

RESEARCH

Open Access



# Uncertainty-aware workflow migration among edge nodes based on blockchain

Zhanyang Xu<sup>1,2</sup>, Qingfan Geng<sup>1,2</sup>, Hao Cao<sup>1,2</sup>, Chuanjian Wang<sup>3</sup> and Xihua Liu<sup>1,2\*</sup>

## Abstract

Workflow is one of the most typical applications in distributed computing, which makes a variety of complex computing work orderly. However, assigning workflow tasks to nodes in the process of multi-node collaboration is still a challenge, because there are some unpredictable emergencies, i.e., uncertainty, in the process of workflow scheduling. The paper proposes a blockchain-powered resource provisioning (BPRP) method to solve the above problems. Technically, we use the directed acyclic graph in the graph theory to represent the workflow task and optimize the workflow scheduling strategy in the presence of uncertainty. The processing time and energy consumption of workflow tasks are also optimized by using non-dominated sorting genetic algorithm III (NSGA-III). Finally, we carry out experimental simulations to verify the effectiveness of the proposed method.

**Keywords:** Blockchain, Uncertainty-aware, Edge computing, Workflow, NSGA-III

## 1 Introduction

### 1.1 Background

With the advent of 5G era, various new applications of mobile Internet emerge in endlessly, the basic framework of smart city has been further improved, and people's intelligent living standards have also been greatly improved. The operation of a city produces a large number of data [1], and the organization of the facts carried by these data forms valuable information products and promote the development of the city in an efficient, convenient, and low-carbon direction. The information resource center is the cornerstone of the construction of smart cities [2, 3]. At the same time, the frequent use of mobile devices consumes a lot of bandwidth resources, and the processing efficiency of local mobile devices has been unable to meet the high-quality services required by users [4–7] [8]. Therefore, computing tasks are migrated to the cloud data center (CDC) for execution.

Cloud computing divides huge computing processing programs into innumerable smaller subroutines automatically through the network and then submits them to a

system composed of multiple servers [9]. After processing by searching, computing, and analysis, the results are returned to users [10]. In the cloud platform, users use computing resources in the form of the lease, and computing resources are provided in the form of virtual machines (VMs) [11]. Users perform migrated computing tasks by accessing the shared pool of computing resources in the cloud. Nevertheless, due to the long distance between mobile devices and the cloud, cloud computing will cause data transmission delay [12] that cannot be ignored when it is used, which brings a lot of problems to delay-sensitive networks in smart cities [13].

In order to meet the requirements of low latency and high bandwidth in the 5G era, cloud computing and Mobile Edge Computing (MEC) are integrated; that means the cloud platform is migrated from the inside of the mobile network to the edge of the mobile network to realize the flexible utilization of computing resources. The integration of cloud computing and MEC make full and reasonable use of network computing resources [14], thus improving the quality of service and user experience.

In order to achieve orderly and efficient utilization of computing resources, the workflow is introduced to simulate scientific applications requiring huge computing resources [15, 16]. Workflow refers to a kind of business process that can be fully automated. According to a series of different pre-set rules, information or computing

\*Correspondence: liuxihua710@gmail.com

<sup>1</sup>School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China

<sup>2</sup>Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, China

Full list of author information is available at the end of the article

tasks are transmitted and executed among different executives. Workflow is represented by data as interrelated computing tasks [17, 18]. The process of mapping computing tasks in a workflow to computing resources in the cloud resource sharing pool is called workflow scheduling. Workflow scheduling problems are divided into two categories: independent task scheduling and interdependent task scheduling. In the former, a task is an independent entity and does not depend on other tasks. In the latter, there is a dependency relationship between a group of tasks, which means that in a workflow, only when the parent task is completed can the task begin to execute, which is relatively complex. In the paper, the above two cases are considered at the same time.

In the process of workflow scheduling, there are some unexpected situations, such as resource conflicts, node failures and so on, namely uncertainty [19]. Uncertainty makes the initial static scheduling strategy unable to work properly, which brings great challenges to workflow scheduling. In order to ensure that tasks are completed on time, a new real-time dynamic scheduling strategy is needed to deal with the affected tasks. When the task is dependent on other tasks, the scheduling of its subtasks should also be considered.

## 1.2 Motivation

By using the appropriate allocation method, the workflow system can select the most suitable participants to perform tasks, thus improving the efficiency of workflow operation and quality of service [20]. Traditional business workflow is usually driven by control flow, which is used to specify paths and constraints, and may require manual intervention [21]. In the paper, we use the permissioned private blockchain to record the real-time status of each edge node in the system and the workflow scheduling scheme used in the current system. The essence of blockchain is a distributed public account, which is checked by edge nodes in the system [22]. All the edge nodes which participated in the blockchain system maintain the updating of the account book. The account book is modified in accordance with strict rules and consensus. The linkage mechanism of blockchain makes it multi-consensus, record traceability, and non-tampering. This makes the state of an edge node to be broadcast to other nodes in the system in real-time and reliably, thus making the workflow scheduling strategy more efficient and intelligent [23].

In the blockchain, each task corresponds to a block. The blockchain consists of some connected blocks. Each block contains two parts: block head and block body. The block head records the meta-information of the current block, such as the generation time, the hash value of the current block, and the hash value of the previous block, and the block body record task's actual data. The effectiveness of

each block is verified by calculating its own hash value and the hash value of the previous block. When a task block is generated, all edge nodes compete for the right to record the block [24]. The successful node broadcasts the transaction to other edge nodes, notifies them to stop competing for the right to record, and adds the task block to the blockchain after obtaining the consent of more than half of the other edge nodes [25]. As a result, blockchain is used as public accounts to record edge node status and real-time scheduling strategy, which ensures the security of data flow and reduces the impact of uncertainty on workflow scheduling [26–29].

## 1.3 Paper contributions

The main contributions of the paper are as follows:

- We present basic concepts and definitions to analyze the processing time of tasks and the energy consumption of edge nodes in workflow scheduling in the hybrid environment of cloud computing and edge computing.
- We propose a blockchain-powered resource provisioning (BPRP) for uncertainties in the workflow scheduling process, which uses directed acyclic graph to represent the workflow of each edge node, and the success rate of tasks which are affected by uncertainties is quantified.
- Non-dominated sorting genetic algorithm III (NSGA-III) is used to optimize multi-objective, which reduces the processing time and energy consumption of workflow scheduling.
- Numerous simulations confirm the validity of BPRP. The simulation results show that the method effectively reduces energy consumption and execution time and solve the uncertainty problem.

The rest of the paper is as follows. Section 2 presents the basic concepts and model formulas. Section 3 describes the resource scheduling method based on blockchain. In Section 4, simulation experiments are given, which show the performance of our method. Section 5 describes the relevant work. Finally, Section 6 summarizes the conclusions and future work.

## 2 System model and problem formulation

In this section, basic concepts and definitions for the workflow scheduling based on edge computing are introduced. In addition, the energy consumption and the execution time of workflow scheduling are also analyzed. Key terms and the descriptions are listed in Table 1.

