

Research Article

Multiple Wirelessly Powered Sensing Platform Scheduling Algorithm Based on Dynamic Priority Preemption

Zhijun Xie,^{1,2} Tao Zhang¹, Chenlu Wang,^{2,3} Jiancheng Yu,^{1,2} and Roozbeh Zarei⁴

¹*Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China*

²*Institute of Mass Spectrometry Technology and Application, Ningbo University, Ningbo 315211, China*

³*School of Materials Science & Chemical Engineering, Ningbo University, Ningbo 315211, China*

⁴*School of Information Technology, Deakin University, Melbourne 3125, Australia*

Correspondence should be addressed to Tao Zhang; ztao0828@foxmail.com

Received 3 January 2020; Revised 13 December 2020; Accepted 1 June 2021; Published 25 June 2021

Academic Editor: Ghufran Ahmed

Copyright © 2021 Zhijun Xie 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.

The available energy of a wirelessly powered sensing platform is not enough, and there are constant real-time tasks to join the wirelessly powered sensing platform to run. So the wirelessly powered sensing system composed of many wirelessly powered sensing platforms is easy to enter the overloaded state, which may cause some tasks not to be executed on time. Therefore, to obtain as much task value as possible for the wirelessly powered sensing system when it is under the overloaded state, it is essential to design a reasonable task scheduling algorithm to arrange the task execution order. In this paper, we propose a policy named Wirelessly Dynamic Allocation Priority (WDAP) policy suitable for the wirelessly powered sensing system. The proposed WDAP is divided into a dynamic task priority allocation policy and a dynamic node priority allocation policy. Firstly, this paper analyzes the dynamic value density based on task value and execution time, studies the urgency of execution according to the execution time and the remaining idle time, and proposes the energy intensity through the task energy consumption and execution time. Based on the three impact factors of dynamic value density, urgency, and energy intensity, a policy for dynamic task priority allocation is proposed. Then, a policy for dynamic node priority allocation is proposed by combining the available energy and the energy acquisition speed of the nodes. Finally, the algorithm suitable for the wirelessly powered sensing system is proposed named Wirelessly Dynamic Real-time Task Scheduling (WDRTS) algorithm based on the WDAP. The algorithm clarifies the execution order of each task, responds to high-priority tasks first, and effectively guarantees task benefits. The experimental results show that compared with the main algorithms used in the literature among which is Generalized Earliest Deadline First, the WDRTS algorithm reduces the number of preemptive tasks by at least 36.49% and increases the successful scheduling rate of tasks by at least 15.17% and the overall system task income by at least 16.37% under high load.

1. Introduction

The wirelessly powered sensing system [1–3] has the characteristic of converting RF signals to its own energy, which solves the problem of insufficient battery life of traditional IoT nodes [4, 5]. The wirelessly powered sensing system contains many wirelessly powered sensing platforms, and it is widely used in many fields, like crack detection [6, 7] and warehouse storage [8]. And once the pressure data is abnor-

mal, the relevant personnel is notified to maintain the bridge. At the same time, the wirelessly powered sensing platform receives RF energy for its own power supply, without removing the bridge deck and replacing the battery, which greatly reduces the maintenance cost. As a new generation of wireless sensor network nodes [9–11], the wirelessly powered sensing platform integrate sensing, computing, and communication capabilities. They are equipped with real-time system awareness environment data [12, 13]. The real-time

task scheduling algorithm directly determines the performance of the real-time system [14, 15], so it is available how to design a real-time task scheduling algorithm.

At present, the most suitable task scheduling algorithm under an overloaded state is the algorithm based on the priority preemption policy, and the factors most referenced by the priority preemption policy are task value and task deadline. The general priority allocation policy is that the higher the task value and the closer the deadline, the higher the priority of the task. Traditional task scheduling algorithms such as the Earliest Deadline First (EDF) algorithm consider deadlines to assign priorities. The Generalized Earliest Deadline First (GEDF) algorithm is proposed based on the EDF algorithm, deadline is earlier, the task value is higher, and the assigned task will have higher priority. The Highest Value First (HVF) algorithm considers task value to assign priorities. The Least Slack Time Rate First (LSF) algorithm considers the urgency of task slack time to assign priorities. The dynamic value density (DVD) algorithm considers time urgency and task value to assign priorities. The Dynamic Priority Assignment (DPA) algorithm considers the task value and deadline to assign the task priority [16]. The above algorithm is described in detail in Section 2.2. These algorithms only consider the task value and the urgency of the deadline and ignore the task energy consumption. The wirelessly powered sensing platform can obtain the RF energy emitted without relying on battery power, but the energy capture efficiency is not enough to keep working for a long time [17]. If the wirelessly powered sensing system runs these algorithms, it may result in some wirelessly powered sensing platforms having insufficient energy, going to sleep in advance, and being unable to continue to monitor environmental data. Therefore, these algorithms are not suitable for the wirelessly powered sensing platform to run. For example, a wirelessly powered sensing system can be used to monitor the fire in a barn. It can be deployed inside the barn to detect the barn temperature and humidity data. If you do not consider the energy and only consider the task value and task deadline, it may lead to some wirelessly powered sensing platforms with low energy becoming exhausted in advance due to too many tasks. This not only can cause the tasks on these nodes to be unable to continue but also may lead to disruption in detecting the indoor temperature and humidity in real time. Therefore, when the barn has local high temperature and dry conditions, it may trigger the fire and cause huge property damage. Therefore, it is very important to reasonably allocate task priority according to task energy consumption and to combine the energy differences of the wirelessly powered sensing platform for task scheduling.

In a wirelessly powered sensing system, there are two major challenges. First, it is not enough to consider priority allocation only in task deadline and task value, but also the energy consumed by the task. At the same time, it is not enough to consider only a single wirelessly powered sensing platform. It is necessary to consider the task scheduling algorithm based on the energy difference of multiple wirelessly powered sensing platforms.

The WDAP includes two parts, a dynamic task priority allocation policy and a dynamic node priority allocation pol-

icy, based on the task value, task deadline, and task consumption energy as factors that affect the allocation of task priority. Combining the two parts of available energy and the energy capture speed of each wirelessly powered sensing platform, a dynamic node priority allocation policy is proposed. At the same time, according to the task priority and node priority, a multiple wirelessly powered sensing system scheduling algorithm based on priority preemption is proposed to reasonably schedule real-time tasks and solve the bumpy problem. The work done in this paper greatly increases the schedulability of real-time tasks, reduces energy consumption, and improves the success rate of tasks performed by the wirelessly powered sensing system.

To summarize, the main contributions of our work are as follows:

- (1) This paper increases the task energy consumption to design a policy for dynamically assigning task priority. The dynamic task priority allocation policy can dynamically assign priorities to real-time tasks and reasonably arrange the task execution order
- (2) Combining the available energy of each wirelessly powered sensing platform and the energy acquisition speed, this paper proposes a policy for dynamic node priority allocation. It dynamically assigns node priority to each wirelessly powered sensing platform and assigns tasks for execution in order of node priority
- (3) According to the WDAP, this paper proposes a WDRTS suitable for the wirelessly powered sensing system to reasonably schedule tasks. The algorithm deduces the bump coefficient by theoretical deduction and reduces the bump behavior of the algorithm and the number of task preemptions. At the same time, the WDRTS algorithm is executed in order of task priority. Compared with the main used algorithms listed above, the execution success rate and the system revenue are increased. To a certain extent, it solves the problem that the wirelessly powered sensing system loses a lot of task value due to overloaded problems

The paper is organized as follows. Section 1 introduces mainly writing the background, significance, and contribution of this article. Section 2 mainly introduces the experimental platform used and development status. Section 3 mainly introduces the main framework of this article and the symbols that need to be used. Section 4 mainly introduces the process of dynamic task priority allocation policy. Section 5 mainly introduces the process of dynamic node priority allocation policy. Section 6 mainly proposes the WDRTS algorithm and solutions to the bump problem. Section 7 is mainly about the experimental part, and Section 8 is about the summary of this article and thought about the future.

2. Related Work

2.1. Wirelessly Powered Sensing Platform. In recent years, research on a wirelessly powered sensing platform has



FIGURE 1: WISP physical map. It can convert RF signals into its own energy and is a platform for computing, sensing, and communication.

become increasingly popular. The wirelessly powered sensing platform is a wirelessly powered RF tag that integrates sensing, computing, and communication. It has many advantages such as its small size, strong computing power, wireless identification, and integration of various peripherals. To ensure the normal execution of each task, it is necessary to design a reasonable real-time task scheduling policy to reasonably arrange the execution order of tasks. The goal of real-time task scheduling is to satisfy the time constraints of each task and respond to external requests in a timely manner.

The wireless identification and sensing platform is one of the wirelessly powered sensing platforms that integrates sensing, computing, and communication functions and is combined with RFID readers. Figure 1 shows WISP5.0.

