

## Research Article

# A Dynamic Pricing Reverse Auction-Based Resource Allocation Mechanism in Cloud Workflow Systems

Xuejun Li,<sup>1,2</sup> Ruimiao Ding,<sup>1</sup> Xiao Liu,<sup>3</sup> Xiangjun Liu,<sup>1</sup> Erzhou Zhu,<sup>1</sup> and Yunxiang Zhong<sup>1</sup>

<sup>1</sup>*School of Computer Science and Technology, Anhui University, Hefei, China*

<sup>2</sup>*School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia*

<sup>3</sup>*School of Information Technology, Deakin University, Melbourne, Australia*

Correspondence should be addressed to Xiao Liu; [xiao.liu@deakin.edu.au](mailto:xiao.liu@deakin.edu.au)

Received 22 July 2016; Accepted 3 October 2016

Academic Editor: Wenbing Zhao

Copyright © 2016 Xuejun Li 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.

Market-oriented reverse auction is an efficient and cost-effective method for resource allocation in cloud workflow systems since it can dynamically allocate resources depending on the supply-demand relationship of the cloud market. However, during the auction the price of cloud resource is usually fixed, and the current resource allocation mechanisms cannot adapt to the changeable market properly which results in the low efficiency of resource utilization. To address such a problem, a dynamic pricing reverse auction-based resource allocation mechanism is proposed. During the auction, resource providers can change prices according to the trading situation so that our novel mechanism can increase the chances of making a deal and improve efficiency of resource utilization. In addition, resource providers can improve their competitiveness in the market by lowering prices, and thus users can obtain cheaper resources in shorter time which would decrease monetary cost and completion time for workflow execution. Experiments with different situations and problem sizes are conducted for dynamic pricing-based allocation mechanism (DPAM) on resource utilization and the measurement of Time\*Cost (TC). The results show that our DPAM can outperform its representative in resource utilization, monetary cost, and completion time and also obtain the optimal price reduction rates.

## 1. Introduction

Workflow model is often used to manage complex business applications. A workflow is defined as a collection of tasks which are handled in a specific order [1, 2]. A workflow management system needs to allocate and execute tasks efficiently to meet users' needs. Cloud computing uses a pay-as-you-go model which provides virtually unlimited computational resources at lower costs with better reliability and delivers the resources by means of virtualization technologies [3]. Cloud workflow systems are workflow systems deployed on cloud computing environment to gain unlimited resources including computation, storage, and network [4].

Resource allocation for cloud workflow systems has received much attention. Allocating cloud resources to workflow is an NP-hard problem and needs to consider the overall performance of system especially monetary cost and completion time [5]. Resource allocation mechanism includes conventional methods and market-oriented methods [6, 7].

Conventional methods require global knowledge and complete information. Users pay for the resources based on reserved price. In contrast, market-oriented methods can offer incentives to participants and the methods decide the price based on the values that users can get from the resources [8].

Different from conventional counterparts, market-oriented methods assume that providers and users are rational and intelligent. And resource allocation depends on many factors including supply-demand relationship and resource price. Auction is a powerful tool to allocate resources in the market. Generally speaking, auction is a protocol that allows participants to indicate their interests in different resources and use these indications of interest to determine both resource allocation and price [9]. Reverse auction method is a typical auction. In conventional auction, there are one seller and multiple buyers. As for reverse auction, there are multiple sellers and only one buyer [10]. The user sends the specification of resource requirement to the cloud broker and requests

resources. The cloud broker transfers the specification to all cloud providers. The cloud providers sell resources with proper price and capabilities. And then the users select the optimal resources according to some criteria, for example, Quality of Service [11, 12]. In general, reverse auction is used to prevent the occurrence of trading fraud and achieve dynamic pricing and automatic procurement [11]. In the most relevant literature [12], the authors propose Biobjective Scheduling Strategy (BOSS) based on reverse auction to allocate resources for tasks of workflows and each task starts an auction and gets a resource to minimize the monetary cost and completion time. However, the resource price is fixed during the auction, so that some providers with weaker competitiveness may lose the auction all the time. This leads to low efficiency of resource utilization, which decreases the allocation equilibrium of tasks on resources in the cloud market.

In this paper, we firstly present a dynamic pricing strategy to change resource prices according to the trading situation and then present a DPAM mechanism to improve the efficiency of resource utilization. Changing resource price is a common and efficient way to increase competitiveness [13]. In the dynamic pricing strategy, the resource price changes according to the trading situation. Those providers who have strong competitiveness (i.e., higher chance of winning in historic auctions) in the market often keep resource prices unchanged during an auction. But for those who have weak competitiveness, decreasing the resource price with a certain rate is an effective way to increase the winning chances. Therefore, changing prices can increase the chance of winning an auction and gaining more revenue for providers with weaker competitiveness. In the meantime, the users can gain cheaper resources. In DPAM, many providers with weak competitiveness use dynamic pricing strategy to increase winning chances and revenue, so that resource utilization of the cloud market increases. Meanwhile, cloud workflows will be executed timely with less monetary cost. In our experiment, the measurement of TC is employed to evaluate the performance of our proposed strategy.

In our previous preliminary work, a dynamic pricing strategy in reserve auction was presented to change resource prices according to the trading situation, and then a novel DPAM was proposed to improve the efficiency of resource utilization [14]. Based on this work, we have further investigated the resource allocation based on reverse auction and made the following substantial extensions in this paper:

- (i) In problem analysis, a real world stock trading workflow is given and resource allocation processes of BOSS and DPAM are described based on this example. The process illustrates the difference of pricing mechanism between BOSS and DPAM.
- (ii) In evaluation, firstly the performance of BOSS and DPAM on resource utilization and the measurement of TC with different problem sizes are tested. Then simulation on DPAM with different price reduction rates evaluates its performance on resource utilization and TC.

Dynamic pricing based allocation mechanism is designed to improve resource utilization and decrease monetary cost and completion time. In summary, this mechanism has three major advantages:

- (i) In reverse auction, dynamic pricing strategy is more effective to increase the revenue of providers with weak competitiveness and decrease users' cost than fixed pricing. Providers with weak competitiveness decrease the price to increase the chance of winning auction and gaining more revenue. Simultaneously, users who choose the resource with lower price will have less monetary cost.
- (ii) Dynamic pricing based allocation mechanism can improve resource utilization for providers. Providers change resource prices to sell more resources according to dynamic pricing strategy especially for those with weak competitiveness. This brings higher resources utilization because more resources are chosen.
- (iii) Dynamic pricing based allocation mechanism can decrease monetary cost and completion time for users. More price-competitive resources will appear in the market because of increasing competition among providers. So users can easily obtain cheaper and more resources and hence decrease monetary cost and completion time.