### 2.1 Basic concepts

Edge computing sinks computing tasks into the vicinity of the wireless access network, i.e., the base station, and

**Table 1** Key Terms and Descriptions Terms Descriptions

Terms	Descriptions
$E$	The set of edge nodes, $E = \{e_1, e_2, \dots, e_N\}$
$S$	The set of mobile devices, $S = \{s_1, s_2, \dots, s_M\}$
$T$	The set of workflow tasks, $T_i = \{t_1, t_2, \dots, t_Q\}$
$G$	The graph model workflow operations
$D$	The set of dependencies between tasks
$P(t_i)$	The parent tasks of the $i$ th task
$C(t_i)$	The children tasks of the $i$ th task
$ES(t_i)$	The earliest start time of the $i$ th task
$EF(t_i)$	The earliest finish time of the $i$ th task
$LF(t_i)$	The latest finish time of the $i$ th task
$DL(t_i)$	The deadline of the $i$ th task
$RT(t_i)$	The real completion time of the $i$ th task
$EC$	The energy consumption of workflow scheduling
$SR$	The success rate of workflow rescheduling

provides the services and computing functions required by mobile devices through the wireless access network. On the basis of the smart city, we establish a mobile computing model based on mobile edge computing. Each base station is defined as an edge node, which provides computing resources for nearby mobile devices. Suppose there are  $N$  edge nodes deployed is the edge computing environmental, denoted as  $E = \{e_1, e_2, \dots, e_N\}$ .

Let  $S = \{s_1, s_2, \dots, s_M\}$  be the set of mobile devices, where  $M$  is the amount of mobile services. Each mobile device in  $S$  generates workflows when task arrives, which schedules its computing tasks to different edge nodes to get sufficient computing resources.  $T_i = \{t_1, t_2, \dots, t_Q\}$  represents set of workflow tasks, where  $Q$  is the amount of it. The edge nodes in  $E$  provide computing resources for tasks in  $T$ ; in other words, tasks are scheduled to different edge nodes for execution.

**Definition 1 Dependencies among workflows.** *There are dependencies among some workflow tasks [30]; in order to express these relationships clearly, we use a direct acyclic graph  $G = (T, D)$  to model workflow operations, where  $D$  is the set of dependencies between workflow tasks. Each dependency is in the form of directed edge  $d_{i,j} = (t_i, t_j)$  ( $1 < i, j \leq Q$ ), which means only when the task  $t_i$  is completed can the task  $t_j$  begin. The dependencies of tasks in the workflow can be calculated by conditional probability [21].*

### 2.2 Completion time analysis of workflow

All workflows constitute a directed acyclic graph. In order to reduce the processing time of workflow, we find critical paths to meet the deadlines of workflow tasks. In order to simplify the problem, we add two virtual tasks  $t_{start}$  and

$t_{last}$  at the beginning and end of the workflow to ensure the single entry and single output of tasks.

**Definition 2 Parent tasks.** *Due to the dependencies among the tasks, a task  $t_i$  ( $i = \{1, 2, \dots, Q\}$ ) can start only when all its parent tasks are finished. We use the set  $P(t_i)$  ( $i = \{1, 2, \dots, Q\}$ ) to represents parent tasks of the  $i$ th task.*

The earliest start time of each task in a workflow is obtained from the earliest start time of its parent task, and the start time of the task  $t_1$  is regarded as zero. The earliest start time of the  $i$ th ( $i = \{1, 2, \dots, Q\}$ ) task is calculated by:

$$ES(t_i) = \begin{cases} 0, & i = \text{start}, \\ \max(ES(t_j) + TD(t_j, t_i) + ED(t_j)), & 0 < i \leq Q, t_j \in P(t_i), \end{cases} \quad (1)$$

where  $TD(t_i, t_j)$  represents the transmission delay between the parent tasks and the task  $t_i$  ( $i = \{1, 2, \dots, Q\}$ ) itself, and  $ED(t_i, t_j)$  denotes the minimum execution delay of the parent tasks. The earliest start time of the task  $t_i$  ( $i = \{1, 2, \dots, Q\}$ ) means that all parent tasks before it is completed as early as possible.

Therefore, the earliest finish time of the task  $t_i$  ( $i = \{1, 2, \dots, Q\}$ ) is calculated by:

$$EF(t_i) = ES(t_i) + ED(t_i). \quad (2)$$

**Definition 3 Children tasks.** *The set  $C(t_i)$  represents the subtasks, i.e., the children tasks of the task  $t_i$  ( $i = \{1, 2, \dots, Q\}$ ). The children tasks of the task  $t_i$  ( $i = \{1, 2, \dots, Q\}$ ) can start only when it is finished.*

Let  $LF(t_i)$  be the latest finish time of the task  $t_i$  ( $i = \{1, 2, \dots, Q\}$ ), which is defined as:

$$LF(t_i) = \begin{cases} D, & i = \text{last}, \\ \min(LF(t_j) - TD(t_i, t_j) - ED(t_j)), & 0 < i \leq Q, t_j \in C(t_i), \end{cases} \quad (3)$$

where  $D$  is the deadline of the whole workflow which is determined by the mobile users themselves. The latest finish time of the task  $t_i$  ( $i = \{1, 2, \dots, Q\}$ ) means the latest completion time without delaying the completion of subsequent children tasks and workflows.

**Definition 4 Parent critical path.** *The parent critical path of a task  $t_i$  ( $i = \{1, 2, \dots, Q\}$ ) represents the critical parent task  $t_j$  ( $j = \{1, 2, \dots, Q\}$ ) of task  $t_i$  and the parent critical path of  $t_j$  if it exists [1].*

The task sequence on the parent critical path of the task  $t_i$  ( $i = \{1, 2, \dots, Q\}$ ) is  $S_i = (p_1, p_2, \dots, p_H)$  ( $i = \{1, 2, \dots, Q\}$ ), where  $H$  is the amount of the tasks. The deadline of  $S_i$  is denoted as:

$$DL(S_i) = LF(p_H) - ES(p_1). \quad (4)$$

Each task needs to be completed before its deadline. Assume that the task  $p_j$  in  $S_i$  corresponds to the task  $t_i$  in  $T$ , then the deadlines [31] of each task on the parent

critical path is calculated by:

$$DL(p_j) = \frac{EF(p_j) - ES(p_1)}{EF(P_H) - ES(P_1)} \times (LF(p_H) - ES(p_1)), \quad (5)$$

$$DL(t_i) = DL(p_j). \quad (6)$$

After determining the deadlines of each task on the parent critical path, the unallocated tasks on the parent critical path will be allocated with optimal edge nodes according to the scheduling strategies.

### 2.3 Energy consumption analysis of workflow scheduling

In our model, we also consider the energy consumption in workflow scheduling and strive to minimize it. Firstly, we use a variable  $\alpha_{i,k}$  to represent the allocation status between the task  $t_i$  in  $T$  and the edge node  $e_k$  in  $E$ . If the task  $t_i$  is allocated to the edge node  $e_k$ ,  $\alpha_{i,k} = 1$ ; otherwise,  $\alpha_{i,k} = 0$ . Therefore, the energy consumption of workflow scheduling is denoted by:

$$EC = \sum_{i=1}^Q \sum_{k=1}^N \alpha_{i,k} \cdot te_{i,k} + \sum_{i=1}^Q ee_i, \quad (7)$$

where  $te_{i,k}$  is the transmission energy consumption which is generated when the task  $t_i$  is transferred to the edge node  $e_k$ , and  $ee_i$  represents the execution energy consumption of the task  $t_i$ . The  $ee_i$  is determined by the size of the computing tasks.

### 2.4 Success rate analysis of workflow rescheduling

Finally, we will evaluate the success rate of workflow rescheduling. Assume that there are  $A$  tasks needed to be rescheduled because of the uncertainties. The variable  $\beta_v$  ( $v = \{1, 2, \dots, A\}$ ) denotes whether the task  $t_v$  is successfully completed or not.

$$\beta_v = \begin{cases} 1, & RC(t_v) \leq DL(t_v), \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

where  $RT(t_i)$  is the real completion time of the task  $t_i$  in  $T$ . Whether the actual completion time of a task is prior to its limited deadline is the criterion to determine whether the task is rescheduled successfully.