At the same time, there needs to introduce the task attribute symbols related to this article. Assume that the task set in a real-time task system is $T = \{T_1, T_2, \dots, T_n\}$, where task T_i has the following properties:

- (1) P_i refers to the execution cycle of T_i . If T_i is not a periodic task, the cycle is determined to be infinite
- (2) D_i refers to the relative deadline of T_i , that is, the maximum acceptable execution time
- (3) C_i refers to the theoretical execution time of T_i
- (4) b_i refers to the time at which T_i arrives and is ready for execution
- (5) e_i refers to the completion time of T_i
- (6) d_i refers to the absolute deadline of T_i
- (7) V_i refers to the task value of T_i , and we use it to measure the importance of the task
- (8) E_i refers to the energy consumption of T_i

Assume that the task switching time is 0 or so small that it can be ignored.

2.2. Development Status. The traditional real-time task scheduling algorithms include the EDF algorithm, the LSF algorithm, the HVF algorithm, the DVD real-time scheduling algorithm, and so on. Table 1 briefly introduces the missing feature of each classic algorithm.

Based on the deadlines of arrival tasks, the EDF [18, 19] selects a task with the earliest deadline for priority scheduling. The deadline is earlier, and its priority is higher [20]. The biggest advantage of this algorithm is to increase processor utilization; however, it cannot handle an overloaded state.

TABLE 1: Algorithm comparison. Here are some of the missing features of the algorithm.

Algorithm name	Algorithm shortcomings
EDF	Cannot handle overloaded state
GEDF	Not consider the energy factor
LSF	Only consider slack time or deadline
HVF	Only consider task values
DVD	Not consider the energy factor
DPA	Not consider the energy factor

Therefore, based on the EDF algorithm and considering the value and deadline of the task, the Generalized Earliest Deadline First (GEDF) algorithm [21] is proposed to improve the EDF algorithm's ability to handle an overloaded system. The deadline is earlier, the task value is higher, and the assigned task will have higher priority.

The LSF [22] assigns the highest priority to the task with the shortest slack time and ensure the task with the shortest slack time priority response. The shorter the remaining slack time of the task, the more urgent the task is to execute, so the priority assigned by the task is higher. The algorithm solves the limitation of the EDF algorithm to a certain extent. However, this algorithm has the problem of repeated task preemption, which reduces execution efficiency.

The HVF [23] scheduling algorithm is proposed according to the value of the task [24]. The algorithm only considers one factor, that is, the task value. The higher the task value, the higher the priority assigned to the task. The HVF algorithm makes the task of the highest value execute firstly, discard some unimportant tasks, and improve the value gain of the system per unit time. However, one-sided consideration of value will affect the successful execution of low-value tasks and also will affect the total value of the system.

Camponogara et al. proposed the DVD algorithm [25] based on the task value and remaining execution time. The task execution time increases continuously, the task priority increases dynamically, and the task in progress is not easy to preempt. However, considering only the time urgency and the task value, it may lead to the inability to perform low task value and reduce the system task income.

Xia et al. [16] proposed a priority preemption policy named DPA based on the value of the task and the urgency of task execution. It can dynamically assign priorities to increase system flexibility. However, it is not enough to consider only the task value and the remaining time of task execution. It needs to consider from the characteristics of the wirelessly powered sensing platform. The excessive energy

of the wirelessly powered sensing platform cannot be excessively consumed due to the high value of high-energy consumption tasks.

Chniter et al. [26] design a plating algorithm for real-time periodic tasks based on heterogeneous multicore platforms. The algorithm distributes the execution of real-time tasks to different processors. At the same time, in order to reduce energy consumption, a lower processor speed is used to perform tasks. If the above solution is not feasible, then the plan to adjust the task cycle will be taken, or some tasks will be transferred to the processor with the least load. The algorithm reduces the energy consumption of the system, but the premise of this algorithm is that all tasks can be completed before its worst-case execution time, which is the biggest difference from the WDPRS. Due to the limited available energy, the wirelessly powered sensing platform may be in a dormant state for charging at any time, so some tasks may not be successfully executed, and only the most important part of the tasks can be selected for priority execution.

Wang et al. [27] designed a scheduling algorithm for periodic and probabilistic tasks to alleviate the problem that the system is not feasible due to the wrong deadline of the task. The algorithm can dynamically modify the task time parameters to solve this problem. At the same time, in order to reduce energy consumption, the algorithm proactively extends the task cycle. However, the problem that needs to be solved in this paper is the rational assignment of tasks under overloaded conditions, using the available energy of wirelessly powered sensing platform to get the maximum benefit.

In the existing scheduling algorithms, the research on the scheduling algorithms for multiple wirelessly powered sensing platforms under an overloaded state is relatively rare.

Wang [28] simply combined the task value, task running time, and energy consumption, but it cannot reasonably allocate task execution according to the available energy of each wirelessly powered sensing platform and cause the remaining energy of each wirelessly powered sensing platform to be unbalanced. So the system execution income is not high, and low-value tasks cannot be effectively executed.

Guo [29] simply built the energy of the wirelessly powered sensing platform mode. According to the available energy for each wirelessly powered sensing platform to assign tasks, the task benefits are not high. Although these algorithms all run on the wirelessly powered sensing platform, they simply assign tasks to each node. Although the execution is relatively simple and the priority is statically assigned, the low-value tasks with high urgency are not implemented, and the energy waste is serious. The efficiency is low and the value obtained is not high.

When formulating a policy for dynamically assigning task priorities, this paper fully considers the characteristics of the wirelessly powered sensing system's own energy mode and combines the three factors of task value, deadline, and task energy consumption. Furthermore, to take into account the energy difference, this paper combines the two aspects of the available energy of the node and the speed of capturing energy to design a dynamic node priority allocation policy. Finally, according to the dynamic task priority allocation policy and the dynamic node priority allocation policy, this

paper proposes a suitable task scheduling algorithm named WDPRS and analyzes the scheduling relationship which alleviates the problem of system thrashing.

3. System Overview

In this paper, aimed at the available energy of the wirelessly powered sensing platform, a dynamic allocation priority policy is proposed and named WDAP. The WDAP includes the dynamic task priority allocation policy and dynamic node priority allocation policy. And the priority preemption task scheduling algorithm suitable for the wirelessly powered sensing platform is proposed based on WDAP named WDRTS; meanwhile, the bumpy problem of repeated preemption of tasks in the preemption algorithm is proposed. Finally, for the overloaded phenomenon, it is proposed to set the task deceleration factor to slow down the overloaded state. Figure 2 is a block diagram of the algorithm.

At the same time, there needs to introduce the task attribute symbols related to this article. Assume that the task set in a real-time task system is $T = \{T_1, T_2, \dots, T_n\}$, where task T_i has the following properties:

- (1) P_i refers to the execution cycle of T_i . If T_i is not a periodic task, the cycle is determined to be infinite
- (2) D_i refers to the relative deadline of T_i , that is, the maximum acceptable execution time
- (3) C_i refers to the theoretical execution time of T_i
- (4) b_i refers to the time at which T_i arrives and is ready for execution
- (5) e_i refers to the completion time of T_i
- (6) d_i refers to the absolute deadline of T_i
- (7) V_i refers to the task value of T_i , and we use it to measure the importance of the task
- (8) E_i refers to the energy consumption of T_i

4. Task Priority Allocation

This section proposes the WDAP. It mainly considers the three factors of task value, time urgency, and task consumption energy to dynamically adjust the task priority allocation.

4.1. Task Value. Task value is the basic attribute of the task, and it refers to the execution benefits that the task brings to the system. It can measure the importance of the task. Three key concepts are defined: cumulative income, average price density, and dynamic value density.

