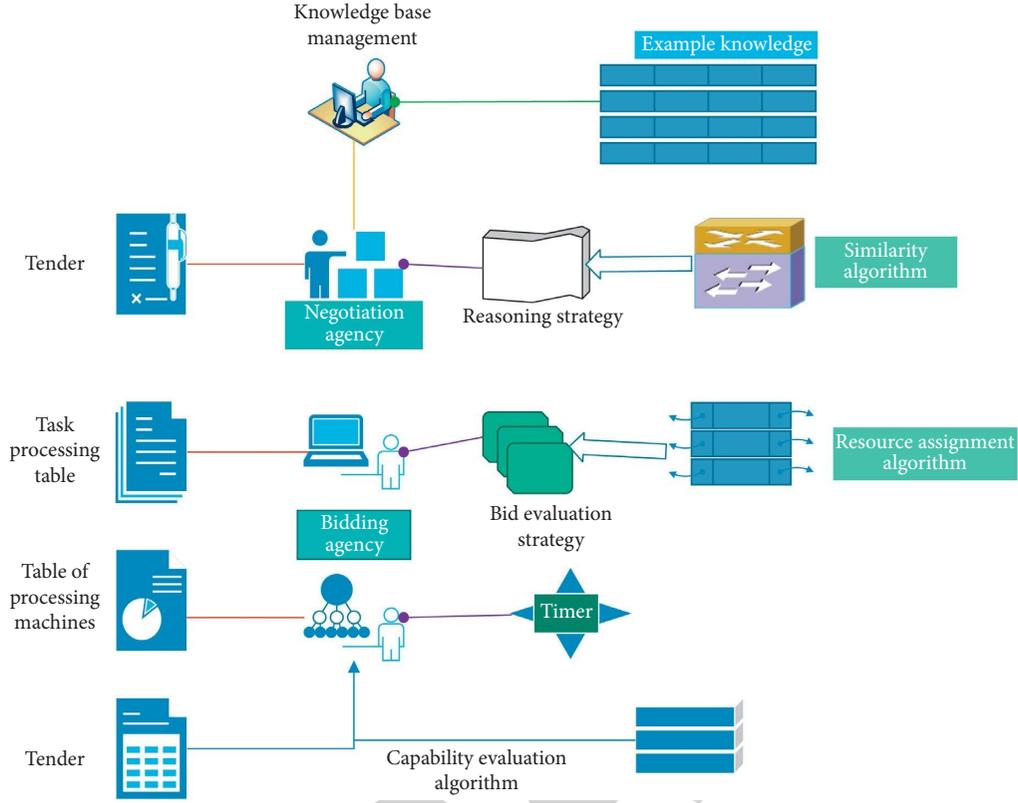


FIGURE 3: Task assignment class architecture.

$$D_{i,uti} = \frac{\sum_{j=1}^l (T_{j,etime} - T_{j,stime}) \times \Delta_{ij}}{T_{time}}, \quad 1 \leq i \leq m, \quad (4)$$

where $T_{j,etime}$ and $T_{j,stime}$ are used to describe the end time and the start time of the corresponding task; $\Delta_{ij} = 1$ is used to describe that task J is allocated on schedule I , otherwise $\Delta_{ij} = 0$.

The overall load balance V_m of scheduling resources under multiple tasks can be described by the standard value of variance. The specific calculation formula is as follows:

$$\left\{ \begin{array}{l} V_m = \sqrt{\frac{1}{m} \sum_{i=1}^m (V_{i,uti} - \bar{V})^2}, \\ \bar{V} = \frac{1}{m} \sum_{i=1}^m V_{i,uti}. \end{array} \right. \quad (5)$$

The optimization goal of scheduling is to achieve the minimum task scheduling span $T_{j,etime}$ and the minimum system resource overall load V_m through the reasonable scheduling of task set t on the scheduling resource set v . In the management of the company, the pipeline work task management mode is adopted, and the multithread and multitask mode is adopted for the work task under this management. In the process of multitask scheduling, the correlation between tasks is very complex, which is greatly affected by the scheduling order. The traditional task

scheduling model is based on single linear order optimization. Once there are too many tasks, the contradiction between different task scheduling will increase, the stability of the model will decline, and the efficiency of scheduling will be low.