Consequently, the success rate of workflow rescheduling is calculated by:

$$SR = \frac{1}{A} \sum_{v=1}^A \beta_v. \quad (9)$$

### 2.5 Problem definition

From the foregoing, execution time and energy consumption of the workflow are analyzed and quantified. In this paper, we aim to achieve the goal of minimizing the execution time variance and reducing the energy consumption presented in Eq.(7) while optimizing the workflow

scheduling strategies. The multi-objective optimization problem is formulated by:

$$\min(RT(t_{last}), \min(EC), \quad (10)$$

$$s.t. \forall t_i, RT(t_i) \leq DL(t_i). \quad (11)$$

## 3 Blockchain-powered resource provisioning method in edge computing

In this paper, a blockchain-powered resource provisioning (BPRP) is proposed to resolve the uncertainty during workflow scheduling. Firstly, blockchain is adopted to record the strategy of workflow scheduling and help all edge nodes to learn the strategy. Then, NSGA-III is employed to select balanced strategies, which is a multi-objective optimization problem. Finally, the optimal scheduling strategy is obtained by the SAW (simple additive weighting) and MCDM (multiple-criteria decision-making).

### 3.1 Select balanced workflow scheduling strategy using NSGA-III

#### 3.1.1 Encoding

Each task in the workflow is numbered and given a value, which is expressed by integers. In GA, the value of each task is viewed as a gene and represents that the task is migrated to an edge node with a corresponding number. The strategies of the workflow, consists of the scheduling strategies of all tasks, is treated as a chromosome.

#### 3.1.2 Fitness function

Fitness functions are critical metrics to ensure the diversity of the solutions in genetic algorithms (GA). In this paper, the fitness functions of the total energy consumption and the completion time are given in Eq.(3) and Eq.(7), respectively. To evaluate the performance of our method, these functions need to be normalized and minimized.

#### 3.1.3 Initialization

Before performing NSGA-III, some critical parameters need to be given for initialization, including the number of iterations  $It_n$  and the size of the population  $C_p$ , i.e., the number of the chromosomes. The crossover probability  $p_c$  and the mutation probability  $p_m$  are initialized to ensure diversity of the iterative results.

#### 3.1.4 Crossover and mutation

The crossover is the core in GA, and the crossover distinguishes GA from other evolutionary algorithms. After the positions of the intersections are determined, two individuals exchange partial gene fragments to obtain an effective search in the string space, and excellent traits are inherited by the offspring. The specific crossover operation is

shown in Fig. 1. The genes  $g_3$  and  $g_4$  in the chromosome  $p_1$  and  $p_2$  are selected as intersections, and two new chromosomes are generated after gene exchange. The mutation prevents the population from falling into local optimum by modifying part of the gene fragment in the chromosome. The specific mutation operation is shown in Fig. 2. The gene fragments  $g_3$  and  $g_4$  in the chromosome  $p_1$  are modified to obtain a new chromosome, which has a higher fitness value.

### 3.1.5 Selection

In NSGA-III, a set of reference points is first defined for further association with individuals, and reference points solve poor diversity caused by the crowded distance used in NSGA-II, which easily leads to the algorithm falling into local optimum. Next, the initial population  $P_{o_t}$  of size  $C_p$  is randomly generated. The population  $M_t$  of size  $2C_p$  is composed of the initial population  $P_{o_t}$  and the offspring population  $Q_t$  which is processed by the crossover and the mutation.

To select the best  $C_p$  individuals from the population  $M_t$  for the next generation, the fitness value of each individual is calculated, and the non-dominated sorting based on Pareto dominance divides the population  $M_t$  into different non-dominated layers according to the fitness values.  $FL = \{F_1, F_2, \dots, F_K\}$  is employed to represent each layer, and individuals with lower fitness values are divided into lower layers. From the first level, the individuals are selected into the offspring population  $M_{t+1}$  until the size of  $M_{t+1}$  is equal to  $C_p$ . Suppose that  $F_H$  is the largest acceptable layer, the individuals in larger layers are discarded. However, if all individuals in  $F_H$  are received, the size of the population  $M_{t+1}$  may be larger than  $C_p$ . Unlike the crowded distance-based approach in NSGA-II, reference point-based approach is employed to select the appropriate individuals from  $F_H$  in NSGA-III.

The normalization is a key part of the reference point-based approach. First, the ideal points, i.e., the minimum values of the population targets, are calculated for further

scalarization and the scalar formulas are given below:

$$RT_n = RT - RT_{\min}, \tag{12}$$

$$EC_n = EC - EC_{\min}. \tag{13}$$

where  $RT_{\min}$  and  $EC_{\min}$  are the minimum of the completion time and energy consumption, respectively.

Then, each fitness function is viewed as an axis and the ideal point is the origin. There is a maximum value for each fitness function, which is connected to the respective origin, and the intercept is determined by the intersection of the line and the coordinate axis. The normalized equations are given in Eq.(14) and Eq.(15).

$$RT_t = \frac{RT_n}{a_{rt} - RT_{\min}}, \tag{14}$$

$$EC_t = \frac{EC_n}{a_{ec} - EC_{\min}}. \tag{15}$$

where  $a_{rt}$  and  $a_{ec}$  are the intercepts.

The reference points are equidistantly distributed on the axis and the individuals are connected to these reference points for selection. When selecting the remaining individuals, a principle needs to be followed, i.e., the individuals corresponding to the reference points with fewer connections should be retained to ensure diversity. Therefore, if one reference point has little links in the previous layers, the corresponding closest individual in  $F_H$  is selected into  $M_{t+1}$ , and the process continues until the size of  $M_{t+1}$  is equal to  $C_p$ .

### 3.2 Assess the strategy using sAW and mCDM

In our paper, SAW and MCDM are used for evaluating the scheduling strategy. Since the completion time and the total energy consumption satisfy a certain degree of independence, different weights are specified to determine better strategies. The utility function is given as follows:

$$val = \chi_{rt} \cdot RT_t + \chi_{ec} \cdot EC_t. \tag{16}$$

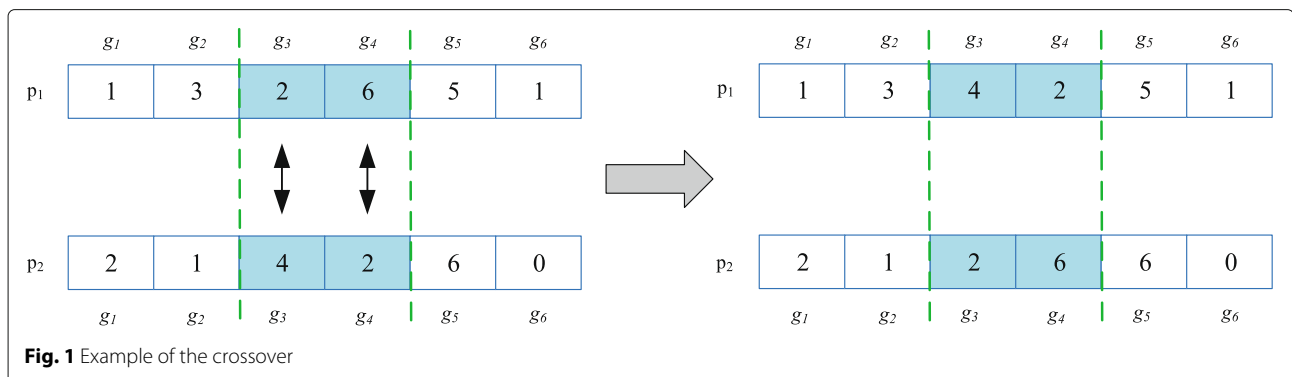
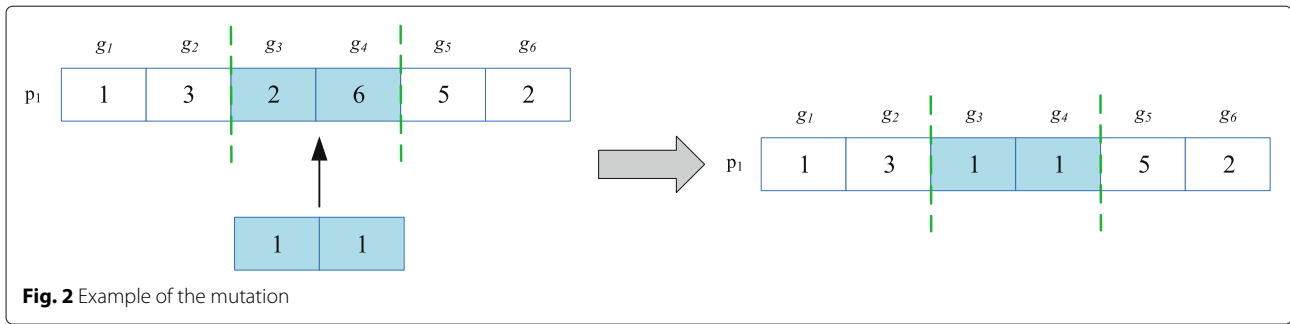