The rest of the paper is organized as follows. Section 2 describes some related research. In Section 3, we show an example to analyze the problem. Section 4 proposes a dynamic pricing strategy, and Section 5 presents dynamic pricing-based allocation mechanism. In Section 6, traditional mechanism and ours are evaluated. Finally, Section 7 concludes the paper and discusses our future work.

## 2. Related Work

Resource allocation has become an important task to provide efficient and economical resources in cloud computing environment [15]. Wood et al. [16] propose an approach for dynamic allocation of resources by defining a unique metric based on the consumption of the three resources: CPU, network, and memory. Görlach and Leymann propose a method for dynamic provisioning of services in clouds in order to optimize the distribution of services [17]. However, their proposed approaches are not efficient and economic because the allocation of resource does not consider market situation.

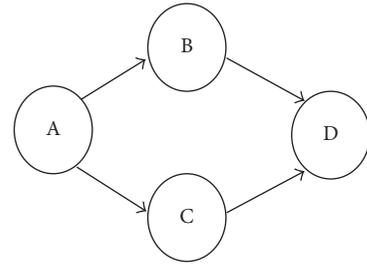
Market-oriented resource allocation has received much attention as it is a significant problem of large-scale distributed systems. In [18], authors present a model for resource allocation in grid using market-oriented concepts including commodity market, posted price modeling, and contract net models bargaining modeling. Mao and Humphrey explore the cloud autoscaling framework for resource allocation. The goal is to ensure that all jobs finish before their respective deadlines while running on these resources which consume the least amount of money [19]. In [20], in order to effectively manage resource allocation and workflow execution, Wang et al. design a mechanism which responds to the user's

continuous workflow requests and schedules their execution. In [21, 22], a lot of heuristic methods are presented to solve the problem in services systems. These heuristic methods consider the optimization algorithm from the aspects of cost, deadline, and reliability which can improve the performance of the algorithm. Ludwig presents a heuristic program for resource allocation on utility computing infrastructure. This heuristic program optimizes the number of resources allocated to tasks of workflow and speeds up the execution within a limitation of budget [23]. In [24], the authors propose a resource allocation approach to better match the resource allocated to the job with the cloud's residual resource.

Auction [25, 26] is a popular method to solve resource allocation problem. In [27, 28], the authors present auction-based mechanisms to determine optimal resource price, taking into account the user's budget and time constraints. Prasad G et al. present a combinatorial auction mechanism to allocate multiple resources in one auction [6]. However, they consider the pricing model of only one seller. Reverse auction is a popular auction in which the roles of buyer and seller are reversed [29]. In an ordinary auction buyers compete to obtain goods or service by offering increasingly higher prices [30]. In the reverse auction [31], the sellers typically decrease prices to compete against each other and obtain business from the buyer. The authors employ reverse auction to select the optimal resource based on available information to maximize their own profits [32]. A cloud resource allocation approach based reverse auction [33] is presented to select suitable cloud resource providers for users.

However, these methods do not focus on the pricing mechanism of the resource allocation. Resource pricing is an important aspect in resource allocation. In [34], authors mention Commodity Market which means that the sellers set the price for merchandise and the buyers pay money to get it. The price is predetermined by the seller and does not change over time based on supply-demand relationship. But fixed pricing is not suitable for the changeable market of cloud resources.

Dynamic pricing has gained wide attention from both industry and academia in the cloud computing. Amazon EC2 [35] has introduced a "spot pricing" scheme where the spot price is set according to resource supply and demand. Because fixed pricing does not reflect the dynamic changes of supply and demand, a dynamic scheme for allocation of multiple-type resources [36] is proposed to increase the percentages of successful buying and selling. In [12], authors introduce a pricing model and a truthful mechanism for scheduling single tasks considering two objectives: completion time and monetary cost based on reverse auction. However, they do not consider the competitiveness among providers which leads to the fact that the losers may always lose auction because they do not try to improve their competitiveness. To solve this problem, we propose dynamic pricing based scheduling mechanism for allocating resources efficiently, in which providers who lose an auction will decrease resource price in order to win so as to gain more revenue.



A: stock issuance      C: continuous auction  
B: fixed auction      D: formation of price chart

FIGURE 1: Stock trading workflow.

TABLE 1: Characteristic of tasks.

Task	Workload
A	6
B	4
C	5
D	7

TABLE 2: Characteristics of resources.

RN	CA	RP	SP
1	1	0.09	0.14
2	1.4	0.14	0.20
3	1.2	0.12	0.17

RN: resource number, CA: computation ability, RP: reserve price, and SP: starting price.

### 3. Problem Analysis

In this section, a stock trading workflow is given to explain the difference of pricing mechanism between the BOSS mechanism and ours. Stock trading is a typical process in the market. At first, stock exchange starts with stock issuance and then price formation process follows. During this process, fixed auction and continuous auction happen simultaneously. At last, price chart is generated.

As shown in Figure 1, stock trading workflow contains four tasks. The execution sequence of tasks partially depends on relation. The succeeding tasks start only when their predecessor tasks finish. Table 1 indicates the workload of each task. Table 2 shows some characteristics of three resources including resource number, computation ability, reserve price, and starting price. Here, computation ability is the computation speed of CPUs, the reserve price is the lowest price of resource during the auction, and the starting price is the first resource price when auction starts.

**3.1. Resource Allocation Process of DPAM.** For the BOSS mechanism [12], tasks start an auction according to the specific order to select the resource with the minimum product of completion time and total monetary cost. And providers give their bids  $bid_j = (CA_j, SP_j)$  which indicate computation ability and starting price of resource  $j$ . The workflow

TABLE 3: Resource allocation process of BOSS.

RN	ST	ET	CT	MC	TC	WR
(a) First auction						
1	0	6.00	6.00	0.84	5.04	
2	<b>0</b>	<b>4.29</b>	<b>4.29</b>	<b>0.86</b>	<b>3.67</b>	2
3	0	5.00	5.00	0.85	4.25	
(b) Second auction						
1	4.29	4.00	8.29	0.56	4.64	
2	<b>4.29</b>	<b>2.86</b>	<b>7.15</b>	<b>0.57</b>	<b>4.08</b>	2
3	4.29	3.33	7.62	0.57	4.32	
(c) Third auction						
1	4.29	5.00	9.29	0.70	6.50	
2	7.15	3.57	10.72	0.71	7.66	3
3	<b>4.29</b>	<b>4.17</b>	<b>8.46</b>	<b>0.71</b>	<b>5.99</b>	
(d) Last auction						
1	8.46	7.00	15.46	0.98	15.15	
2	<b>8.46</b>	<b>5.00</b>	<b>13.46</b>	<b>1.00</b>	<b>13.46</b>	2
3	8.46	5.83	14.29	0.99	14.17	

RN: resource number, ST: start time, ET: execution time, CT: completion time, MC: monetary cost, TC: Time \* cost, and WR: winner.

completion time is the time required for executing the whole workflow under the partially ordered relation of tasks. The total monetary cost of workflow is the sum of all tasks' monetary cost. The task's monetary cost only covers the execution cost on the allocated resource. It is assumed that there is no additional cost while moving execution from one cloud provider to another. Once a task is assigned to one resource, it will be executed on this resource until its completion and cannot be reallocated to a cheaper or better resource during its execution. The resource allocation process of BOSS is shown in Table 3.