In addition, due to the need to maintain the stack of each task, there is also a lot of memory overhead. In order to reduce the space-time cost of real-time multitasking, it is common to map multiple tasks to the same thread, so that the thread will receive and process the requests of each task in the queue in turn. However, it is difficult to assign a schedulable priority to each task in the design model due to the limitation of multiple times. At the same time, it is difficult to avoid the undesirable priority inversion. Therefore, the design model cannot be used to analyze the fixed priority scheduling theory. The other method is to assign an implementation thread for each task; that is, the number of tasks is equal to the number of threads, which will have the disadvantage of high space-time cost.

3. Research on Multitask Scheduling Algorithm Based on Priority

3.1. Priority Workflow Scheduling Prototype. The level of a workflow can be determined by the level of user payment, as shown in Figure 4. For example, if a cloud service provider divides users into free users, VIP, the priority value can be 0, 1, and 2. If there are multiple workflows applying for one cloud resource at the same time, the higher the level of

workflow, the easier it is to obtain the cloud computing resources needed. If a workflow does not get the cloud computing resources it needs for a long time due to its low level, we can set a value for the workflow. Counting from 0 and adding one after each scheduling. If the value exceeds the set value after multiple rounds of scheduling, the workflow can obtain the cloud computing resources it needs regardless of its priority level in the next round of scheduling. The set value can be calculated according to the following steps.

Firstly, suppose that there are M workflows applying for the required cloud computing resources and the cloud resource provider divides the user level of cloud computing resources into N levels, then the value can be set as M/N . For the workflow being executed, its priority may be lower than that of the workflow that is applying for cloud computing resources. We prefer to let the workflow continue to execute, rather than stop its execution forcibly, because this will waste the consumed resources and make the total consumption more.

3.2. Scheduling Algorithm Based on Workflow Priority (PISA). PISA algorithm is based on the access control model; that is, the workflow task can enter the scheduling sequence and calculate the priority factor only after it has been authenticated by the access control model. Therefore, in PISA, this paper first introduces the access policy audit step. This is based on access control model authentication. When a workflow applies for cloud computing resources, the cloud computing resource provider first queries the access policy library and then queries the corresponding access policy according to the application information of the workflow. If the access policy is accessible, the workflow can enter the second step, that is, entering the workflow scaling sequence. First of all, the workflow task of computing resource application is authenticated. Only the task that meets the rules can enter the second step. In order to facilitate the experiment, we call the access policy of the workflow task as WAPC. The definition of WAPC is as follows:

$$\text{WAPC}[\text{work, task, } (T_s, T_c, P)], \quad (6)$$

where T_s is the start time of task T_i , T_c is the end time of task T_i , and P is a periodic expression. The value priority defines the corresponding level of cloud computing resources that the workflow task can obtain. If the level of resource requested by the workflow task matches the value of priority, the request of the workflow will be passed. If the level of the workflow does not meet the corresponding requirements or the time quota is insufficient, the application is rejected. We define the time quota T_q , which indicates the maximum execution time quota of the task. Work defines the set priority value of the workflow to which the workflow task belongs. The authorization rules for the policy are as follows.

Suppose that user U applies to the cloud computing service provider for a workflow instance W , and the task ID contained in the request obtains the required cloud computing resources at time T . In the workflow task, the priority

value of the corresponding level resources required is priority. If $T_s < t < T_c$, the access application is passed, otherwise the application is rejected.