Fig. 1 Example of the crossover



**Fig. 2** Example of the mutation

where  $\chi_{rt}$  and  $\chi_{ec}$  are the weights of the completion time and the total energy consumption, respectively.

In Algorithm 1, the size of population  $C_p$ , the number of iterations  $It_n$ , the crossover probability  $p_c$ , and the mutation probability  $p_m$  are inputs. After initializing the population  $P_{0t}$  by the fitness functions, the population  $M_t$  is also determined. In the process of selecting the next generation, more balanced scheduling strategies are selected after each iteration. Finally, scheduling strategies  $R$  with higher utility values are selected through SAW and MCDM.

### 3.3 Deal with the uncertainty during the workflow scheduling

In the process of workflow scheduling, scheduling strategies are usually dynamically changed. Due to network congestion and performance degradation, new strategies

need to be developed to further optimize the completion time and total energy consumption. It is assumed that the uncertainty is generated when the task  $t_i$  is executed, the task  $t_i$  and other tasks on the path where  $t_i$  is located, i.e., the affected subsequent tasks, need to be organized into a new workflow for rescheduling. The corresponding new blocks are generated and added to the blockchain. In the process of admitting the transaction by other edge nodes, new scheduling strategies are simultaneously added to the task execution list by all edge nodes for updating.

### 3.4 Method view

In this paper, to address the uncertainty generated during the workflow scheduling, a blockchain-based resource provisioning method is proposed. During the scheduling process, the scheduling strategy for each task in the workflow is treated as a transaction and added to the blockchain. The transactions need to be recognized by all miners, i.e., edge nodes, so it is necessary to reduce the completion time and total energy consumption as much as possible. NSGA-III is able to perform well in this multi-objective optimization problem. First, NSGA-III is adopted to determine initial workflow scheduling strategies and new transactions are appended to the blockchain. Afterward, better scheduling strategies are selected by the utility function. Finally, the workflow will be rescheduled if the uncertainty is generated.

The proposed BPRP is summarized in Algorithm 2. First, the inputs are initial scheduling strategies  $R$  and the number of the affected tasks  $U_n$ . After the uncertainty is generated, the workflow is rescheduled and new scheduling strategies  $R_n$  are generated (lines 1–3). The new transactions corresponding to the new scheduling strategies also need to be recognized by all edge nodes. Finally, the deadline for each task in the workflow is determined, and the success rate is calculated by the efficiency of solving the affected tasks (lines 4–11).

## 4 Experimental evaluation

In this section, to evaluate the feasibility of our method BPRP, some experiments are performed to test the performance under simulated conditions. First, the simulated environment is built and the experiments are evaluated

---

#### Algorithm 1 Selection of The Strategy

---

**Require:**  $I_{it}, C_p, p_c, p_m$

**Ensure:**  $R$

- 1: Calculate  $RT$  and  $EC$
  - 2: Use  $RT$  and  $EC$  to initial  $P_{0t}$
  - 3: Obtain  $Q_t$  through  $p_c$  and  $p_m$
  - 4:  $M_t = P_t \cup Q_t$
  - 5:  $i = 1$
  - 6: **while**  $i < It_n$  **do**
  - 7:     Non-dominated sorting of  $M_t$
  - 8:      $j = 1$
  - 9:     **while**  $M_{t+1} \leq C_p$  **do**
  - 10:          $M_{t+1} = M_t \cup F_j$
  - 11:     **end while**
  - 12:     Normalize the individuals in  $F_j$
  - 13:     Determine reference points and connect reference points to  $F_j$
  - 14:     **while**  $M_{t+1} > C_p$  **do**
  - 15:         Delete the individuals from  $M_{t+1}$
  - 16:     **end while**
  - 17: **end while**
  - 18: Use SAW and MCDM to select the scheduling strategy  $R$  with higher utility value
  - 19: **return**  $R$
-

---

**Algorithm 2** Blockchain-powered Resource Provisioning

---

**Require:**  $R, U_n$

**Ensure:**  $R_n, SR$

- 1: The uncertainty is generated in blockchain-based framework
  - 2: Reschedule the workflow and generate a new scheduling strategy  $R_n$
  - 3: New transactions are appended to the blockchain and recognized by all edge nodes
  - 4:  $DL = \varepsilon \cdot RT_{old}$
  - 5:  $i = 1, sum = 0$
  - 6: **while**  $i \leq U_n$  **do**
  - 7:     **if**  $RT_{new} \leq DT$  **then**
  - 8:          $sum = sum + 1$
  - 9:     **end if**
  - 10: **end while**
  - 11:  $SR = sum/U_n$
  - 12: **return**  $R_n, SR$
- 

through some metrics. Afterward, comparative experiments are performed to compare with our method. Finally, the experimental results are assessed.

**4.1 Experimental context**

In our experiments, the workflows with different tasks are employed to simulate applications in mobile devices, and the number of tasks is set to 50, 100, 150, 200, 250, and 300, respectively. There are a total of 30 edge nodes, and different VMs are placed on these edge nodes to simulate different processing capabilities [32] [33]. Some other key parameters are described in Table 2.

Benchmark, First Fit (FF), and Next Fit (NF) are three algorithms that are chosen for comparison. The detailed introductions are given as follows.

(1) Benchmark: Each task in the workflow is migrated to a random edge node in order until all tasks have their own scheduling strategies. If the uncertainty arises while performing a task, new scheduling strategies are not regenerated. In our design, the affected tasks will continue to be executed until the tasks are completed.

(2) First Fit: The tasks in the workflow are preferentially migrated to the nearest edge nodes. If an edge node is unable to provide enough computing resources, a farther

edge node will accept the task. Moreover, new scheduling strategies are regenerated when uncertainty occurs.

(3) Next Fit: Unlike First Fit, this method does not seek service from the nearest edge node every time, but instead searches from the edge node that last provided the service.

These scheduling strategies are implemented on a laptop PC with Intel Core i7-7700HQ 2.80 GHz processors and 8G RAM.

**4.2 Performance assessment of bPRP**

The proposed BPRP is a multi-objective optimization problem, and two metrics, i.e., the completion time and the total energy consumption, are two optimization goals in this paper. There are many balanced methods generated in NSGA-III, and the optimal method is selected by SAW and MCDM. The utility value is a key indicator for evaluating the performance of a solution, and the utility values for workflows with a different number of tasks are given in Fig. 3. A solution with higher utility value is considered to be better. Therefore, solution 1 is considered as the optimal solution in Fig. 3a.