Definition 1. Cumulative value refers to the task value generated when the task is completed. It is recorded as I_i . Assuming that the task has executed t unit time, it can be obtained from

$$I_i = \frac{V_i \times t^p}{C_i^p}. \quad (1)$$

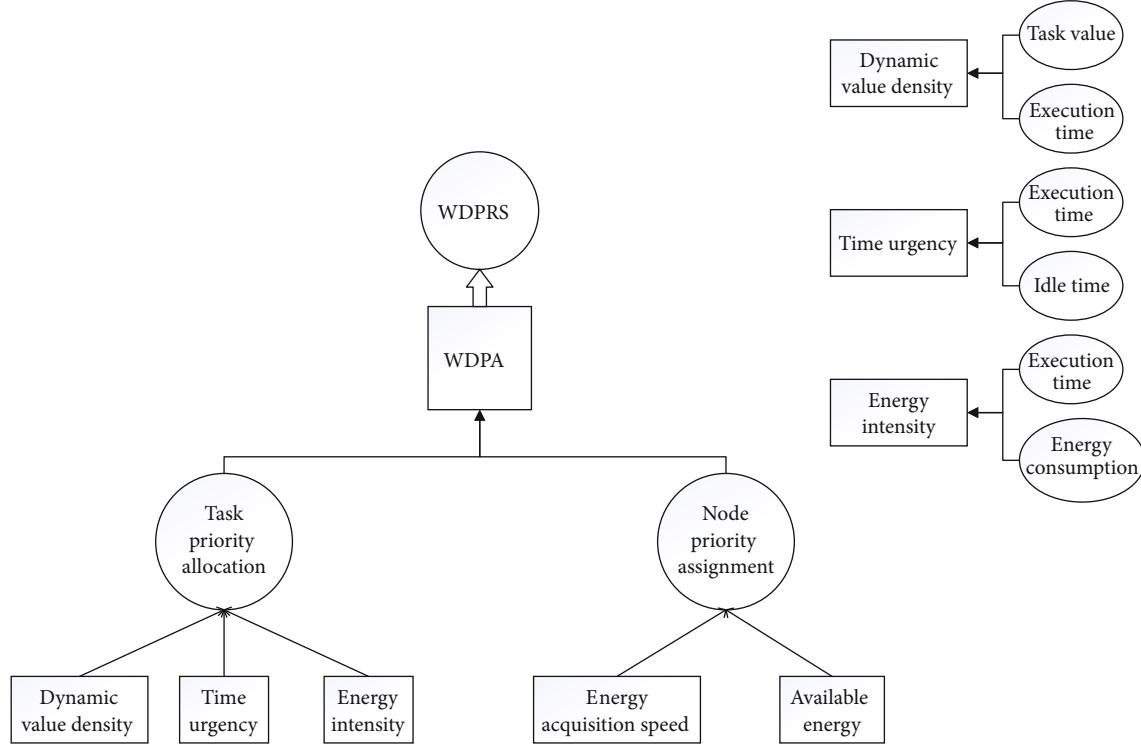


FIGURE 2: System overview. Here is the general flow of the WDRTS algorithm.

Among them, p is the influence parameter of dynamic value density on task priority. Note that before the task is successfully executed and submitted, the value generated by the task is only a simulated benefit, not an actual value gain. Task T_i can bring value gains of the system only after the task is successfully executed and committed. Otherwise, the system discards the part of the value.

Definition 2. Average value density refers to the ratio of task value to theoretical execution time, recorded as \bar{A}_i only related to the task itself. It can be obtained by

$$\bar{A}_i = \frac{V_i}{C_i}. \quad (2)$$

Definition 3. Dynamic value density refers to the ratio of the remaining task value to the remaining execution time, denoted as Y . Assuming that task T_i has executed t unit time, it can be obtained by

$$Y_i(t) = \frac{V_i - I_i(t)}{C_i - t} = \frac{(C_i^p - t^p)V_i}{C_i^p(C_i - t)}. \quad (3)$$

From equation (3), when $p = 1$, $Y_i(t) = \bar{A}_i$. Meanwhile, assuming that $0 \leq t \leq C_i$, $Y_i(t)$ is incremented with the parameter p ; assuming that $0 \leq t \leq C_i$, $p > 1$ and p are fixed values and $Y_i(t)$ is incremented with t . During the execution of the task T_i , the dynamic value density ranges from $[\bar{A}_i, p \times \bar{A}_i]$. It can be seen that as the execution time increases, the task priority of performing task allocation can be dynam-

ically increased, and the possibility that low-value tasks are preempted can be reduced.

4.2. Time Urgency. To improve the probability of successful execution while satisfying the task response before the deadline, the concept of execution density and urgency is defined.

Definition 4. Execution density, which is the ratio of the execution time required by the slack time of the task, is recorded as

$$\varepsilon_i = \frac{C_i - t}{d_i - \pi_i}. \quad (4)$$

Assuming that the task has executed t unit time, where π_i represents the current time, it can be obtained that in the slack time, the execution time of the task is larger and ε_i is larger. At the same time, ε_i has a value range of $(0, 1]$.

Definition 5. Urgency refers to the urgency of the successful execution of a task in order to meet the deadline before the deadline. It is recorded as δ and is obtained by

$$\delta_i(t) = q^{\varepsilon_i}, \quad (5)$$

where q is the parameter where the urgency affects the task priority. At the same time, the range of δ is $(0, 1]$. And when q ($q \geq 1$) is constant, the urgency increases with time. When $t = 0$, the urgency is the smallest, which is q^{C_i/D_i} . At a certain moment, the slack time just meets the time to complete the task. At this time, the urgency is q at most. Therefore, task

```

Let  $T_r$  be the executing task
Let  $Q$  be the queue of tasks including waiting tasks and active tasks
Begin
    if ( $T_i$  is  $T_r$  or  $T_i \in Q$ )// $T_i$  is a no-sleeping task
         $R(T_i) = Y_i(t) \times \delta_i(t) \times \omega_i$ 
    else
         $R(T_i) = 0$ 
End

```

ALGORITHM 1: Task priority allocation of WDAP.

T_i urgency interval is $[q^{C_i/D_i}, q]$. For tasks that are in progress, the urgency of the task T_i is the urgency of the most recent execution.

4.3. Task Energy Consumption. In the wirelessly powered sensing platform, not only the task value and time urgency but also the energy consumption of the task needs to be considered. The wirelessly powered sensing platform acquires the RF signal and converts it into its own energy. This solves the problem of frequent battery replacement in IoT nodes, but the lack of available energy can occur; the task of performing high-energy consumption may lead to insufficient available energy that causes the node to enter the sleep period in advance. Therefore, it is necessary to consider the energy consumed of the task as a factor that affects the priority of the task. This paper combines task energy consumption and execution time to propose a new evaluation index.

Definition 6. Energy consumption coefficient refers to the amount of energy consumed per unit time, denoted as f , which can be expressed as

$$f = \frac{E_i}{C_i}. \quad (6)$$

Definition 7. Energy intensity refers to the indicator that measures the energy consumption of real-time tasks in the remaining execution time, which is denoted as ω_i . Supposing that task T_i has been executed for t unit time, the energy intensity is obtained by

$$\omega_i = v^{-(E_i - ft)}, \quad (7)$$

where v is the influence parameter where the task energy consumption affects the task priority. It can be known that when $v \geq 1$, ω_i is in the range of $[v^{-E_i}, 1]$. The energy consumption intensity is related to the execution time. As the execution time increases, the energy consumption intensity increases. This means that as the execution time increases, the impact of energy consumption on the priority becomes lower. It can avoid wasting system resources caused by preemption.

4.4. Assign Task Priorities Dynamically. From the above, the dynamic value density is obtained according to the task value and the execution time. According to the execution time and the task deadline, the task urgency is obtained. Also, the energy intensity is obtained according to the task energy con-

sumption and the execution time. Based on these three aspects, the dynamic task priority allocation in WDAP is proposed.

Assume that π_i is the current time, task T_i has executed t unit time, and its dynamically assigned priority is recorded as $R(T_i)$, which can be obtained by

$$R(T_i) = Y_i(t) \times \delta_i(t) \times \omega_i \\ = \frac{(C_i^P - t^P)V_i}{C_i^P(C_i - t)} \times q^{(C_i - t)/(d_i - \pi_i)} \times v^{-(E_i - (E_i/C_i)t)}. \quad (8)$$

Among them, the influence of the task value, the urgency, and the task energy consumption on the task priority are dynamically adjusted by p, q, v . Prove that $R(T_i)$ has a value range of $[V_i/C_i \times q^{C_i/D_i} \times v^{-E_i}, p \times q \times V_i/C_i]$.

At the same time, the initial priority of the task T_i is recorded as $O(T_i)$, which can be obtained by

$$O(T_i) = \frac{V_i}{C_i} \times q^{C_i/D_i} \times v^{-E_i}. \quad (9)$$

When the real-time task system starts running, the tasks with high task priority get the execution right. As the execution time progresses, the priority of each task of the system changes. The urgency of the task during execution is equal to the urgency of the last execution, dynamic value density increases with time, and energy intensity increases with time, so its task priority will increase continuously. It can effectively protect tasks from being preempted by other tasks. For other tasks, as time goes by, the task deadline is getting closer, the urgency will be higher, and the priority will increase dynamically, which has the opportunity to seize the task. For tasks with higher energy consumption, the assigned priority is smaller, so as to reflect the impact of task energy consumption on task priority. At the same time, we can adjust p, q, v to dynamically adjust the impact of task value, execution urgency, and task energy consumption on task priority. If the value of p is relatively large, the task value will increase the influence weight of the task priority allocation. As the execution time increases, the value density will increase, reducing the possibility of being preempted by the task. If the value of q is relatively large, it will increase the influence weight of the task priority allocation and the possibility of task execution with low value density; if the v value is large, the impact weight of the task energy factor on the task priority allocation will increase. The energy consumption is

higher, and the priority of task dynamic allocation is smaller. At the same time, the impact of task energy consumption on task priority will become smaller, and the task priority will increase with execution time. It can reduce the possibility of being seized to avoid wasting energy.