In Table 3, the start time is the larger one between the resource's free time and predecessor tasks' finish time. The execution time is calculated as the workload divided by computation ability. The completion time is the sum of the start time and the execution time. The measurement of TC is the product of completion time and monetary cost. Tasks select the resource with the minimum TC. In Table 3(a), task A selects resource 2 as the winner, because its TC is the minimum. Similarly, task B selects resource 2 as the winner in the second auction and task C selects resource 3 as the winner in the third auction. In the last auction, task D selects resource 2 as the winner. From the tables, the completion time is 13.46 which is the completion time of last task D on resource 2. The total monetary cost of all tasks is  $0.86 + 0.57 + 0.71 + 1.00 = 3.14$ . The product of completion time and total monetary cost is 42.2644.

**3.2. Resource Allocation Process of DPAM.** In our mechanism, the resource price can be dynamically changed to improve providers' competitiveness. If one provider wins an auction, resource price will be kept unchanged. Otherwise, resource price will be decreased at a certain rate in the next auction. However, during any auction, the resource price cannot be

TABLE 4: Resource allocation process of DPAM.

RN	CP	ST	ET	CT	MC	TC	WR
(a) First auction							
1	0.14	0.00	6.00	6.00	0.84	5.04	
2	<b>0.20</b>	<b>0.00</b>	<b>4.29</b>	<b>4.29</b>	<b>0.86</b>	<b>3.67</b>	2
3	0.17	0.00	5.00	5.00	0.85	4.25	
(b) Second auction							
1	0.11	4.29	4.00	8.29	0.45	3.71	
2	0.20	4.29	2.86	7.15	0.57	4.08	3
3	<b>0.14</b>	<b>4.29</b>	<b>3.33</b>	<b>7.62</b>	<b>0.45</b>	<b>3.46</b>	
(c) Third auction							
1	<b>0.09</b>	<b>4.29</b>	<b>5.00</b>	<b>9.29</b>	<b>0.45</b>	<b>4.16</b>	
2	0.16	4.29	3.57	7.86	0.57	4.49	1
3	0.14	7.62	4.17	11.79	0.57	6.68	
(d) Last auction							
1	0.09	9.29	7.00	16.29	0.63	10.22	
2	<b>0.14</b>	<b>9.29</b>	<b>5.00</b>	<b>14.29</b>	<b>0.70</b>	<b>10.00</b>	2
3	0.12	9.29	5.83	15.12	0.70	10.59	

RN: resource number, CP: current price, ST: start time, ET: execution time, CT: completion time, MC: monetary cost, TC: Time \* cost, and WR: winner.

lower than its reserve price. Resource allocation process of DPAM is shown in Table 4. Here the price reduction rate is set as 20%.

At first, task A starts an auction and three resources give their bids (CA, CP). As shown in Table 4(a), the current price is the resource price of current auction. Task A selects resource 2 as the winner, because its TC is the minimum. So resource 1 and resource 3 lose the auction and they decrease the current prices by 20% in Table 4(b). In the second auction, task B selects resource 3 as the winner. Then resource 1 and resource 2 decrease their current price by 20% as shown in Table 4(c). Similarly, task C selects resource 1 as the winner and task D selects resource 2. From Table 4, the completion time of the workflow is 14.29 which is the completion time of last task. The total monetary cost of all tasks is  $0.86 + 0.45 + 0.45 + 0.70 = 2.46$ . The product of completion time and total monetary cost is 35.1534.

**3.3. Comparison of BOSS and DPAM.** In this subsection, resource prices and the winner of each auction are compared between BOSS and DPAM as shown in Table 5.

In Table 5, the winners of BOSS are 2, 2, 3, and 2. But the winners of DPAM are 2, 3, 1, and 2. In BOSS, resource price is always fixed and resource 1 with low competitiveness is never used. However, in DPAM resource 1 becomes new winner by dynamic pricing strategy which brings higher resource utilization. Resource utilization shows the allocation of tasks on resources (see Formula (2)). Therefore, resource utilization of DPAM is  $1/[(2 - 4/3)^2 + (1 - 4/3)^2 + (1 - 4/3)^2]/3 = 9/2$ , which is bigger than resource utilization of BOSS ( $1/[(3 - 4/3)^2 + (1 - 4/3)^2 + (0 - 4/3)^2]/3 = 9/14$ ). The reason is that these providers which never become winner in BOSS may win the auction in DPAM. This means more providers win the auction and sell their resources.

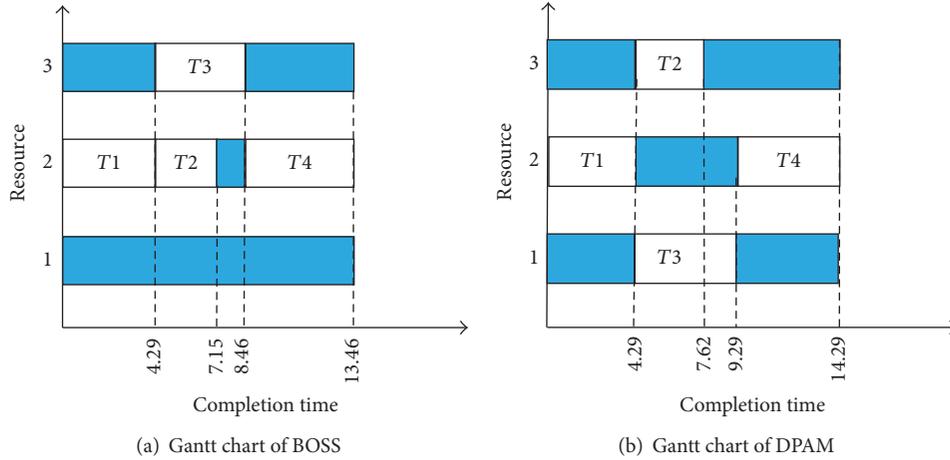


FIGURE 2: Gantt charts of resource allocation and task execution.

TABLE 5: Comparison of BOSS and DPAM.