To evaluate the success rate of our method in addressing the uncertainty, a graph of success rate is given in Fig. 4. This figure compares the success rate of workflows with a different number of tasks when the deadline factor is set to 1.4. It is not difficult to find that a workflow with fewer tasks has a higher success rate when dealing with the uncertainty.

**4.3 Comparison of the completion time**

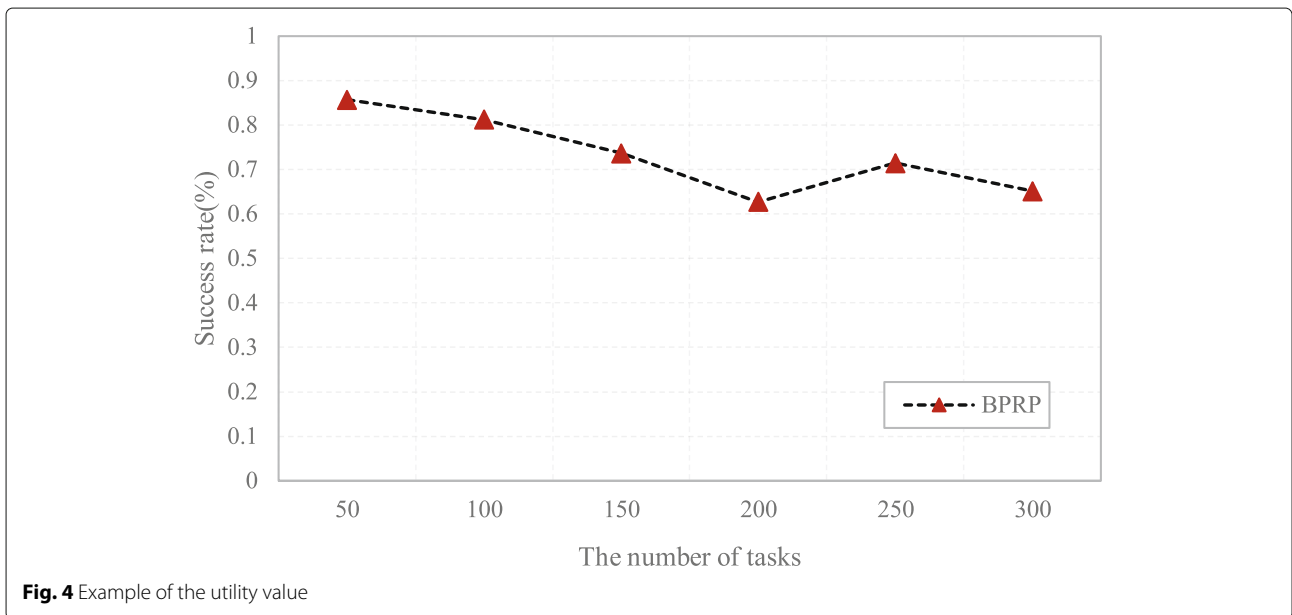
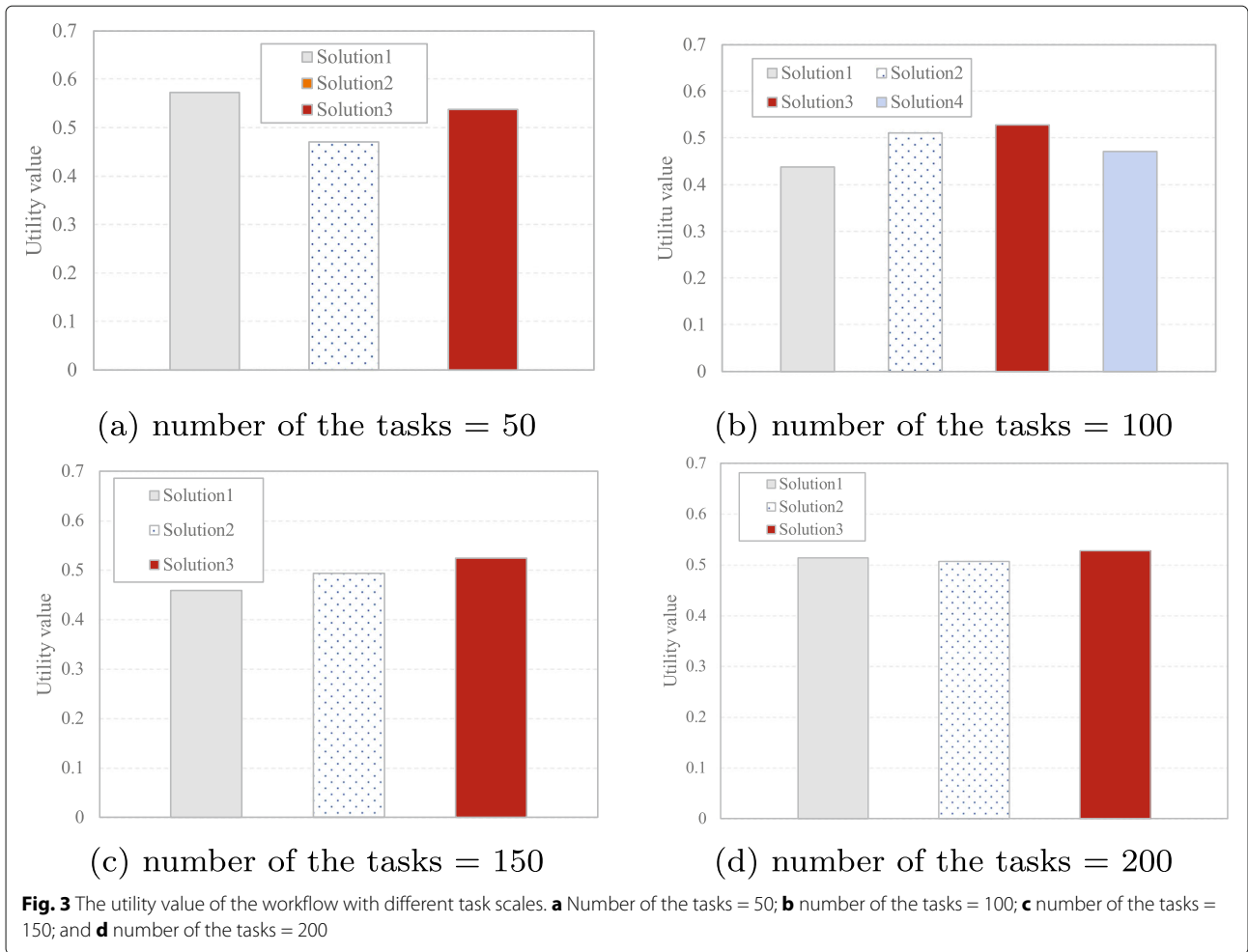
Figure 5 shows that BPRP consumes less time than two other methods. With the expansion of the task size in the workflow, the advantage of BPRP in terms of completion time is more obvious. Benchmark consumes the most time in three methods, which confirms that new scheduling strategies are usually superior in the event of the uncertainty. BPRP has less completion time than FF, which also confirms the superiority of NSGA-III in dealing with a multi-objective optimization problem. Moreover, in the process of the transactions being confirmed, the scheduling strategies are also detected by each edge node, which saves some time.