For a workflow with multiple tasks, the completion of one task may depend on the completion degree of other tasks, so their running time is related. In order to facilitate the statistics of the possible time completion rate of a task, we define the time parameter T_q , which takes the time parameters of each task into account:

$$\text{time} = \frac{\text{Min}(t_i) + \text{WS}(t_i)(t_e - t_s)}{\sum_{i=1}^m t_i}, \quad (7)$$

where $\text{Min}(t_i)$ is the minimum execution time of task t_i , $\text{WS}(t_i)$ is the time parameter required for the task to use all available cloud computing resources, and t_q cannot be greater than $t_e - t_s$, otherwise its request will not be passed.

For a workflow W_i , it may contain multiple tasks. Similarly, users can specify the hierarchical weights of these tasks. Suppose that there are n tasks whose rank weights are T_1, T_2, \dots, T_n and W_i are the priority weights of the workflow to which these tasks belong, then the average rank weight bps of the tasks is

$$\text{bps} = \sum_{i=1}^m \frac{t_i}{N}. \quad (8)$$

Figure 5 only shows the work scheduling process, so there is no numerical value in the graph. It is only a reference for the process, and there may be other types. Figure 5 illustrates the scheduling process of workflow tasks.

For task scheduling, we also have the following considerations:

- (1) The amount of cloud computing resources that a cloud computing service provider can provide is limited.
- (2) Tasks belonging to high-level workflow should be executed first, which means that the first-level impact factor A is more important than the second-level impact factor B . In some cases, some tasks of low-level workflow may be higher in the scheduling queue after being calculated by the above formula, so we need to add the above conditions.

The second step is to bind the tasks that have been prioritized to the virtual machine and establish the mapping relationship between tasks and virtual machines. This involves the concept of quality of service. In cloud computing, service quality is used to measure the user satisfaction with cloud computing services. Different users have different standards for QoS, some may need to ensure real-time performance, while others want low cost. According to the importance of the user's task, the priority factor of the related task is calculated, and all the tasks participating in the scheduling are prioritized. Then the virtual machine resources are sorted. The higher the level of the task, the more abundant the resources can be obtained.