RN	PB	PD	WB	WD
(a) First auction				
1	0.14	0.14		
2	0.20	0.20	2	2
3	0.17	0.17		
(b) Second auction				
1	0.14	0.11		
2	0.20	0.20	2	3
3	0.17	0.14		
(c) Third auction				
1	0.14	0.09		
2	0.20	0.16	3	1
3	0.17	0.14		
(d) Last auction				
1	0.14	0.09		
2	0.20	0.14	2	2
3	0.17	0.12		

RN: resource number, PB: price of BOSS, PD: price of DPAM, WB: winner of BOSS, and WD: winner of DPAM.

The Gantt charts of BOSS and DPAM are depicted in Figure 2. The charts show tasks execution order and the resource executes on which task. In Figure 2(a), only resources 2 and 3 are used. While in Figure 2(b) all resources are used. It is easy to draw that DPAM has higher resource utilization than BOSS. In DPAM resource price is dynamic, so resource with weak competitiveness decreases price to improve competitiveness until it wins one auction. However, in BOSS the resource with low competitiveness may never win any auction.

#### 4. Dynamic Pricing Strategy

As the number of cloud resource providers increases in reverse auction, they compete against each other to maximize their revenue. So an effective pricing strategy is necessary

for providers to increase their competitiveness. Firstly, two propositions are described to prove that the dynamic pricing strategy can improve the revenue of providers and also decrease the monetary cost of users. Then dynamic pricing strategy is proposed.

**Proposition 1.** *Dynamic pricing strategy can increase the revenue of provider with weak competitiveness.*

*Proof.* Assume that provider A and provider B have the resources with same computation ability. Resource price of A is  $p_A$  and resource price of B is  $p_B$ , where  $p_A < p_B$ . So provider B will lose auction because his competitiveness is weaker than A. If the resource price is fixed, competitiveness of B is always weaker than A and then provider B would never win an auction. Otherwise, if resource price is dynamic, provider B can decrease the price from  $p_B$  to  $p'_B$  which is lower than  $p_A$ . Then B can win auction and hence increase revenue because its competitiveness is higher than that of A. Hence, the dynamic pricing strategy can increase the revenue of provider with weak competitiveness.  $\square$

**Proposition 2.** *Dynamic pricing strategy can decrease user's monetary cost.*

*Proof.* Assume that provider A and provider B have the resources with same computation ability. Resource price of A is  $p_A$  and resource price of B is  $p_B$ , where  $p_A < p_B$ . User will select A's resource because its resource price is lower. If resource price is fixed, user will always select A's resource and monetary cost is  $p_A$ . Otherwise, if resource price is dynamic, provider B must decrease the price from  $p_B$  to  $p'_B$  to win the auction. Here  $p'_B$  is lower than  $p_A$ . So user will select B's resource and monetary cost is  $p'_B$ . Hence, dynamic pricing strategy can decrease user's monetary cost.

Each provider sets reserve price, starting price, and price reduction rate for a resource. When one task starts auction, providers join the auction and give their bids with computation ability and price. After one auction finishes, providers change or do not change the resource price according to

transaction situation. If providers want to increase competitiveness and win auctions, they will change their price according to the dynamic pricing strategy in Formula (1):

$$p_A^{\text{cur}'} = \begin{cases} p_A^{\text{cur}}, & \text{if A is winner,} \\ p_A^{\text{cur}} \cdot (1 - \gamma), & \text{if A is loser and } p_A^{\text{cur}} \cdot (1 - \gamma) > p_A^{\text{res}}, \\ p_A^{\text{res}}, & \text{if A is loser and } p_A^{\text{cur}} \cdot (1 - \gamma) < p_A^{\text{res}}, \end{cases} \quad (1)$$

where  $p_A^{\text{cur}'}$  refers to the current resource price of provider A.  $p_A^{\text{res}}$  refers to the reserve price of resource.  $\gamma$  denotes the price reduction rate. In the strategy, if provider A is the winner, its resource price will still be  $p_A^{\text{cur}}$  in the next auction. Otherwise, if provider A is the loser and  $p_A^{\text{cur}} \cdot (1 - \gamma) > p_A^{\text{res}}$ , its resource price will be  $p_A^{\text{cur}} \cdot (1 - \gamma)$  in the next auction. The resource price will be  $p_A^{\text{res}}$ , if  $p_A^{\text{cur}} \cdot (1 - \gamma) < p_A^{\text{res}}$ .

In conclusion, dynamic pricing strategy is efficient during the auction. Providers decrease the prices to increase the chance of winning auctions in order to gain more revenue. Simultaneously users choose the resources with lower prices and spend less monetary cost.  $\square$

## 5. Dynamic Pricing Based Allocation Mechanism (DPAM)

In this section, firstly resource utilization (Formula (2)) and evaluation value TC (Formula (3)) are defined and then novel dynamic pricing based allocation mechanism is proposed. In the auction-based cloud market, the purpose of providers is to sell resources at the most proper price so as to gain the highest revenue. And the purpose of users is to execute workflows with shortest completion time and lowest monetary cost. In this mechanism, users select the best resource according to the product of completion time and monetary cost. And the provider with the minimum product will be the winner. After each auction, providers change their resource price according to current trading situation. If their competitiveness is weak and loses the auction, they usually decrease the price in certain rate to increase competitiveness. Otherwise, if they win, it is effectively to keep the price unchanged or increased.

Resource utilization shows the allocation equilibrium of tasks on resources. It is described by variance of winning auction times for each provider. Resource utilization is inversely proportional to variance. Especially when the variance is zero, resource utilization is optimal.

$$\text{ResourceUtilization} = \frac{1 / \sum_1^n (\text{num}_j - \overline{\text{num}})^2}{n}, \quad (2)$$

where  $n$  is the amount of resources.  $\text{num}_j$  refers to winning times of resource  $j$  and  $\overline{\text{num}}$  is the average value of winning auction times of all providers.

During auction, a task selects the resource with the minimum TC as the winner.  $\text{TC}_{ij}$  is the product of completion time and monetary cost of task  $i$  on resource  $j$ . In [12], authors use the measurement TC to measure the BOSS with other

mechanisms. There are two reasons for using TC as a measurement: (1) it presents the whole evaluation of completion time and monetary cost for workflow execution; (2) the truthfulness of the BOSS mechanism depends on TC. So we use the measurement TC in order to make a more accurate comparison with BOSS.

$$\text{TC}_{ij} = \left( t_{ij} + \frac{\text{workload}_i}{\text{ability}_j} \right) * \left( \text{price}_j * \frac{\text{workload}_i}{\text{ability}_j} \right). \quad (3)$$

Each task starts execution only when its predecessor tasks have finished according to the partially ordered relation of tasks.  $t_{ij}$  refers to the start time of task  $i$  executing on resource  $j$ . It equals the latest time when its predecessor tasks have finished and simultaneously resource  $j$  is idle.  $\text{workload}_i$  is the workload of task  $i$ ;  $\text{ability}_j$  and  $\text{price}_j$  are computation ability and price per time unit of resource  $j$ , respectively. So  $\text{workload}_i / \text{ability}_j$  is the time required for task  $i$  on resource  $j$ . And  $(t_{ij} + \text{workload}_i / \text{ability}_j)$  refers to the finishing time of task  $i$  on resource  $j$ .  $(\text{price}_j * \text{workload}_i / \text{ability}_j)$  is the monetary cost required for task  $i$ .