**4.4 Comparison of total energy consumption**

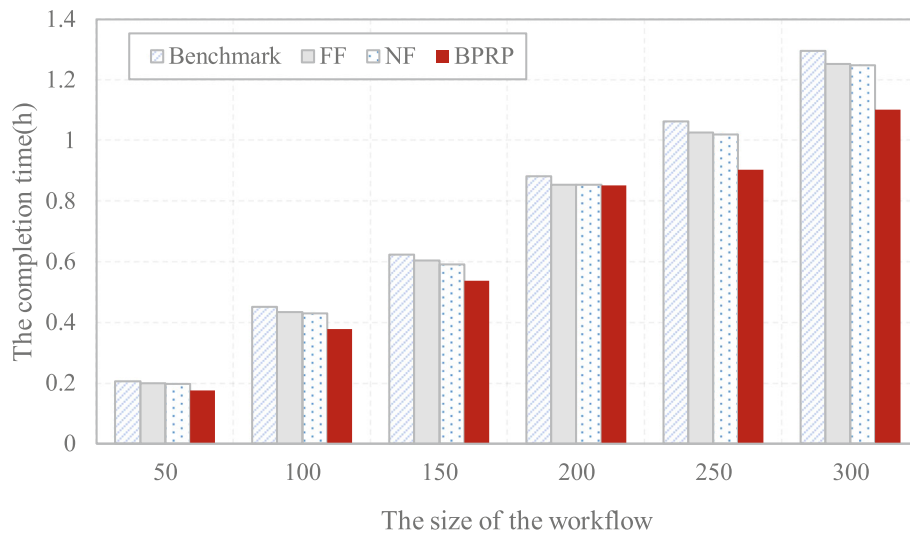
Figure 6 shows the comparisons of the total energy consumption for the three methods. It is clear that our method consumes less energy. In the decision process, the transaction corresponding to the scheduling strategy of each task is confirmed by each edge node. Therefore, the state of each edge node is able to be known by other edge nodes, and better strategies are generated. However, the tasks in the workflow are randomly migrated to other edge nodes in Benchmark, which results in many poor strategies being developed and consuming more energy.

**Table 2** Parameter settings

Parameter description	Value
The power of open VM $\mu$	0.25 KW
The idle power of VM $\kappa$	0.04 KW
The working power rate of VM $\gamma$	0.06 KW
The processing power of edge node $\omega$	2000 MHz
The bandwidth between edge nodes and mobile devices $\eta$	2000 mbps







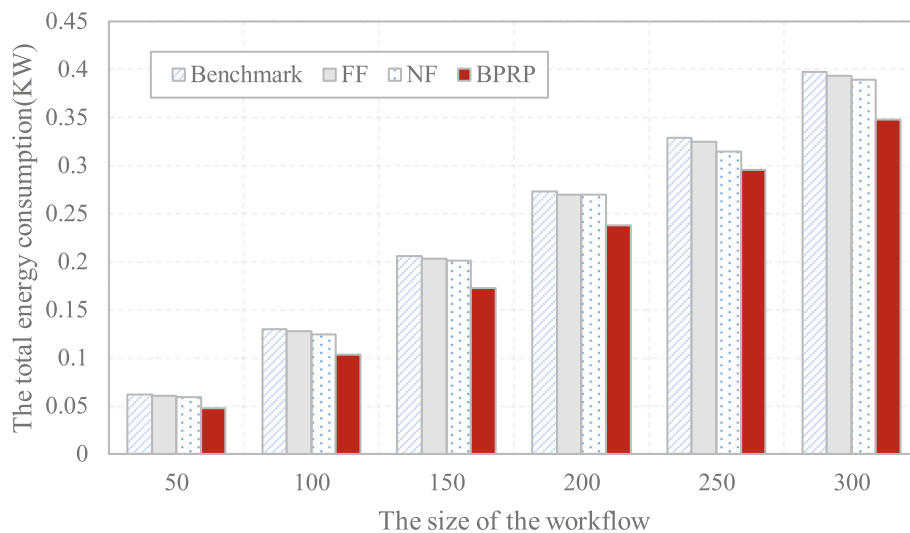
**Fig. 5** Comparison of the completion time

#### 4.5 Comparison of the success rate

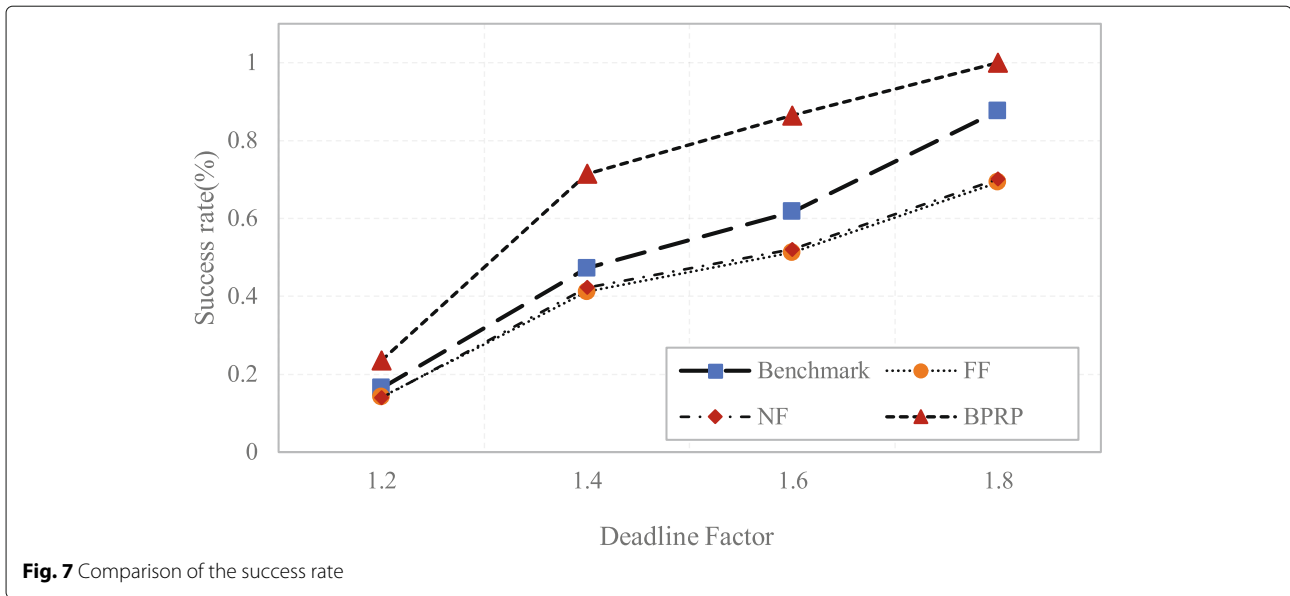
There is no doubt that tasks take more time to execute when uncertainty arises. In this experiment, the setting of the deadline is related to the ideal completion time of the workflow, i.e., the completion time when no uncertainty occurs. The relevant parameters, i.e., the deadline factors, are set to 1.2, 1.4, 1.6, and 1.8, respectively, which mean that the deadline is set a multiple of the ideal time. According to Fig. 7, BPRP has a higher success rate and increases as the deadline increases. When the deadline factor is equal to 1.8, the success rate reaches 1. Therefore, it predicts that the success rate of solving the uncertainty will gradually approach 1 if the time is sufficient.

#### 5 Related work

With the development of mobile communication technology, the basic framework of smart cities has been further improved. The concept of a smart city integrates big data and the Internet of Things (IoT) to connect everything in life to optimize the efficiency of urban processes and services and connect them to residents. At present, the blockchain has become a new model to support the smart city framework [34]. The blockchain technology has non-modifiability and no centrality, and the hash verification of the block makes the data transmission and recording real and reliable [35]. Blockchain has a great advantage in terms of the transfer value, which makes economic



**Fig. 6** Comparison of total energy consumption



value in data from mobile devices at relatively low cost. The author of [36] established a blockchain platform that uses the PBFT-DPOC consistency algorithm to achieve decentralized autonomy of mobile devices. [37] proposed an algorithm which helps to select the best resources from the cloud resource pool to save costs. [38] introduced a blockchain network simulator called SimBlock, which consists of thousands of nodes, simulating a peer-to-peer network in a blockchain, and verifies large-scale blockchain network behavior [39].