In summary, the task priority allocation policy flow is shown in Algorithm 1.

5. Node Priority Allocation

Earlier, we discussed the allocation policy for task priority. However, for a wirelessly powered sensing system, how to assign tasks to each node is a problem we need to consider below. Therefore, in this section, a new policy for dynamically node priority allocation is proposed. It dynamically allocates node priority by considering available energy and energy acquisition speed. It is assumed that the set of wirelessly powered sensing system nodes is $\{S_1, S_2, \dots, S_n\}$, where the node with the most available energy is denoted as S_{\max} and the available energy of node S_i is denoted as en_i .

5.1. Available Energy. Energy is one of the important factors that affect the working operation of a node. The wirelessly powered sensing platform captures the RF signal and converts it into its own energy, thus getting rid of the battery. Nodes with more energy should try to assign tasks with higher priority to ensure the successful execution of the task. In order to know the available energy of the current node, it is necessary to establish a node energy model. In addition, in order to dynamically obtain the current energy, it is necessary to dynamically obtain the distance between the reader and the node. In this section, the concept of available energy intensity is used to measure the energy of each node.

5.1.1. Slack Time. Slack time refers to the maximum time that the real-time system can remain slack at the current time t without missing the task deadline, denoted as $s(t)$.

Assume that the slack time of the task T_i is the maximum slack time of the system after executing task T_i and higher priority tasks, which can be expressed as

$$s(T_i, t) = (d_i - t) - \sum_{D_j \leq D_i} C_j. \quad (10)$$

Finally, the slack time of all tasks is compared to select the smallest one, as the system slack time, represented by

$$s(t) = \min(s(T_i, t)). \quad (11)$$

5.1.2. Received Energy Model. Because the wirelessly powered sensing platform captures energy primarily through the antenna, it is ultimately stored in a capacitor to provide the energy. The ideal energy transfer model is represented by the Friis formula [30]:

$$P_R = P_r G_r G_R \left(\frac{\lambda}{4\pi r} \right)^2, \quad (12)$$

```

Let S be the node set in our system
Begin
  if ( $S_i \in S$ )
     $N(S_i) = k_1 \alpha_i + k_2 \Gamma_i$ 
  else
     $N(S_i) = 0$ 
end

```

ALGORITHM 2: Node priority allocation of WDAP.

where P_R is the received power, P_r is the transmitter power, G_r is the transmitter gain, G_R is the receiver gain, λ is the incident wavelength, and r is the receiver-to-node distance.

In the real environment, due to the limitation of the communication distance of the node, there are problems such as electromagnetic wave scale attenuation and multipath effect. Therefore, the Friis formula is not suitable for these cases, and the formula needs to be processed.

This paper introduces the fading factor θ and the loss factor ξ , and the modified formula is

$$P_R = \theta P_r G_r G_R \left(\frac{\lambda}{4\pi(r - \xi)} \right)^2. \quad (13)$$

The two parameters, the fading factor θ and the loss factor ξ , can be obtained by collecting power values at different positions in the experimental environment.

In the real-time system, the node enters a sleep state during slack time and completes the energy capture process. Thus, at the beginning of the current time t , the energy collected during the slack time is represented by the following equation.

$$W_R = P_R s(t) = \theta P_r G_r G_R \left(\frac{\lambda}{4\pi(r - \xi)} \right)^2 s(t). \quad (14)$$

Through formula (14), the currently available energy of each node can be obtained.

5.1.3. Dynamic Acquisition Distance. Through the above research, the corrected energy model can be obtained, but it is necessary to know the distance between the receiver and the node to know the power, thereby obtaining the captured energy. If you need to dynamically understand the changes in the captured energy, you need to dynamically obtain the distance in the energy model.

The receiver obtains the value of the node signal strength and establishes a model of the signal strength. So the distance between the node and the receiver is represented by

$$r = 10^{(\text{abs}(rssi) - A)/(10n)}, \quad (15)$$

where $rssi$ is the RSSI value of the receiver signal strength, $\text{abs}(rssi)$ is the absolute value of the RSSI value, A is the signal strength when the node is separated by one meter from the receiver, and n is the environmental weakening factor.

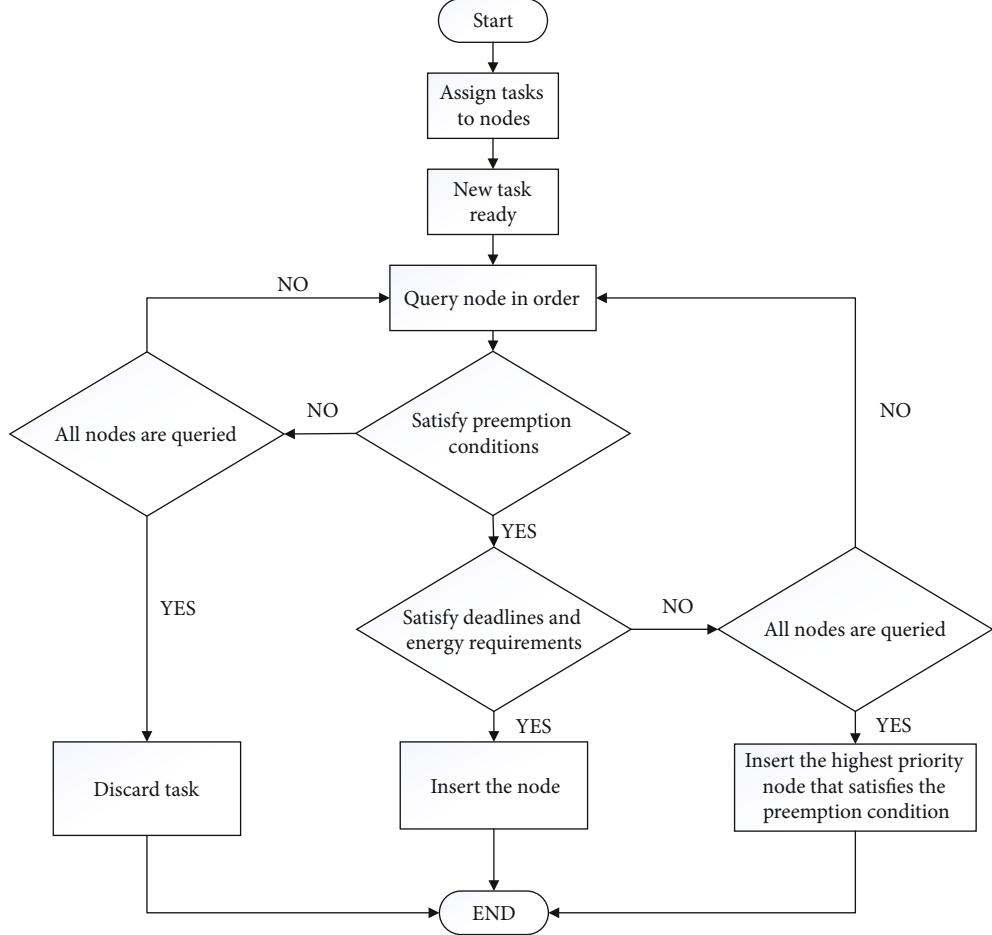


FIGURE 3: According to the process, the ready task is assigned to each node.

For the value of the parameter, different values of RSSI can be obtained at different positions in the experimental environment to obtain n . Finally, the distance can be dynamically obtained from the obtained RSSI value, so that the node energy model can be dynamically updated to accurately obtain the available energy.

5.1.4. Available Energy Intensity

Definition 8. Available energy intensity refers to the ratio of the available energy of the node to the available energy of S_{\max} , denoted as α .

Assuming that the node is S_i , the available energy intensity of node S_i is given by

$$\alpha_i = \frac{\text{en}_i}{\text{en}_{\max}}. \quad (16)$$

The available energy intensity of each node is obtained by comparing the amount of available energy to each node. The value of α_i ranges from $(0, 1]$. It can be seen that the available energy of the node is larger, the priority of the node assigned is higher, and the task can be better successfully executed.

5.2. Energy Acquisition Speed. Because energy is one of the important reasons that affect the work of the node. Comparing the available energy of each node is very important, and energy acquisition speed will also affect the endurance of the node. Therefore, the energy acquisition speed is also one of the factors that affect the priority of the node.

Since the energy acquisition speed can be expressed in terms of power, for this reason, we use equation (17) to represent the energy acquisition speed. Assuming that the node closest to the receiver is $S_{G \max}$, it can be known that $S_{G \max}$ is the node that gets the fastest energy acquisition speed. We propose a concept of energy intensity to measure the impact of energy harvesting on node priority.

Definition 9. Energy acquisition intensity is the ratio of the speed at which the node acquires energy and the speed at which $S_{G \max}$ obtains energy, denoted as Γ .