In Algorithm 1, there are  $n$  tasks and  $m$  resources (lines (1)-(2)). Each user starts an auction in order and calculates the product of completion time and monetary cost for every resource (lines (3)-(10)) and then selects the resource with the minimum product (line (11)). Then user pays to the winner (line (12)). At last, all providers change price according to dynamic pricing strategy and join the next auction (line (13)).

When workflows are submitted and tasks start auctions, providers give their bids to compete for the opportunity of providing resources. In the auction, providers change price and increase chances of selling resource, so that they can gain more revenue and higher resource utilization. In addition, users always select the optimal resource, so the product of the completion time and monetary cost of executing tasks is the minimum.

## 6. Evaluation

In this section, experiments are conducted for evaluation of the performance of BOSS and DPAM on resource utilization and the measurement of TC with different situations and problem sizes. Moreover, the performance of DPAM on resource utilization (see Formula (2)) and TC with different price reduction rates is verified. Firstly, experiment setup is given (see Section 6.1). Secondly, simulation of specific workflow is described for evaluating BOSS and DPAM (see Section 6.2). Thirdly, we conduct experiments with the medium problem size and evaluate BOSS and DPAM with different situations and the performance of DPAM with different price reduction rates (see Section 6.3). At last, both from different situations and different problem sizes, simulation results show the performance of BOSS and DPAM and the performance of DPAM with different price reduction rates (see Section 6.4).

```

Input: workflows and resources
Output: allocation of tasks on resources
(1)  $tasks \leftarrow [n]$ ; /* Assign the tasks to  $tasks$  list with partially relation */
(2)  $resources \leftarrow [m]$ ; /* Assign the resources to  $resources$  list */
(3)  $i = 1$ ;
(4) While  $i \leq n$  do
(5)    $tempTask \leftarrow ith$  task;
(6)    $j = 1$ ;
(7)   While  $j \leq m$  do
(8)      $tempResource \leftarrow jth$  resources;
(9)      $TCs \leftarrow CalculateTCs(tempTask, tempResource)$ ;
        /* calculate TC of  $tempTask$  on  $tempResource$  (Formula (3)) */
(10)   End
(11)   $winner \leftarrow resourceWithMinTC(TCs)$ ;
        /* select the optimal resource with the minimum TC */
(12)   $tempTask$  pays to  $winner$ ;
(13)  all providers  $changePrice$ ;
        /* change resource price (Formula (1)) */
(14) End

```

ALGORITHM 1: Dynamic pricing based allocation mechanism.

TABLE 6: Problem size classification.

Small	Medium	Large
$1 \leq n \leq 40$	$50 \leq n \leq 100$	$200 \leq n \leq 300$
$1 \leq m \leq 10$	$10 \leq m \leq 50$	$80 \leq m \leq 120$

**6.1. Experiment Setup.** The simulation environment runs on a PC with the following configurations: 2 CPU cores, 4 GB RAM, and Microsoft Windows 7 OS. The workflows are classified into three situations: balanced, semibalanced, and unbalanced [12]. Task workload follows normal distribution  $N(1000000, 1000)$ . The resource ability is set from 200 to 1200 with an arithmetic sequence and the common difference is quotient of 1000 divided by task amount. The resource price is set from the real Amazon Web Services price (<https://aws.amazon.com/>). In BOSS resource price is set from 0.14 to 0.84 per time unit. In DPAM, to implement dynamic pricing strategy, all resources have starting prices and reserve prices. The starting price is set from 0.14 to 0.84 and reserve price is set from 0.1 to 0.6, respectively.

In simulation of specific cloud workflows, the workflow has 10 tasks and the amount of resources is 7. The price reduction rate for DPAM is 10%. In simulation of general cloud workflows, they are classified into small, medium, and large by problem size besides different situations. Problem size classification is shown in Table 6, where  $n$  is amount of tasks and  $m$  is amount of resources. In addition, the price reduction rate is set from 0 to 1 in step of 0.1.

**6.2. Simulation of Specific Workflows.** In specific experiment, specific workflows are used to verify whether DPAM performs better than BOSS on resource utilization and TC.

As shown in Figure 3(a), resource utilization of DPAM is always higher than that of BOSS. DPAM can improve resource utilization compared with BOSS. This is because

providers with low competitiveness change their resource prices and then these resources have more chances to be sold.

In Figure 3(b), three different situations of TCs of DPAM are all lower than those of BOSS. This means that it takes shorter time and lower monetary cost for workflow execution. In DPAM, providers decrease their resource prices to improve the competitiveness. So users can get the resource with shorter completion time or lower monetary cost.

**6.3. Simulation of General Workflows with Different Balanced Situations.** In this section, two experiments are conducted on general workflows with different balanced situations. The problem size is medium. The first experiment simulates BOSS and DPAM to evaluate their performance on resource utilization and TC (see Figure 4). The second experiment simulates DPAM with different price reduction rates to evaluate its performance on resource utilization and TC (see Figure 5).

**6.3.1. Resource Utilization and TC of BOSS versus DPAM.** As shown in Figure 4(a), resource utilization of DPAM is always higher than BOSS. This indicates that DPAM performs better in resource utilization. In DPAM, more resources are sold by changing prices especially for resources with lower competitiveness. These resources are never sold in BOSS. Figure 4(b) shows that TC of DPAM is lower than that of BOSS. DPAM brings shorter completion time and lower monetary cost. The reason is that resource price is dynamic and then there are more resources with higher computation ability and lower price.