Workflow is one of the most typical applications in distributed computing. Due to the increase in data generated by mobile devices and the rapid increase in the number of workflows, workflow scheduling has become the focus of mobile communication technology development in recent years. [13] proposed a revised discrete particle swarm optimization (RDPSO) algorithm, which schedules workflows in the cloud, thereby reducing the energy consumption of data transmission in cloud workflows. Using a SHEFT algorithm in [40] allows computing resources to be dynamically scaled when workflow tasks are executed, saving computing resources and reducing overall execution time. A method for dynamic expansion of computing resources was also proposed in [41]. The scheduling algorithm in [42] determined the mapping between tasks and computing resources in the artifact path. By determining the effective mapping of tasks to resources on critical paths, computational tasks can be completed within a specified time. But the limitation of this article is that the author does not consider the cost of workflow execution. In the workflow scheduling, there will be uncertainty such as node failure and service reliability. In order to ensure that the tasks in the workflow are completed within the

deadline, these uncertainties need to be processed. The author of [19] proposed an algorithm called PRS that combines active and passive scheduling methods to schedule workflows, minimizing the impact of uncertainty on workflow scheduling. Considering that the available bandwidth is not able to be accurately estimated under the influence of uncertainty factors, [43] uses a particle swarm optimization based scheduler RobWE, which changes the function according to the user's needs. The author of [44] established a new model called R-MOHEFT, which estimated the processing time of tasks in the workflow when the resource elasticity configuration and the uncertainty interval were unknown. Assuming that the probability distribution function of uncertainty is known, two variables are proposed in [45] to measure the robustness of workflow execution, and a new workflow scheduling algorithm is proposed to solve the problem of computing resource allocation in the cloud resource pool.

### 6 Conclusion and future work

Cloud computing has the advantages of resource-saving, on-demand deployment, and high flexibility. In recent years, the development of cloud computing has been very hot. In order to reduce the round-trip delay of information transmission between the local device and the CDC, we use MEC as an extension of cloud computing to migrate the cloud platform from the inside of the mobile network to the edge of the mobile network to realize the flexible utilization of computing resources. Our paper proposes a blockchain-powered resource provisioning (BPRP) method for the uncertainty of workflow scheduling in the cloud. The blockchain is used to record the current state and task assignment of each node. At

the same time, the processing time and energy consumption of the workflow tasks are optimized by using the non-dominated sorting genetic algorithm III (NSGA-III) to ensure that the tasks are completed efficiently within the deadline. A large number of experiments have verified the effectiveness of the method.

In future applications, the proposed BPRP method will be extended to the actual scenario of workflow scheduling in the cloud, reducing the impact of uncertainty on workflow scheduling.

#### Abbreviations

MCDM: Multiple-criteria decision-making; NSGA-III: Non-dominated sorting genetic algorithm-III; SAW: Simple additive weighting

#### Acknowledgements

This work is supported by National Key Research and Development Project No. 2017YFB0504203, the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAET).

#### Authors' contributions

ZX, Q, HC, CW, and XL conceived and designed the study. ZX and XL performed the simulations. QG, HC, and CW wrote the paper. All authors reviewed and edited the manuscript. All authors read and approved the final manuscript.

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup>School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. <sup>2</sup>Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, China. <sup>3</sup>College of Information Science and Technology, Shihezi University, Shihezi, China.

Received: 25 August 2019 Accepted: 17 October 2019

Published online: 11 November 2019

#### References

1. S. Wan, Y. Zhao, T. Wang, Z. Gu, Q. H. Abbasi, K.-K. R. Choo, Multi-dimensional data indexing and range query processing via Voronoi diagram for internet of things. *Futur. Gener. Comput. Syst.* **91**, 382–391 (2019)
2. X. Xu, Y. Li, T. Huang, Y. Xue, K. Peng, L. Qi, W. Dou, An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks. *J. Netw. Comput. Appl.* **133**, 75–85 (2019)
3. H. Wang, C. Guo, S. Cheng, Loc—a new financial loan management system based on smart contracts. *Futur. Gener. Comput. Syst.* **100**, 648–655 (2019)
4. Y. Yin, W. Xu, Y. Xu, H. Li, L. Yu, Collaborative QoS prediction for mobile service with data filtering and SlopeOne model. *Mob. Inf. Syst.* **2017**, 1–14 (2017). <https://doi.org/10.1155/2017/7356213>
5. X. Xu, S. Fu, L. Qi, X. Zhang, Q. Liu, Q. He, S. Li, An IoT-oriented data placement method with privacy preservation in cloud environment. *J. Netw. Comput. Appl.* **124**, 148–157 (2018)
6. Y. Yin, F. Yu, Y. Xu, L. Yu, J. Mu, Network location-aware service recommendation with random walk in cyber-physical systems. *Sensors*. **17**(9), 2059 (2017)
7. H. Gao, W. Huang, X. Yang, Applying probabilistic model checking to path planning in an intelligent transportation system using mobility trajectories and their statistical data. *Intell. Autom. Soft Comput. (Autosoft)* (2019). <https://doi.org/10.31209/2019.100000110>
8. D. Wu, H. Wang, R. Seidu, Smart data driven quality prediction for urban water source management. *Futur. Gener. Comput. Syst.*
9. H. Gao, H. Miao, L. Liu, J. Kai, K. Zhao, Automated quantitative verification for service-based system design: A visualization transform tool perspective. *Int. J. Softw. Eng. Knowl. Eng.* **28**(10), 1369–1397 (2018)
10. L. Qi, Q. He, F. Chen, W. Dou, S. Wan, X. Zhang, X. Xu, Finding all you need: web APIs recommendation in web of things through keywords search. *IEEE Trans. Comput. Soc. Syst.* **6**(5), 1063–1072 (2019). <https://doi.org/10.1109/tcss.2019.2906925>
11. X. Xu, X. Liu, L. Qi, Y. Chen, Z. Ding, J. Shi, Energy-efficient virtual machine scheduling across cloudlets in wireless metropolitan area networks. *Mob. Netw. Appl.*, 1–15 (2019). <https://doi.org/10.1007/s11036-019-01242-6>
12. Z. Yang, Y. Huang, X. Li, W. Wang, F. Wu, X. Zhang, W. Yao, Z. Zheng, L. Xiang, W. Li, et al., Efficient secure data provenance scheme in multimedia outsourcing and sharing. *Comput. Mater. Continua.* **56**(1), 1–17 (2018)
13. Z. Wu, Z. Ni, L. Gu, X. Liu, in *2010 International Conference on Computational Intelligence and Security pages 184–188*. A revised discrete particle swarm optimization for cloud workflow scheduling (IEEE, 2010). <https://doi.org/10.1109/cis.2010.46>
14. J. Zhang, N. Xie, X. Zhang, K. Yue, W. Li, D. Kumar, Machine learning based resource allocation of cloud computing in auction. *Comput. Mater. Continua.* **56**(1), 123–135 (2018)
15. Y. Yuan, Y. S. Ong, A. Gupta, H. Xu, Objective reduction in many-objective optimization: evolutionary multiobjective approaches and comprehensive analysis. *IEEE Trans. Evol. Comput.* **PP**(99), 1–1 (2017)
16. Q. Mo, F. Dai, D. Liu, J. Qin, Z. Xie, T. Li, Development of private processes: a refinement approach. *IEEE Access.* **7**, 31517–31534 (2018)
17. H. Wang, S. Ma, H.-N. Dai, A rhombic dodecahedron topology for human-centric banking big data. *IEEE Trans. Comput. Soc. Syst.* **6**(5), 1095–1105 (2019). <https://doi.org/10.1109/tcss.2019.2918193>
18. X. Xu, Y. Chen, X. Zhang, Q. Liu, X. Liu, L. Qi, A blockchain-based computation offloading method for edge computing in 5g networks. *Softw. Pract. Experience.* <https://doi.org/10.1002/spe.2749>
19. H. Chen, X. Zhu, D. Qiu, L. Liu, in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*. Uncertainty-aware real-time workflow scheduling in the cloud (IEEE, 2016, pp. 577–584). <https://doi.org/10.1109/cloud.2016.0082>
20. Y. Yin, S. Aihua, G. Min, X. Yueshen, W. Shuoping, Qos prediction for web service recommendation with network location-aware neighbor selection. *Int. J. Softw. Eng. Knowl. Eng.* **26**(04), 611–632 (2016)
21. E. E. Mon, M. M. Thein, M. T. Aung, in *2016 9th Workshop on Many-Task Computing on Clouds, Grids, and Supercomputers (MTAGS)*. Clustering based on task dependency for data-intensive workflow scheduling optimization (IEEE, 2016), pp. 20–25. <https://doi.org/10.1109/mtags.2016.07>
22. X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, W. Dou, Become: Blockchain-enabled computation offloading for IoT in mobile edge computing. *IEEE Trans. Ind. Inform.*, 1–1 (2019). <https://doi.org/10.1109/tii.2019.2936869>
23. Z. Gao, D. Y. Wang, S. H. Wan, H. Zhang, Y. L. Wang, Cognitive-inspired class-statistic matching with triple-constrain for camera free 3D object retrieval. *Futur. Gener. Comput. Syst.* **94**, 641–653 (2019)
24. H. Wang, S. Ma, H.-N. Dai, M. Imrand, T. Wang, Blockchain-based data privacy management with nudge theory in open banking. *Futur. Gener. Comput. Syst.* <https://doi.org/10.1016/j.future.2019.09.010>
25. X. Xu, Y. Xue, L. Qi, Y. Yuan, X. Zhang, T. Umer, S. Wan, An edge computing-enabled computation offloading method with privacy preservation for Internet of connected vehicles. *Futur. Gener. Comput. Syst.* **96**, 89–100 (2019)
26. H. Gao, W. Huang, Y. Duan, X. Yang, Q. Zou, Research on cost-driven services composition in an uncertain environment. *J. Int. Technol.* **20**(3), 755–769 (2019)
27. Y. Yuan, W. Banzhaf, Arja: Automated repair of java programs via multi-objective genetic programming. *IEEE Trans. Softw. Eng.* **PP**, 99 (2017)
28. X. Xu, Q. Liu, Y. Luo, K. Peng, X. Zhang, S. Meng, L. Qi, A computation offloading method over big data for IoT-enabled cloud-edge computing. *Futur. Gener. Comput. Syst.* **95**, 522–533 (2019)
29. L. Qi, R. Wang, C. Hu, S. Li, Q. He, X. Xu, Time-aware distributed service recommendation with privacy-preservation. *Inf. Sci.* **480**, 354–364 (2019)
30. K.-H. Kim, in *17th International Conference on Advanced Information Networking and Applications, 2003. AINA 2003*. Workflow dependency analysis and its implications on distributed workflow systems (IEEE, 2003), pp. 677–682. <https://doi.org/10.1109/aina.2003.1192967>