Assuming that the node is S_i , the energy intensity of node S_i can be obtained by

$$\Gamma_i = \left(\frac{r_{G \max} - \xi}{r_i - \xi} \right)^2. \quad (17)$$

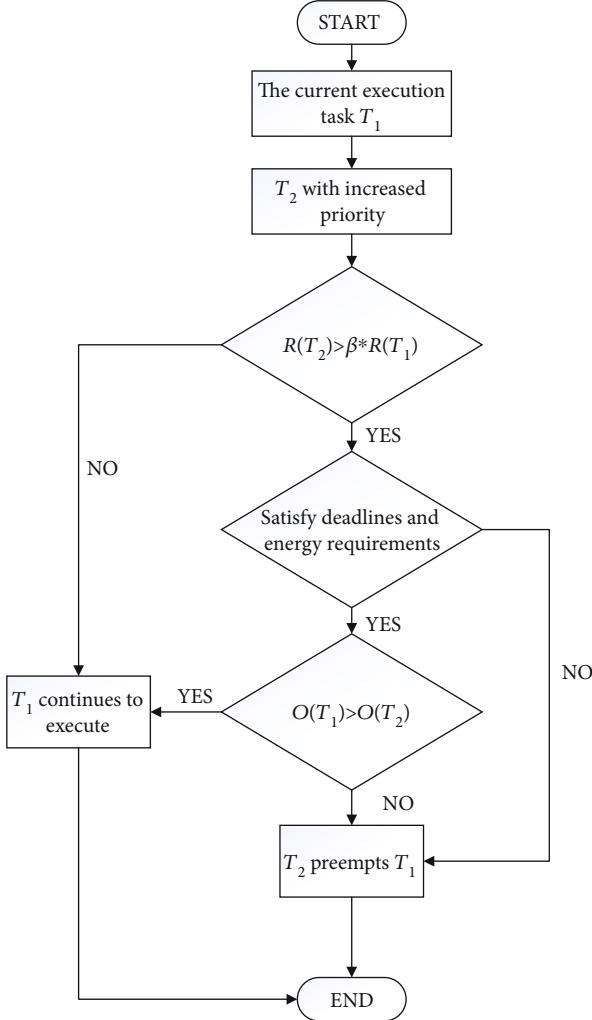


FIGURE 4: According to the process, it can be reasonable preemption of execution rights.

5.3. Node Priority Allocation. For a real-time system, each node can be regarded as a small independent real-time system, which can perform tasks in real time. Therefore, a dynamic node priority allocation policy is established, and tasks are allocated to each node. Based on the combination of available energy and energy acquisition speed, the dynamically assigning node priorities are proposed.

Assuming that the node is S_i , the node priority assigned by node S_i is

$$N(S_i) = k_1 \alpha_i + k_2 \Gamma_i. \quad (18)$$

Among them, k_1 and k_2 are the node priority impact factors, which can dynamically adjust the influence of the available energy and the acquiring energy speed on the dynamical node priority allocation policy. When the real-time system starts running, the nodes are dynamically assigned priority, and the nodes with the highest priority are assigned tasks with higher priority, which better guarantees the execution of tasks. As the available energy changes, the priority assigned to the node will also be different, and the node with

the lower initial priority will increase in priority, thereby assigning task execution. By dynamically changing the position of the node, its energy acquisition speed will change, resulting in a change in the energy acquisition intensity, which in turn changes the node priority. By changing the value of k_1 , k_2 also changes the node priority allocation. In general, the energy acquisition speed affects the amount of energy available, which in turn affects the task allocation. In order to meet the actual demand, k_1 will be bigger than k_2 . However, k_1 and k_2 can be dynamically adjusted to the degree of influence of change of the available energy and the energy acquisition speed on node priority allocation. Generally, according to experience, set $k_1 = 1$ and $k_2 = 3.25$.

In summary, the node priority allocation policy flow is shown in Algorithm 2.

6. Dynamic Scheduling Algorithm WDRTS

In the above, we proposed WDAP, which includes the dynamic task priority allocation policy and dynamic node priority allocation policy. Here, based on WDAP, we propose a real-time task scheduling policy based on wirelessly powered sensing platform named WDRTS.

6.1. System Bumps. In a real-time system, if there are two or more tasks whose priorities are alternately rising and repeatedly grabbing the execution rights of the tasks, it is called system bumps. The system bumps can increase system overhead and consume system resources. Based on our proposed dynamic task priority allocation priority, the priority between tasks will change dynamically with time, which may cause the system bumps. In order to limit this phenomenon, this paper sets a bump coefficient β to avoid preemption due to a small difference in priority.

Assuming that the current time is π_i , T_1 is the execution task and t_1 unit time is executed and T_2 is the waiting task and t_2 unit time is executed. If T_2 can preempt T_1 , then you need to satisfy the following formula:

$$R(T_2) \geq \beta \times R(T_1). \quad (19)$$

Prove that after T_2 can preempt T_1 , x times units are executed. At this time, the priorities of T_1 and T_2 are $R'(T_1)$ and $R'(T_2)$, respectively. To avoid bumps, the following must be satisfied:

$$R'(T_1) < \beta \times R'(T_2). \quad (20)$$

Assuming that the current time is π_i , x is small enough. In extreme cases, T_1 is the task of execution, the dynamic value density reaches the maximum value $p \times \bar{A}_1$, the urgency reaches the minimum value q^{C_1/D_1} , and the energy intensity is $v^{E_1((t_1-C_1)/C1)}$; T_2 is the waiting task, the minimum of its dynamic value density is \bar{A}_2 , the execution urgency is also minimal q^{C_2/D_2} , and the energy intensity is $v^{E_2((t_2-C_2)/C2)}$. Then, T_2 preempts T_1 , T_1 urgency gradually increased, the average price density is unchanged, and the energy intensity is unchanged; T_2 average price density is unchanged, the

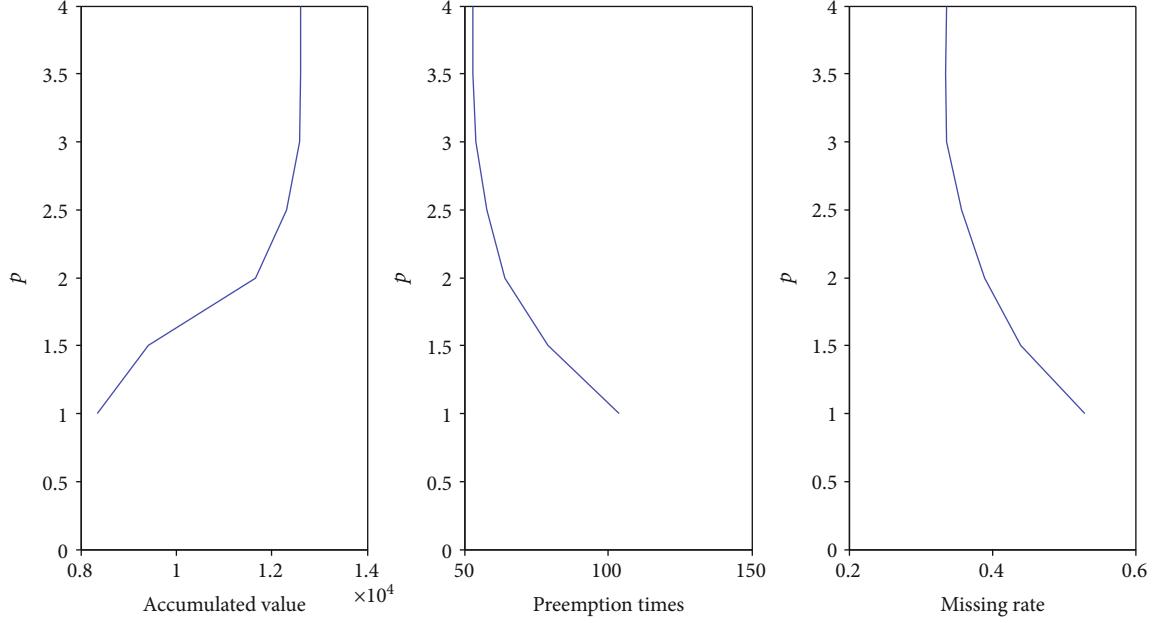


FIGURE 5: Performance analysis of parameter p . It can be seen from the effect of parameter p on the performance of the algorithm.