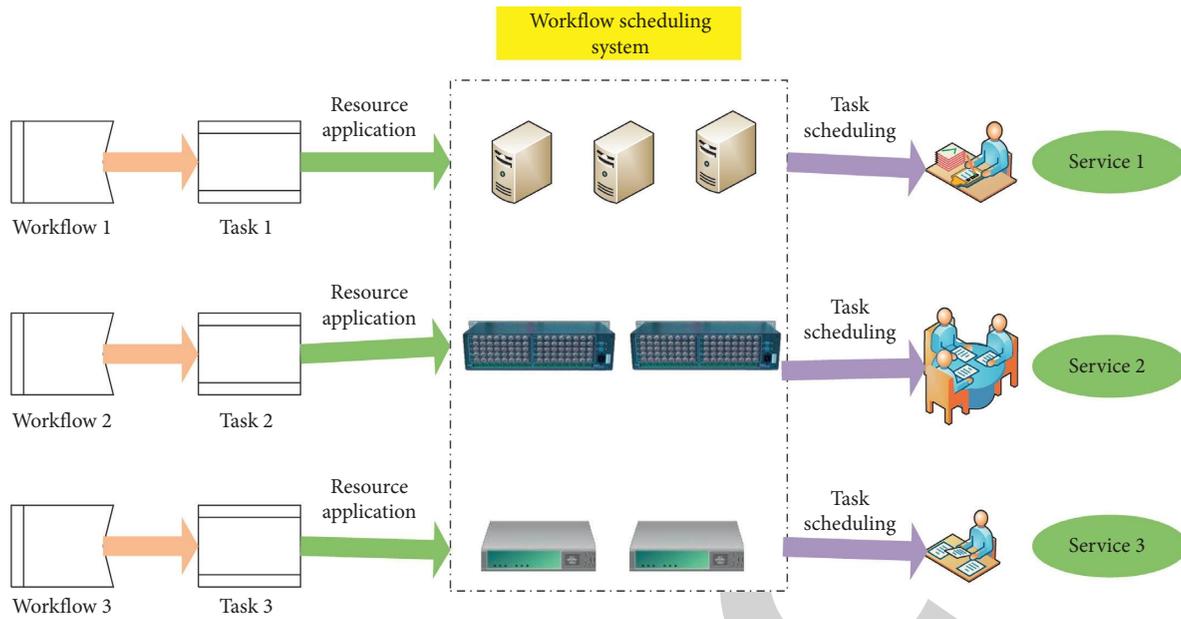


FIGURE 4: Workflow scheduling system.

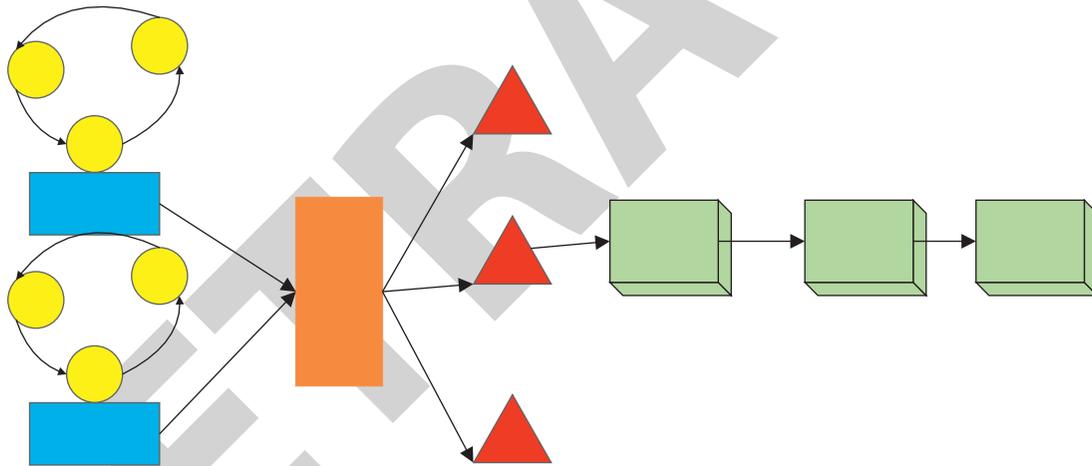


FIGURE 5: Task scheduling block diagram.

4. Simulation Experiment and Analysis

4.1. *GABP Optimized RO*. The cloud simulation environment used in this paper includes 10 virtual machines, each of which performs a task. We set up five users to apply to cloud computing service providers. Each user proposes a workflow, and each workflow package contains two tasks. Based on the convenience of experiments, this paper assumes that all tasks for priority factor calculation have passed the authentication of the access control model, and we only consider the priority of tasks without considering the order of task execution. In order to verify the rationality of the multitask scheduling method based on the multiconstraint scheduling algorithm, an experiment is needed. In the process of the experiment, the resource utilization, personnel utilization, and scheduling efficiency are used to

measure the performance of the multitask scheduling method. The relationship among enterprise personnel, resources, and work tasks is described in Figure 6.

The data related to the relationship among personnel, resources, and work tasks are described in Table 1.

The traditional algorithm and the improved algorithm are, respectively, used for multitask scheduling in company management. In the process of the above experiment, the personnel utilization rate of different algorithms is shown in Figure 7.

The traditional algorithm and the improved algorithm are used to schedule multitasks in company management, respectively. The scheduling efficiency of different algorithms is shown in Figure 8.