**6.3.2. Resource Utilization and TC of DPAM with Different Price Reduction Rates.** Figure 5 shows resource utilization and TC of DPAM with different price reduction rates. In Figure 5(a) resource utilization is constant when price

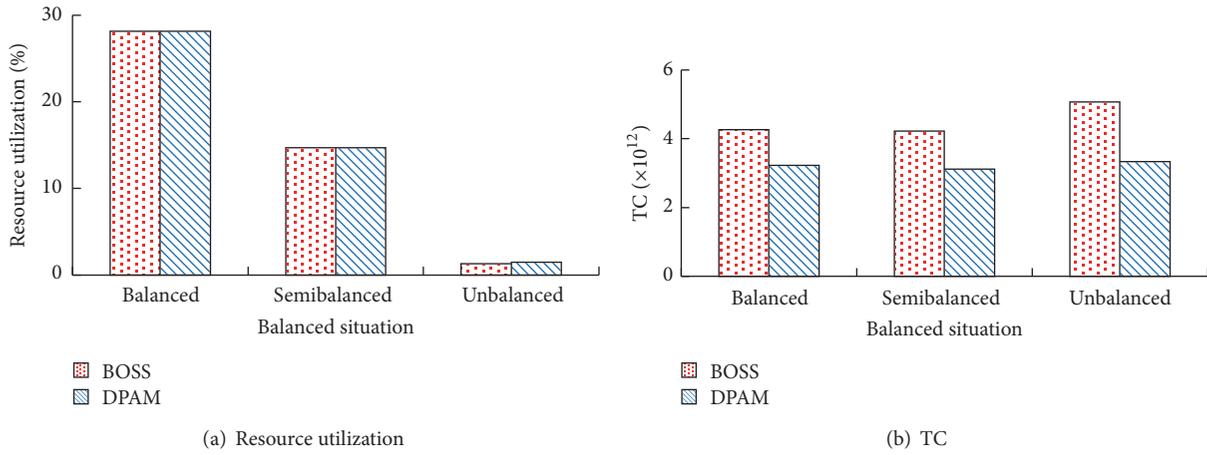


FIGURE 3: Resource utilization and TC of BOSS versus DPAM for specific workflow.

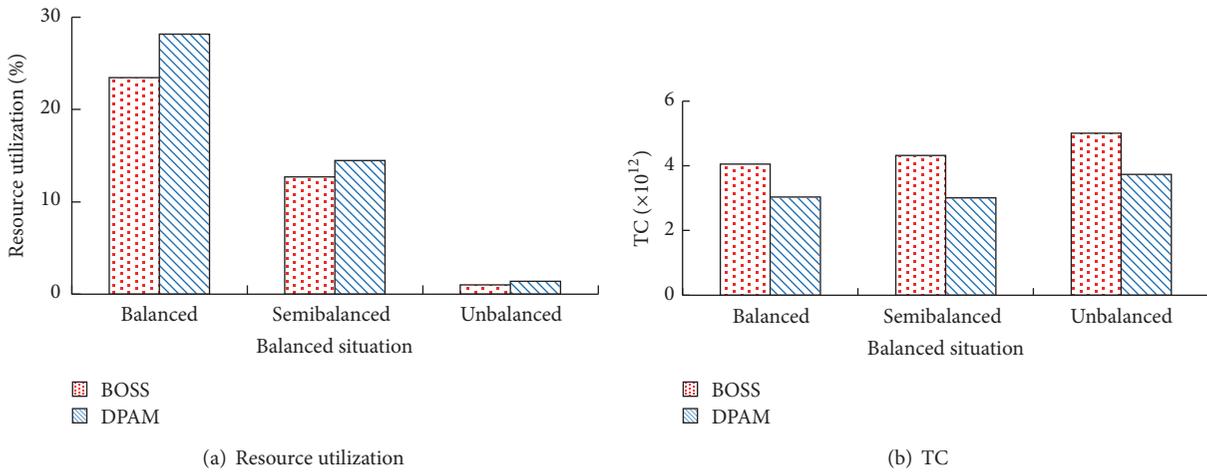


FIGURE 4: Resource utilization and TC of BOSS versus DPAM for general workflows.

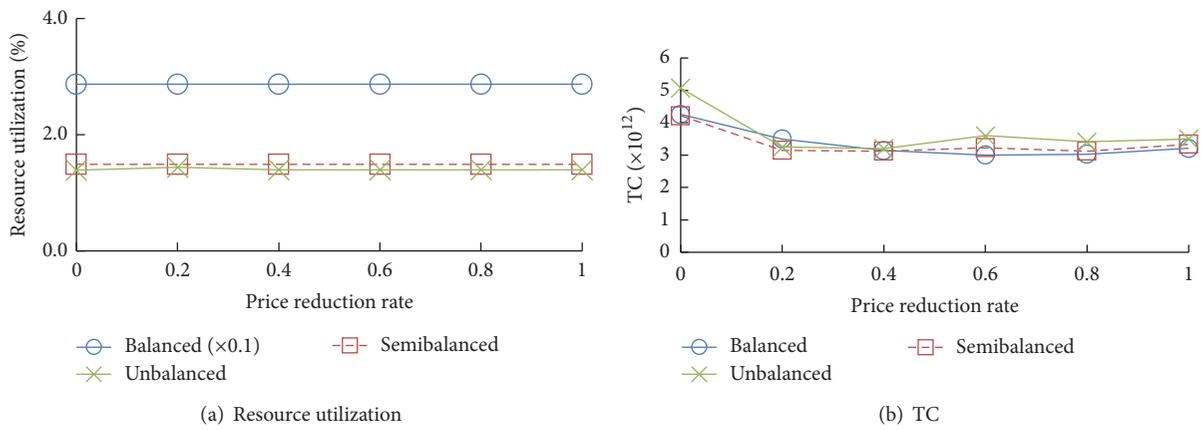


FIGURE 5: Resource utilization and TC of DPAM with different price reduction rates.

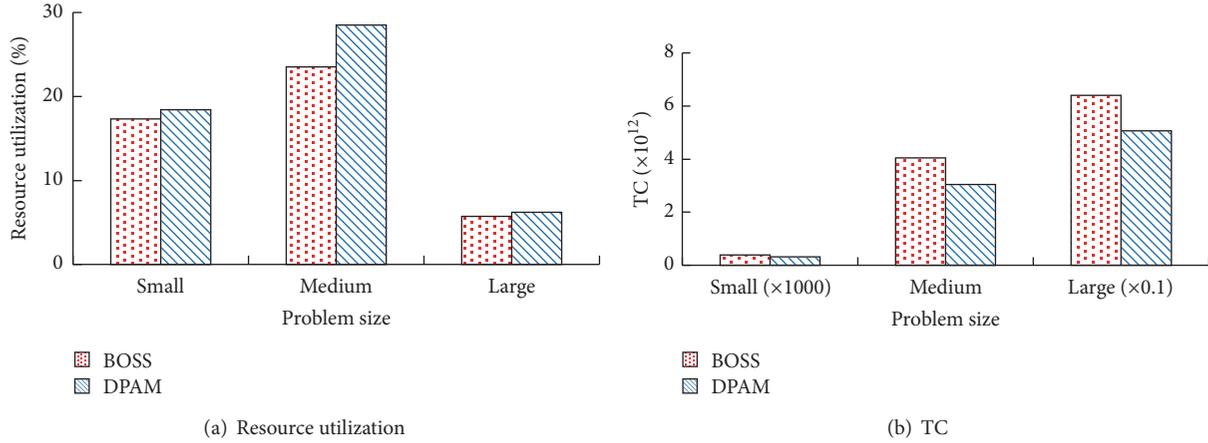


FIGURE 6: Resource utilization and TC of BOSS versus DPAM in balanced situation.