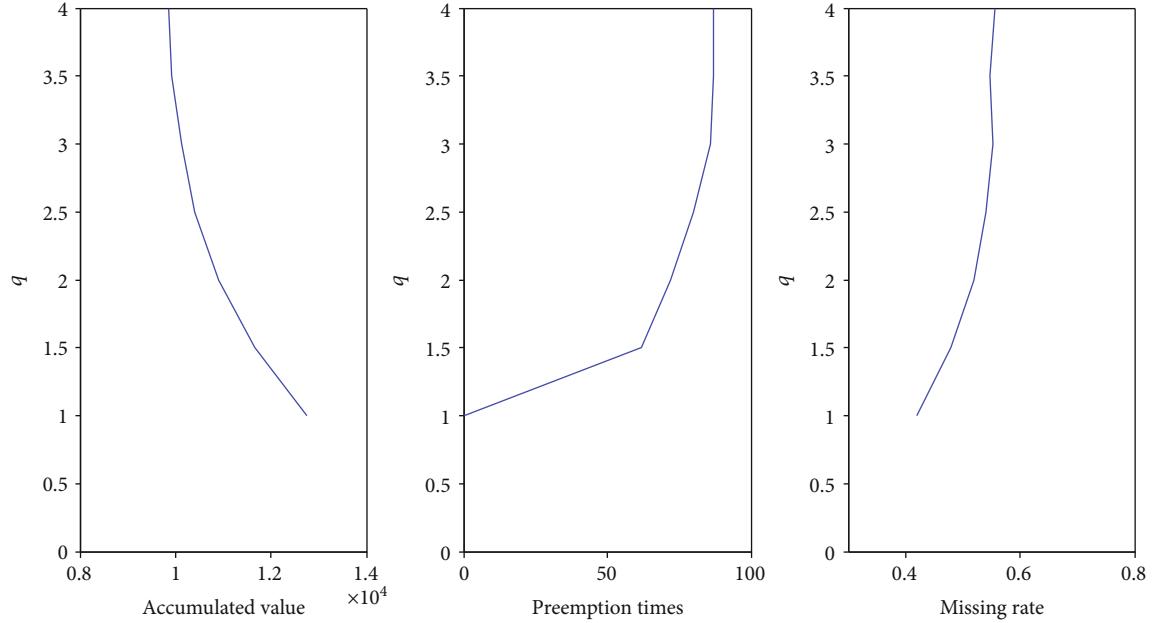


FIGURE 6: Performance analysis of parameter q . It can be seen from the effect of parameter q on the performance of the algorithm.

urgency gradually increases, and the energy intensity gradually increases. At time $\pi_i + x$, the urgency of T_1 reaches the maximum q , and it is assumed that the increase in the dynamic value density of T_2 is small enough to be ignored. According to formula (8), there is obviously

$$\beta \geq \max \left(\frac{R'(T_1)}{R'(T_2)} \right) \geq \frac{\max(R'(T_1))}{\min(R'(T_2))} = \frac{\overline{A}_1 \times p \times q}{\overline{A}_2 \times q^{C_2/D_2} \times v^{-E_2}}. \quad (21)$$

At the π_i time, T_1 are tasks; it is easy to get $\overline{A}_1 > \overline{A}_2$, $C_2 < D_2$, $v^{-E_2} < 1$, and therefore $\beta > 1$. Therefore, when $\beta > 1$ is established, bumps can be avoided to some extent.

6.2. WDRTS Scheduling Algorithm. In the wirelessly powered sensing platform, calculate the dynamic priority of each task and the node priority, assign the task to the appropriate node, and formulate the corresponding policy, so that the system obtains better performance. The algorithm is called WDRTS.

The algorithm is divided into two parts. The first part is how to send the task to each node reasonably, as shown in

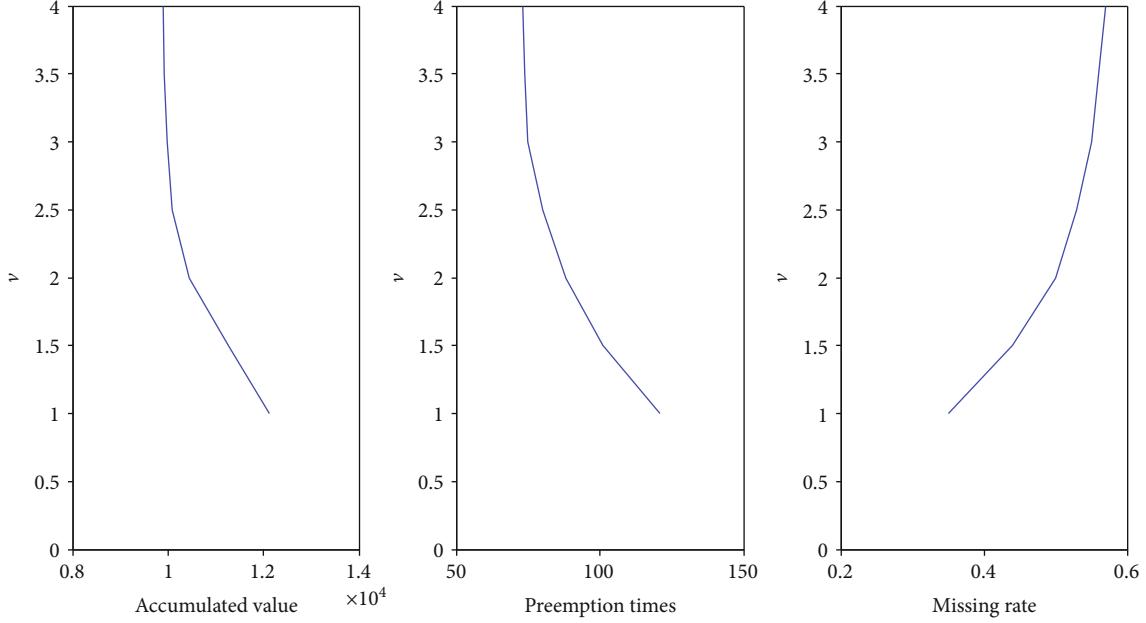


FIGURE 7: Performance analysis of parameter v . It can be seen from the effect of parameter v on the performance of the algorithm.

TABLE 2: Task map. Task properties used in simulation experiments.

Task	C	D	S	V	Energy consumption
T_1	9	11	16	60	22
T_2	7	8	11	80	31
T_3	2	7	8	40	16
T_4	5	6	9	30	15

Figure 3. The second part is how to perform the task scheduling algorithm inside a single node, as shown in Figure 4.

The steps of the WDRTS algorithm are as follows:

Step 1. The algorithm starts running, the reader assigns the ready task to the nodes of different priorities according to the priority of the task, the highest task priority assigns the highest priority node, and so on.

Step 2. The new task T_i is ready, and the task T_i in the ready state starts the query from the highest priority of the node. If the priority $R(T_i) \geq \beta \times R(T_k)$, the preemption condition is satisfied ($R(T_k)$ is the priority of the highest priority task T_k in the query node), find the wirelessly powered sensing platform where the task T_k is located, and judge whether the wirelessly powered sensing node is capable of performing the energy of all its tasks. If the energy is sufficient to complete the task T_i and T_k and their deadline in the wirelessly powered sensing platform is not delayed, the task T_i preempts the task T_k ; otherwise, continue to query the low-priority node to calculate whether the preemption condition, energy requirement, and deadline requirement are met until the requirement is met. If the requirements are not met, the node that satisfies the preemption requirement for the first time in step 2 is found, and the task T_i is inserted.

Step 3. Within a single wirelessly powered sensing platform, assuming that the current time is τ_i , T_1 is the execution task and t_1 unit time has been executed. At this time, the pri-

ority of T_1 is $R(T_1)$, which guarantees that the basic condition of the deadline is $d_1 - \tau_i \geq C_1 - t_1$. T_2 is the highest priority task among all active tasks and waiting tasks and has executed t_2 time units. At this time, the priority of T_2 is $R(T_2)$, which guarantees that the basic condition of the deadline is $d_2 - \tau_i \geq C_2 - t_2$; then, $R(T_2) \geq \beta \times R(T_1)$, if $d_2 - \tau_i - (C_1 - t_1) \geq C_2 - t_2$ and $d_1 - \tau_i - (C_2 - t_2) \geq C_1 - t_1$, indicating that if T_2 waits for T_1 to commit and starts execution, it can still guarantee deadline of T_2 , and if T_2 preempts T_1 , T_1 waits T_2 , will continue to execute after submission, and will not miss the deadline, and the energy guarantee can perform two tasks, which are divided into two cases:

- (1) If task T_2 has a lower initial priority, no preemption is performed, unnecessary context switching between tasks is reduced, and system efficiency is improved
- (2) Otherwise, it will preempt T_1 and T_1 will wait for T_2 to submit

Step 4. If the deadline is not met or the energy is not enough to support the completion of both tasks, then it is normally preempted according to the priority order.

When the energy consumption of the node is exhausted, set the deceleration factor, lengthen the execution cycle of each task, and increase the sleep time of the node so that it can charge as much as possible.