According to Table 1, it can be seen that using the improved algorithm for reasonable scheduling under

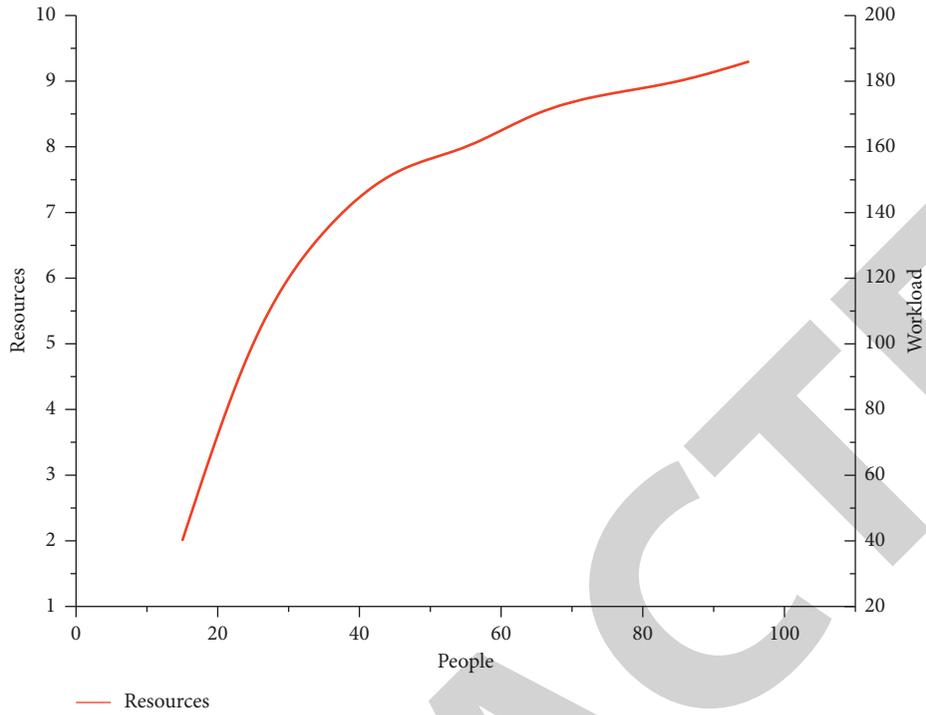


FIGURE 6: Relationship among personnel, resources, and tasks in enterprises.

TABLE 1: Comparison of resource utilization of different algorithms.

Number of experiments	Number of personnel	Number of resources	Workload
1	15	2	20
2	25	4	35
3	35	6	55
4	45	8	70
5	55	10	80
6	65	12	120
7	75	14	150
8	85	16	180
9	95	18	210
10	105	20	220

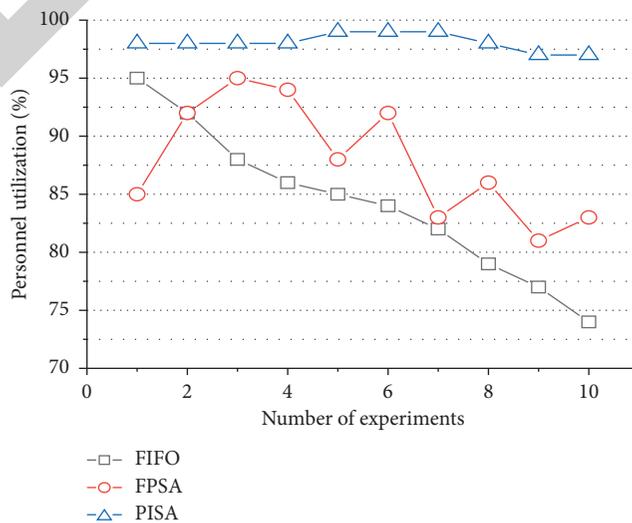


FIGURE 7: Comparison of personnel utilization of different algorithms.