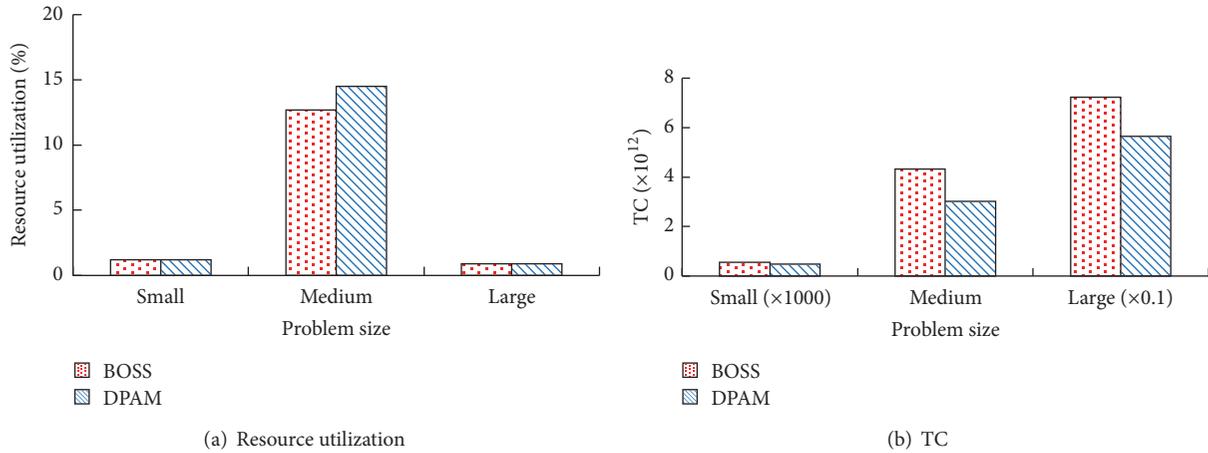


FIGURE 7: Resource utilization and TC of BOSS versus DPAM in semibalanced situation.

reduction rate is bigger than 0.2. This is because the resource price is equal to the reserve price when price reduction rate is high enough. As shown in Figure 5(b), TC of workflows with all situations decreases when price reduction rate is not zero. It is easy to draw that DPAM is better than BOSS.

6.4. Simulation of General Workflows with Different Problem Sizes. In this subsection, another two sets of experiments conducted on general workflows with different problem sizes and balanced situations are described. The first set of experiments simulates BOSS and DPAM to evaluate their performance on resource utilization and TC (see Figures 6–8). The second set of experiments simulates DPAM with different price reduction rates to evaluate the performance on resource utilization and TC (see Figures 9–11).

6.4.1. Resource Utilization and TC of BOSS versus DPAM. Figures 6–8 present the performance of BOSS and DPAM on resource utilization and TC from different balanced situations and different problem size. In Figures 6(a), 7(a), and 8(a), resource utilization of balanced workflow is higher than that of unbalanced workflow. This is because more tasks

in balanced workflow are executed in parallel and many resources are used. In three situations, resource utilizations of DPAM are all higher than that of BOSS. Figures 6(b), 7(b), and 8(b) show that TC of DPAM is always lower than that of BOSS. The reason is that the resource with lower price or higher computation ability is selected as winner.

6.4.2. Resource Utilization and TC of DPAM with Different Price Reduction Rates. Figures 9(a), 10(a), and 11(a) show that resource utilization changes only when price reduction rate is lower than 0.3. This indicates that it is not necessary to make price reduction rate too high. The reason is that resource price cannot be smaller than reserve price. Figures 9(b), 10(b), and 11(b) show that TC decreases when resource price reduces in some rates. And TC of large problem size workflows decreases apparently than other sizes. This is because dynamic pricing brings more competitive resources with lower price and higher ability.

In overall terms, the performance of DPAM on resource utilization and TC with different situations is better than BOSS shown in Figure 4. The performance of DPAM on resource utilization and TC with different problem sizes is

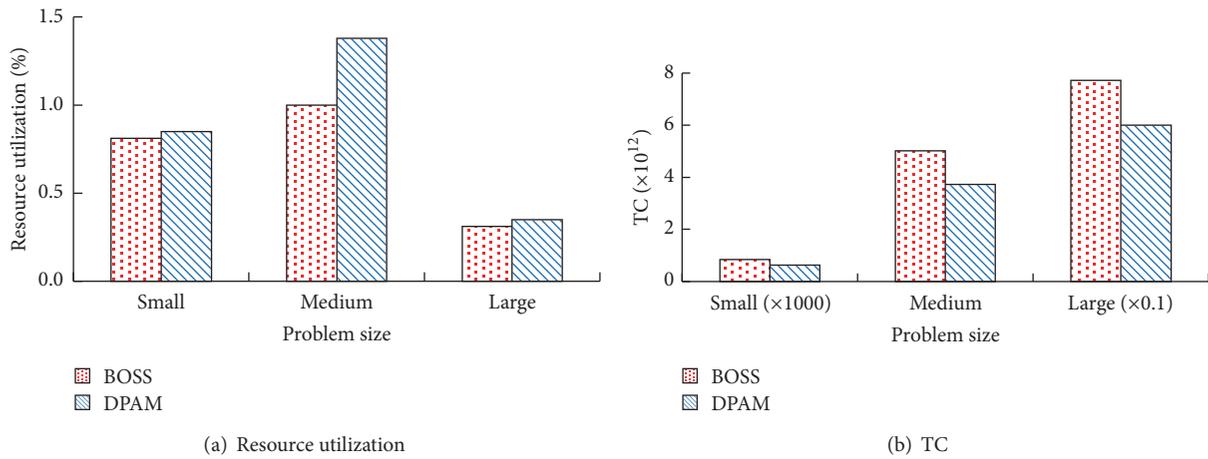


FIGURE 8: Resource utilization and TC of BOSS versus DPAM in unbalanced situation.