7. Results and Discussion

The experiments were all simulated on a PC. To analyze the performance of the policy, this paper compares the WDRTS with GEDF, HVF, and DVD algorithms to evaluate the pros and cons of the policy. Because the wirelessly powered sensing platform has limited hardware conditions and conduct

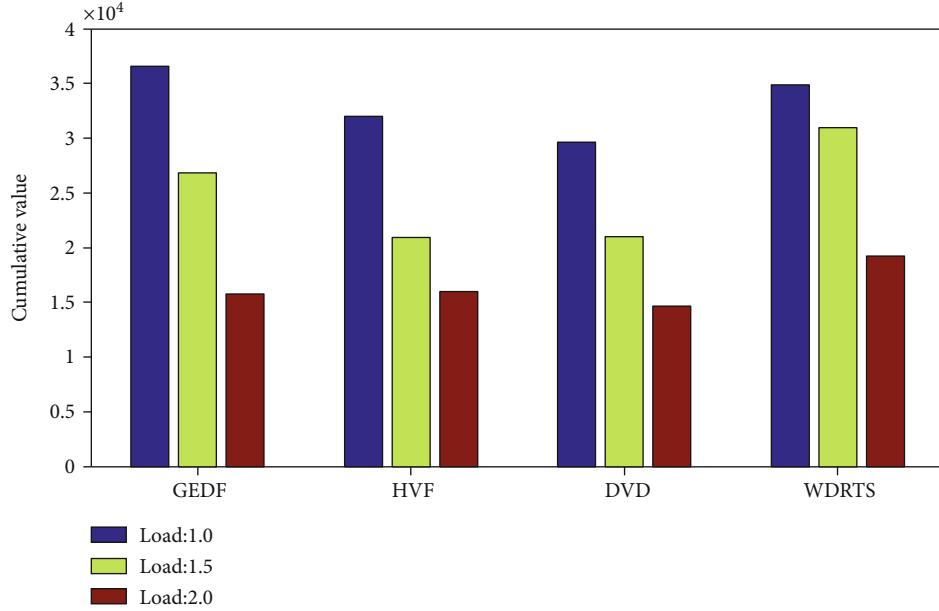


FIGURE 8: Experimental results of the cumulative value comparison of the four algorithms.

the experiment on the wirelessly powered sensing platform in the next step, we choose GEDF, HVF, and DVD for comparative experiments. These algorithms are relatively mature and easy to implement, and the required resources are not high. At the same time, these algorithms are the most classic algorithms and derived from those. So these algorithms are more suitable for running on the wirelessly powered sensing platform. In the experiment, the cumulative value, the missed deadline rate, and the preemption times were used to evaluate the pros and cons.

The cumulative value is the total value obtained by the system after the real-time task is successfully executed.

The missed deadline rate is the probability that the real-time task will not be successfully executed because the deadline is missed.

The preemption times is the number of real-time tasks that preempt each other.

7.1. Comparison of Parameters

7.1.1. Performance Analysis Experiment of Parameter p . First, the parameters $q = 2$ and $v = 1.5$ are fixed, and the value of the parameter p is changed to analyze the influence of the parameter p on the performance of the algorithm. According to Figure 5, as p increases, the impact of task dynamic value density on task priority increases and the impact of task urgency and energy intensity on priority is relatively reduced, so tasks with high dynamic value density receive priority response. It is not easy to be preempted, which greatly improves the successful execution probability and the cumulative value will increase accordingly. At the same time, q, v do not change, and the preemption times increase as p increases, making the execution task difficult to be preempted. When p is large enough, the impact on task priority is also large enough; therefore, there is no significant increase in priority.

7.1.2. Performance Analysis Experiment of Parameter q . First fix the parameters $p = 2$ and $v = 1.5$, change the value of the parameter q , and analyze the influence of the parameter q on the performance of the algorithm. As q increases, the impact of task dynamic value density and energy intensity on task priority is relatively reduced and the impact of task urgency on priority is relatively increased, so tasks with high urgency are given priority response. As shown in Figure 6, the cumulative value decreases as q increases. Moreover, as the urgency of nonexecutive tasks gradually increases, it is easy to cause the priority to rise alternately, the preemption times are greatly increased, and the frequency of missed deadlines is greatly increased.

7.1.3. Performance Analysis Experiment of Parameter v . First, the parameters $p = 2$ and $q = 2$ are fixed and the value of the parameter v is changed to analyze the influence of the parameter v on the performance of the algorithm. As v increases, the impact of task dynamic value density and urgency on task priority is relatively reduced and the impact of energy intensity on priority is relatively increased; therefore, tasks with low-energy consumption get priority response. As shown in Figure 7, the cumulative value decreases as v increases. Moreover, as the task with low-energy consumption gets priority response, the use of system energy is moderated, the tag sleep time is reduced, the work cycle is longer, the impact of urgency is reduced, and the number of preemption is reduced. The impact of urgency is reduced, and the frequency of missed deadlines increases.

7.2. Experimental Comparison. First, the WDRTS was compared with the GEDF, HVF, and DVD algorithms under different loads. The load represents the utilization of the processor by the task set. Note that the values of these three indicators are the average of 50 task sets. The attributes of

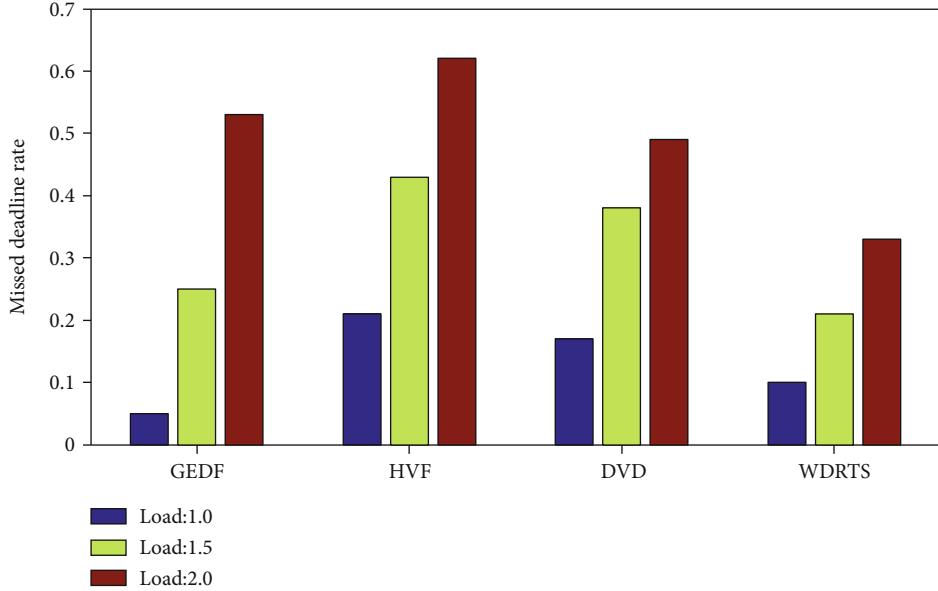


FIGURE 9: Experimental results of the missed deadline rate comparison of the four algorithms.

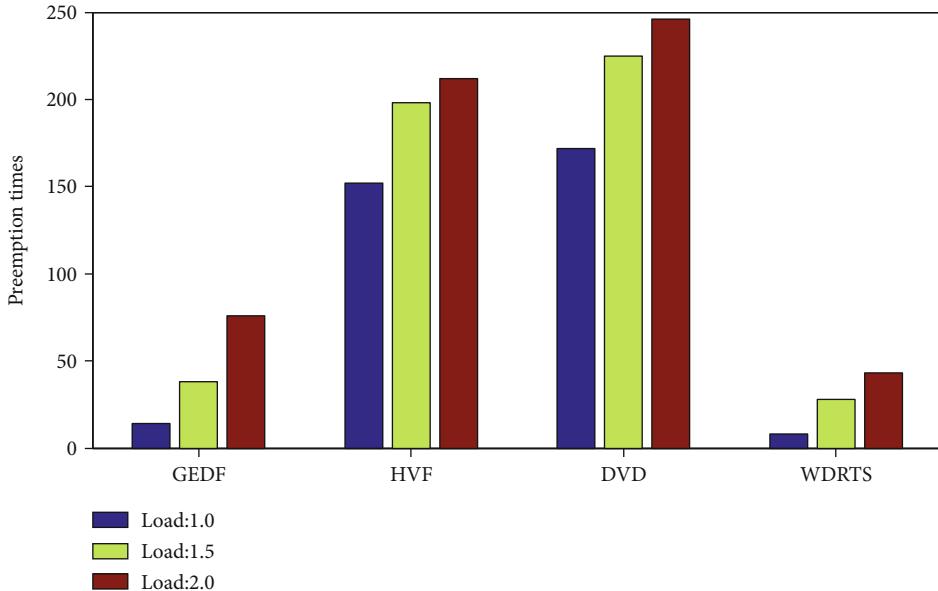


FIGURE 10: Experimental results comparing the preemption times of the four algorithms.

each task set are shown in Table 2. The load represents the utilization of the processor by the task set.

7.2.1. The Experiment of the Cumulative Value. In Figure 8, we can see that the cumulative value obtained by different loads is different. The load is larger, and the cumulative value obtained is smaller. At the same time, when the load is low, WDRTS has no advantage in comparing the cumulative value of GEDF. When the load increases, WDRTS obtains more cumulative value than the other three algorithms, indicating that WDRTS gets the highest task value gains when resources are limited and the system is overloaded.