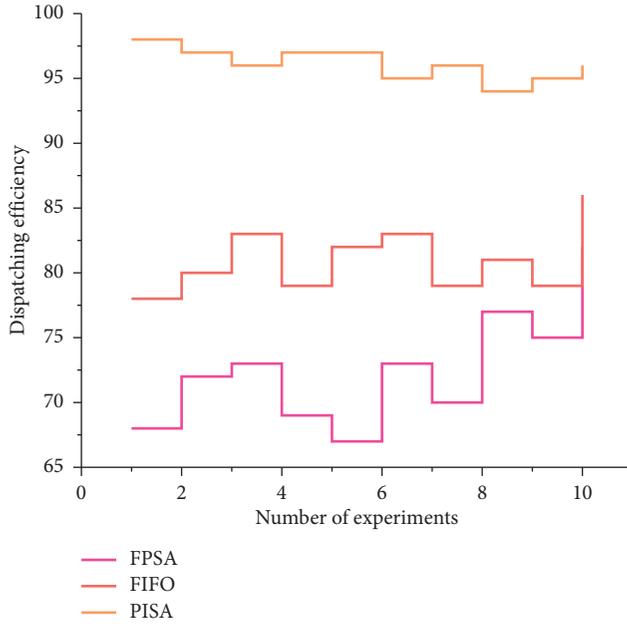


FIGURE 8: Comparison of scheduling efficiency of different algorithms.

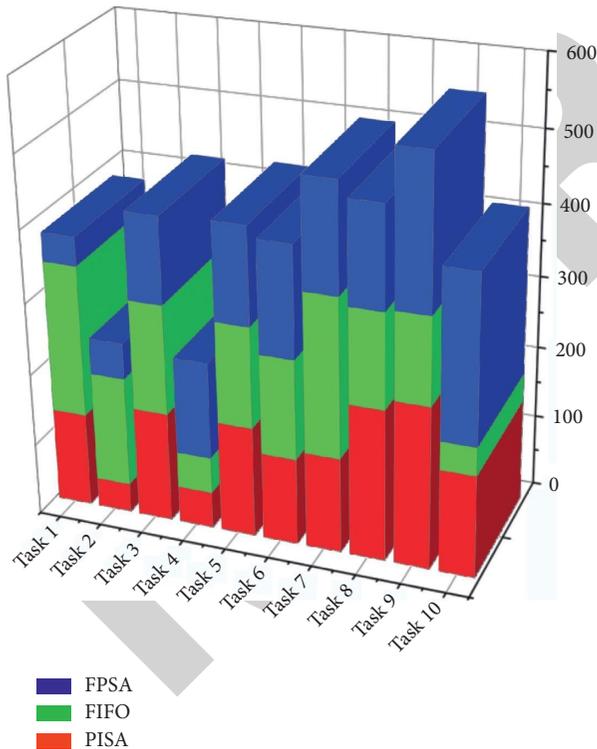


FIGURE 9: Comparison chart of execution time experiment results.

multiple tasks in company management can improve resource utilization and personnel utilization and effectively improve scheduling efficiency. The experimental data are compared and analyzed, as shown in Figures 9 and 10. Generally speaking, the task scheduling algorithm based on PISA has higher execution efficiency for high-priority tasks than those based on FIFO and fixed-priority FPSA.

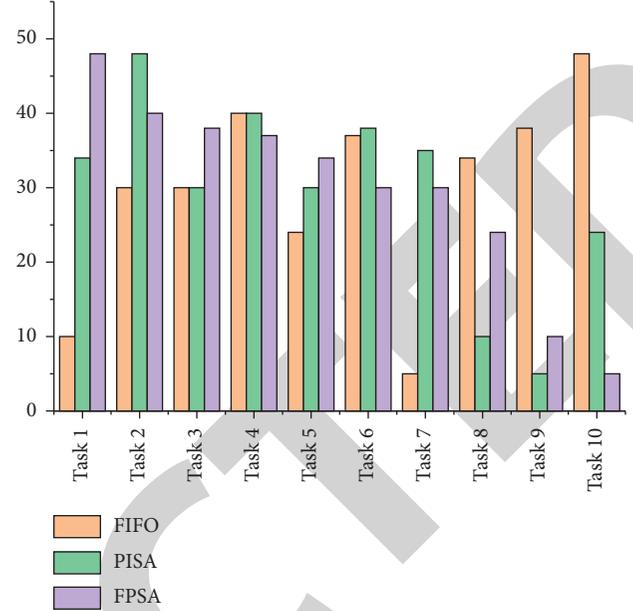


FIGURE 10: Execution cost comparison chart.