31. S. Meng, S. Wang, T. Wu, D. Li, T. Huang, X. Wu, X. Xu, W. Dou, in *2016 IEEE International Conference on Web Services (ICWS)*. An uncertainty-aware evolutionary scheduling method for cloud service provisioning (IEEE, 2016), pp. 506–513. <https://doi.org/10.1109/icws.2016.72>
32. X. Xie, T. Yuan, X. Zhou, X. Cheng, Research on trust model in container-based cloud service. *Comput. Mater. Continua*. **56**(2), 273–283 (2018)
33. X. Xu, X. Zhang, M. Khan, W. Dou, S. Xue, S. Yu, A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems. *Futur. Gener. Comput. Syst.* (2017). <https://doi.org/10.1016/j.future.2017.08.057>
34. Jong-ho Noh, Hun-yeong Kwon, A study on smart city security policy based on blockchain in 5g age, in *2019 International Conference on Platform Technology and Service (PlatCon)* (IEEE, 2019), pp. 1–4. <https://doi.org/10.1109/platcon.2019.8669406>
35. S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, F.-Y. Wang, Blockchain-enabled smart contracts: architecture, applications, and future trends. *IEEE Trans. Syst. Man Cybern. Syst.* **49**(11), 2266–2277 (2019). <https://doi.org/10.1109/tsmc.2019.2895123>
36. S. Yu, K. Lv, Z. Shao, Y. Guo, J. Zou, B. Zhang, in *2018 1st IEEE international conference on hot information-centric networking (HotICN)*. A high performance blockchain platform for intelligent devices (IEEE, 2018), pp. 260–261. <https://doi.org/10.1109/hoticn.2018.8606017>
37. N. Chopra, S. Singh, in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. Heft based workflow scheduling algorithm for cost optimization within deadline in hybrid clouds (IEEE, 2013), pp. 1–6. <https://doi.org/10.1109/icccnt.2013.6726627>
38. R. Banno, K. Shudo, in *Proc. 2019 IEEE Int'l Conf. on Blockchain and Cryptocurrency (IEEE ICBC 2019)*. Simulating a blockchain network with simblock, (2019), pp. 3–4. <https://doi.org/10.1109/bloc.2019.8751431>
39. L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, X. Xu, A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems. *World Wide Web*, 1–23 (2019). <https://doi.org/10.1007/s11280-019-00684-y>
40. C. Lin, S. Lu, in *2011 IEEE 4th International Conference on Cloud Computing*. Scheduling scientific workflows elastically for cloud computing (IEEE, 2011), pp. 746–747. <https://doi.org/10.1109/cloud.2011.110>
41. Y. Ahn, Y. Kim, in *2014 International Conference on Cloud and Autonomic Computing*. VM auto-scaling for workflows in hybrid cloud computing (IEEE, 2014), pp. 237–240. <https://doi.org/10.1109/iccac.2014.34>
42. M. Rahman, S. Venugopal, R. Buyya, in *Third IEEE International Conference on e-Science and Grid Computing (e-Science 2007)*. A dynamic critical path algorithm for scheduling scientific workflow applications on global grids (IEEE, 2007), pp. 35–42. <https://doi.org/10.1109/e-science.2007.3>
43. T. A. L. Genez, L. F. Bittencourt, R. Sakellariou, E. R. M. Madeira, in *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*. A robust scheduler for workflow ensembles under uncertainties of available bandwidth (IEEE, 2017), pp. 504–511. <https://doi.org/10.1109/cloud.2017.70>
44. H. M. Fard, S. Ristov, R. Prodan, in *2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)*. Handling the uncertainty in resource performance for executing workflow applications in clouds (IEEE, 2016), pp. 89–98. <https://doi.org/10.1145/2996890.2996902>
45. D. Poola, S. K. Kumar Garg, R. Buyya, Y. Yang, K. Ramamohanarao, in *2014 IEEE 28th international conference on advanced information networking and applications*. Robust scheduling of scientific workflows with deadline and budget constraints in clouds (IEEE, 2014), pp. 858–865. <https://doi.org/10.1109/aina.2014.105>

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

---