7.2.2. The Experiment of the Missed Deadline Rate. In Figure 9, we can see that GEDF has the lowest missed deadline rate when the load is 1.0; when the load increases, it compares with GEDF, HVF, and DVD algorithms, and the missed deadline rate of WDRTS is the lowest among the four algorithms. It is shown that in the case of overload, WDRTS is not easy to miss the task deadline, and the probability of successful execution of the task is the largest.

7.2.3. The Experiment of the Preemption Times. In Figure 10, we can see that WDPRS has lower preemption times than other algorithms and is superior to other algorithms.

In general, the WDRTS can get the best cumulative value under high load, and the number of preemption times and the missed deadline rate are also the lowest. Because the cumulative value and the missed deadline rate directly reflect the probability of successful completion of real-time task execution, the preemption times reflect the high efficiency of task execution, so the schedulability of the WDRTS is the best among these algorithms.

8. Summary

In this paper, from the perspective of low available energy of the wirelessly powered sensing system and the system being easy to overload, a dynamic task priority allocation policy in WDAP is proposed based on the three factors: task value, time urgency, and task energy consumption. By combining the two factors of available energy and energy acquisition speed, the dynamic allocation of node priorities for WDAP is proposed. At the same time, based on WDAP, the WDRTS based on priority preemption is proposed. The algorithm dynamically adjusts the influence of the task value, the urgency of execution, and the task energy consumption by adjusting p, q, v , which improves the flexibility of the algorithm. In addition, this article discusses the condition and mitigation of system bumps. The experiment proves that the proposed WDRTS algorithm can reduce the number of preemption, reduce the probability of miss deadline, and increase the cumulative value of the system under the condition of high load. Besides, due to the current hardware reasons, the task scheduling algorithm developed in this paper needs to further understand the characteristics of the wirelessly powered sensing platform with software simulation, so as to design its own wirelessly powered sensing platform and prepare for the next step of the full operation of the working platform.

Data Availability

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

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant number 61971248), the Ningbo Science and Technology Bureau (grant number 2016D10008), the Ningbo Key Science and Technology Plan (2025) Project (grant numbers 2018B10075, 2019B10125, and 2019B10028), and the Key Science and Technology Projects of Zhejiang Province (grant number 2020C03064).

References

- [1] R. Du, M. Xiao, and C. Fischione, “Optimal node deployment and energy provision for wirelessly powered sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 2, pp. 407–423, 2019.
- [2] J. Hui, C. Jing, Z. Jumin, and L. Dengao, “The research on energy management and communication optimization in passive sensing system,” *Microelectronics & Computer*, vol. 36, no. 12, pp. 42–47, 2019.
- [3] X. Zhu, L. Jun, and Z. M. Chu, “Optimal deployment of energy-harvesting directional sensor networks for target coverage,” *IEEE Systems Journal*, vol. 13, no. 1, pp. 377–388, 2019.
- [4] R. Garg and N. Garg, “Energy management in a multi-source energy harvesting IoT system,” *Journal of Information Technology Research*, vol. 13, no. 2, pp. 42–59, 2020.
- [5] P. Prakasam, T. R. Suresh Kumar, T. Velmurugan, and S. Nandakumar, “Efficient power distribution model for IoT nodes driven by energy harvested from low power ambient RF signal,” *Microelectronics Journal*, vol. 95, article 104665, 2020.
- [6] M. Sun, Z. Zhou, J. Wang, D. Chu, and W. Gaaloul, “Energy-efficient IoT service composition for concurrent timed applications,” *Future Generation Computer Systems*, vol. 100, pp. 1017–1030, 2019.
- [7] J. Teizer, “Wearable, wireless identification sensing platform: self-monitoring alert and reporting technology for hazard avoidance and training (smart hat),” *Electronic Journal of Information Technology in Construction*, vol. 20, no. 5, pp. 295–312, 2015.
- [8] J. Lu, *Research on Intelligent Sensing of Humans and Commodities in Warehouse Based on Passive RFID*, Southeast University, 2018.
- [9] M. Peng, W. Liu, T. Wang, and Z. Zeng, “Relay selection joint consecutive packet routing scheme to improve performance for wake-up radio-enabled WSNs,” *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 7230565, 32 pages, 2020.
- [10] A. Karthikeyan, V. P. Arunachalam, and S. Karthik, “Performing data assessment in terms of sensor node positioning over three dimensional wireless sensor network,” *Mobile Networks and Applications*, vol. 24, no. 6, pp. 1862–1871, 2019.
- [11] L. Wang, L. Zhao, G. Luo et al., “System level design of wireless sensor node powered by piezoelectric vibration energy harvesting,” *Sensors and Actuators A Physical*, vol. 310, article 112039, 2020.
- [12] B. Yuan, D. Chen, D. Xu, and M. Chen, “Conceptual model of real-time IoT systems,” *Frontiers of Information Technology & Electronic Engineering*, vol. 20, no. 11, pp. 1457–1464, 2019.
- [13] J. Xiaowen, *Multiprocessor Real-Time Systems Scheduling for Energy and Reliability Optimization*, Zhejiang University, 2018.
- [14] F. S. Saadatmand, N. Rohbani, F. Baharvand, and H. Farbeh, “TAMER: an adaptive task allocation method for aging reduction in multi-core embedded real-time systems,” *The Journal of Supercomputing*, vol. 77, no. 2, pp. 1939–1957, 2021.
- [15] X. Ma, R. Guobin, and X. Huahu, “Research on task scheduling in cloud computing,” *Computer Science*, vol. 46, no. 3, pp. 1–8, 2019.
- [16] X. Jiali, C. Hui, and Y. Bing, “A dynamic priority real-time task scheduling algorithm,” *Chinese Journal of Computers*, vol. 35, no. 12, pp. 2685–2695, 2012.
- [17] R. Shigeta, T. Sasaki, D. M. Quan et al., “Ambient RF energy harvesting sensor device with capacitor-leakage-aware duty

- cycle control,” *IEEE Sensors Journal*, vol. 13, no. 8, pp. 2973–2983, 2013.
- [18] C. L. Liu and J. W. Layland, “Scheduling algorithms for multiprogramming in a hard-real-time environment,” *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.
- [19] P. Chandra Sekhar, P. S. Avadhani, P. Suresh Varma, and N. R. M. R. Bhargava, “Four stage scheduling of steel making using earliest deadline first algorithm,” *International Journal of Scientific Engineering and Technology*, vol. 6, no. 1, pp. 1–6, 2017.
- [20] H. Meiling, Q. Deng, Z. Tianyu, F. Zw, and L. Yh, “Limited preemption for GEDF scheduling on multiprocessors,” *Chinese Journal of Computers*, vol. 42, no. 11, pp. 2355–2367, 2019.
- [21] S. Semghouni, L. Amanton, B. Sadeg, and A. Berred, “On new scheduling policy for the improvement of firm RTDBSs performances,” *Data & Knowledge Engineering*, vol. 63, no. 2, pp. 414–432, 2007.
- [22] H. Jin, H. A. Wang, Q. Wang, and G. Z. Dai, “An improved least-slack-first scheduling algorithm,” *Journal of Software*, vol. 15, no. 8, pp. 1116–1123, 2004.
- [23] C. Hui, “A dynamic priority allocation policy for real-time task,” in *International Conference on E-health Networking*, Shenzhen, Guangdong, China, 2010.
- [24] S.-L. Wang and H.-T. Hong, “The roles of collective task value and collaborative behaviors in collaborative performance through collaborative creation in CSCL,” *Educational Technology Research and Development*, vol. 66, no. 4, pp. 937–953, 2018.
- [25] E. Camponogara, A. B. De Oliveira, and G. Lima, “Optimization-based dynamic reconfiguration of real-time schedulers with support for stochastic processor consumption,” *IEEE Transactions on Industrial Informatics*, vol. 6, no. 4, pp. 594–609, 2010.
- [26] H. Chniter, O. Mosbahi, M. Khalgui, M. Zhou, and Z. Li, “Improved multi-core real-time task scheduling of reconfigurable systems with energy constraints,” *IEEE Access*, vol. 8, no. 99, pp. 95698–95713, 2020.
- [27] X. Wang, I. Khemaissa, M. Khalgui, Z. W. Li, O. Mosbahi, and M. C. Zhou, “Dynamic low-power reconfiguration of real-time systems with periodic and probabilistic tasks,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 258–271, 2015.
- [28] R. Wang, *Research on Real-Time Sensing Task Scheduling Method in WISP System*, Taiyuan University of Technology, 2018.
- [29] R. Guo, *Research on Energy Optimization and Mission Planning in WISP System*, Taiyuan University of Technology, 2018.
- [30] J. A. Shaw, “Radiometry and the FRIIS transmission equation,” *American Journal of Physics*, vol. 81, no. 1, pp. 33–37, 2013.