In terms of execution time, taking MS as the unit and 135 ms as the boundary, we can see in Figure 9 that in the PISA test, 5 tasks were completed within 135 ms, including 3 tasks with high priority. In the FIFO test, 5 tasks were completed in time, but only 1 was of high priority. Obviously, the task scheduling strategy based on the PISA algorithm is more suitable for the resource allocation of high-priority tasks. Compared with FPSA fixed priority, both of them can provide better services for high-priority tasks. Therefore, the efficiency difference is relatively small, and the efficiency of FPSA is higher than that of PISA because FPSA takes the maximum satisfaction of high-priority tasks as the optimization condition, and optimization is more extreme.

From the perspective of the execution cost, we can see from Figure 10 that although the fixed priority can meet the user's execution requirements for important priorities to a certain extent, its priority is fixed. Only the priority level of the workflow is considered, and the priority value of each task included in the workflow is not considered. Therefore, the implementation cost of FPSA is not as reasonable as that of PISA. PISA has more advantages in controlling the overall execution cost of workflow tasks. Because of its randomness, FIFO has no corresponding optimization in the execution cost, so its cost-effectiveness ratio is very low.

5. Conclusion

In order to solve the problem that the stability of the model will decline due to the interference of human factors in the process of multitask scheduling in the traditional algorithm, a reasonable scheduling method based on the priority principle in company management based on the multi-constraint scheduling model is proposed. This paper expounds the principle of multitask scheduling in company management, constructs a priority-based multitask scheduling network model based on the above principles,

conducts virtual scheduling on the model, searches the optimal solution in the solution space of the scheduling problem, and obtains the reasonable multitask scheduling method in the company management. The experimental results show that the improved algorithm can improve the resource utilization rate and scheduling efficiency and avoid the defect that the stability of the scheduling model is reduced in the case of multiwork doping, and the satisfactory effect is achieved. The use environment of the model is more complex, and the role mapping and permission inheritance between multiple domains are taken into account, so that it can be applied to a wider range of fields. The various types of the cloud computing simulation platform are continued to be studied and strived to expand them to better simulate task scheduling.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Jilin Social Science Fund Project: research on the influence mechanism of Confucian culture on corporate governance (no. 2018B71).

References

- [1] A. Noroozi and H. Mokhtari, "Scheduling of printed circuit board (PCB) assembly systems with heterogeneous processors using simulation-based intelligent optimization methods," *Neural Computing and Applications*, vol. 26, no. 4, pp. 857–873, 2015.
- [2] S. Jia, G. Yan, A. Shen, and J. Zheng, "Dynamic simulation analysis of a construction and demolition waste management model under penalty and subsidy mechanisms," *Journal of Cleaner Production*, vol. 147, pp. 531–545, 2017.
- [3] H. Zhang, J. Xie, J. Ge, J. Shi, and W. Lu, "Optimization model and online task interleaving scheduling algorithm for MIMO radar," *Computers & Industrial Engineering*, vol. 127, pp. 865–874, 2019.
- [4] L. He, Y. Ren, Z. X. Feng et al., "Numerical simulation and design optimization Research of slope ratio and platforms influence on a high rock cut slope engineering stability," *Advanced Materials Research*, vol. 1065-1069, pp. 151–158, 2015.
- [5] O. Z. Sharaf and M. F. Orhan, "Thermodynamic analysis and optimization of densely-packed receiver assembly components in high-concentration CPVT solar collectors," *Energy Conversion and Management*, vol. 121, pp. 113–144, 2016.
- [6] Y. L. Xie, D. X. Xia, L. Ji et al., "An inexact stochastic-fuzzy optimization model for agricultural water allocation and land resources utilization management under considering effective rainfall," *Ecological Indicators*, vol. 92, pp. 301–311, 2017.
- [7] A. Veremyev, V. Boginski, and E. L. Pasiliao, "Potential energy principles in networked systems and their connections to optimization problems on graphs," *Optimization Letters*, vol. 9, no. 3, pp. 585–600, 2015.
- [8] M. K. Mehlawat, P. Gupta, and W. Pedrycz, "A new possibilistic optimization model for multiple criteria assignment problem," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 1775–1788, 2018.
- [9] B. Ai, C. C. Wang, and B. Y. Xu, "Optimization of initialization conditions in Aspen simulation process for batch distillation," *Modern Chemical Industry*, vol. 37, no. 12, pp. 190–192, 2017.
- [10] H. Vanharanta and E. Markopoulos, "The applied philosophy concept for management and leadership objects through the company democracy model," *Theoretical Issues in Ergonomics Science*, vol. 20, no. 2, pp. 178–195, 2019.
- [11] F. Zhang and J. Li, "An improved particle swarm optimization algorithm for integrated scheduling model in AGV-served manufacturing systems," *Journal of Advanced Manufacturing Systems*, vol. 17, no. 3, pp. 375–390, 2018.
- [12] M. Dobroselskyi and R. Madleňák, "Model of waste transportation management in the conditions of a production company," *Transportation Research Procedia*, vol. 40, pp. 1023–1029, 2019.
- [13] T. Ning and X. P. Wang, "Study on disruption management strategy of job-shop scheduling problem based on prospect theory," *Journal of Cleaner Production*, vol. 194, pp. 174–178, 2018.
- [14] C. C. David, L. C. D. José, J. M. R. Javier et al., "Self-management model in the scheduling of successive appointments in rheumatology," *Reumatologia Clinica*, vol. 15, no. 6, pp. 333–337, 2018.
- [15] H. Cornelius, S. Zhichao, R. M. Oliveira et al., "Knowledge-aided informed dynamic scheduling for LDPC decoding of short blocks," *Iet Communications*, vol. 12, no. 9, pp. 1094–1101, 2018.
- [16] S. Carrye, "Invisible disabilities: perceptions and barriers to reasonable accommodations in the workplace," *Library Management*, vol. 40, no. 1-2, pp. 113–120, 2019.
- [17] B. Jiang, C. J. Tang, and C. Yan, "A stochastic programming model for outpatient appointment scheduling considering unpunctuality," *Omega*, vol. 82, pp. 70–82, 2019.
- [18] M. Chen, S. Lu, and Q. Liu, "Uniform regularity for a Keller-Segel-Navier-Stokes system," *Applied Mathematics Letters*, vol. 107, Article ID 106476, 2020.
- [19] Y. Qin, G. Zeng, R. Kurachi et al., "Energy-efficient intra-task DVFS scheduling using linear programming formulation," *IEEE Access*, vol. 7, pp. 30536–30547, 2019.
- [20] C. Zeng, J. Tang, and Z. P. Fan, "Auction-based cooperation mechanism for cell part scheduling with transportation capacity constraint," *International Journal of Production Research*, vol. 57, no. 11-12, pp. 3831–3846, 2019.
- [21] K. Cui and X. Jing, "Research on prediction model of geotechnical parameters based on BP neural network," *Neural Computing and Applications*, vol. 31, no. 12, pp. 8205–8215, 2019.
- [22] S. Rubab, M. F. Hassan, A. Mahmood et al., "QoS based multi-constraints bin packing job scheduling heuristic for heterogeneous volunteer grid resources," *The International Arab Journal of Information Technology*, vol. 16, no. 4, pp. 661–668, 2019.
- [23] A. Syed and G. Fohler, "Efficient offline scheduling of task-sets with complex constraints on large distributed time-triggered systems," *Real-Time Systems*, vol. 55, no. 2, pp. 209–247, 2019.