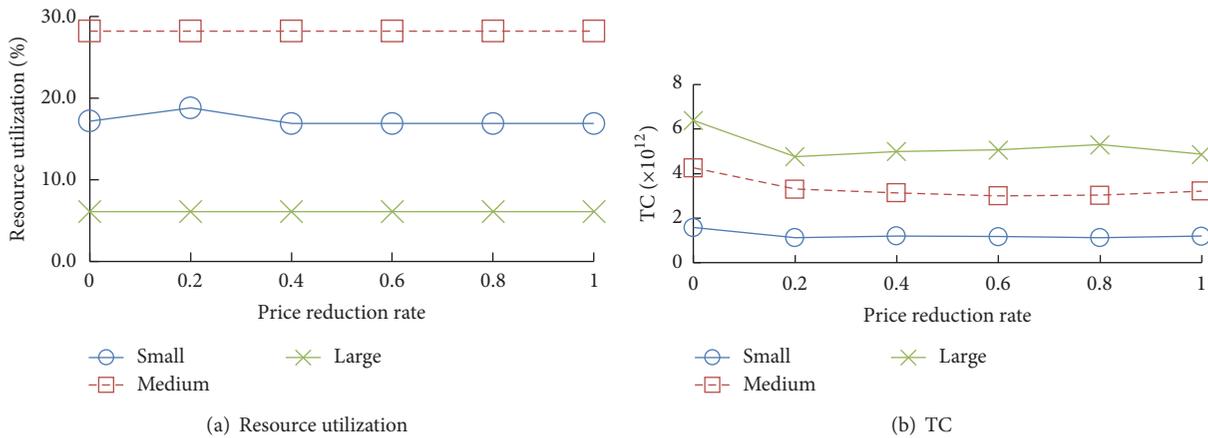


FIGURE 9: Resource utilization and TC with different rates in balanced situation.

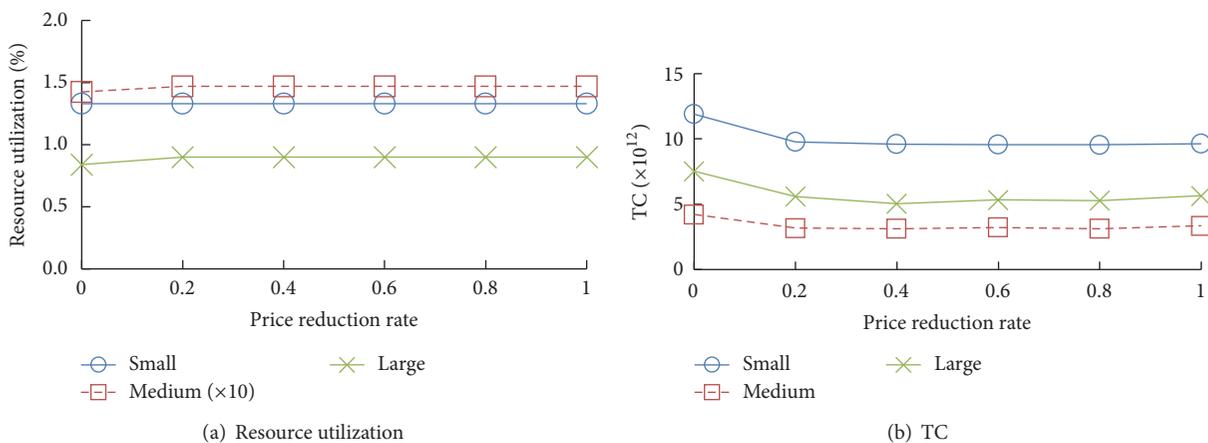


FIGURE 10: Resource utilization and TC with different rates in semibalanced situation.

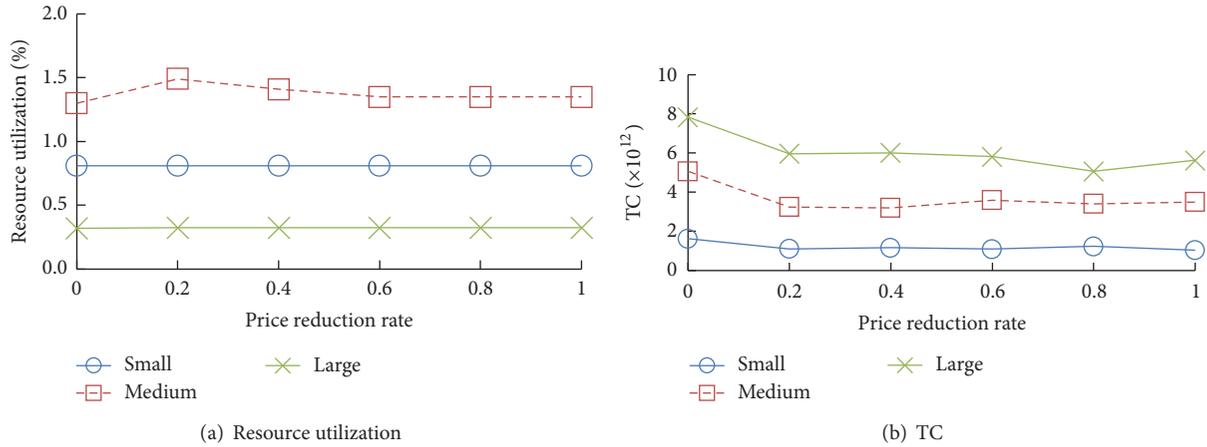


FIGURE 11: Resource utilization and TC with different rates in unbalanced situation.

shown in Figures 6–8. In DPAM, many providers with weak competitiveness use dynamic pricing strategy to increase chances of making a deal and gain more revenue, so resource utilization of market increases. Meanwhile, workflows can execute timely with less cost. So the performance of DPAM on resource utilization and TC is better than that of BOSS. Moreover, the performance of DPAM on resource utilization and TC with different price reduction rates is shown in Figures 5 and 9–11. Resource utilization and TC are invariant when price reduction rate is higher than 0.2. This is because resource price cannot be lower than the reserve price. In addition, performance of TC and resource utilization is always better when price reduction rate is bigger than zero.

## 7. Conclusion and Future Work

In this paper, we proposed a dynamic pricing strategy to improve resource providers' competitiveness in the cloud market. A novel dynamic pricing based allocation mechanism was presented to allocate resources for cloud workflows. With our mechanism, resource providers can change the price to increase the possibility of selling resources and gain more revenue, which improves resources utilization. The users select the best resource with the minimum TC (Time \* Cost), which ensures shorter completion time and lower monetary cost. Finally, we evaluated our mechanism and compared with the representative BOSS strategy. The results showed that our mechanism can achieve high resources utilization, shorter completion time, and lower monetary cost. With the dynamic pricing strategy, providers can decrease their resource price to improve competitiveness.

In future, increasing price will be involved in dynamic pricing strategy. It is a good way for those resource providers who have sharply higher competitiveness to increase price to gain more revenue. At the same time, we will use the standard scientific datasets to run experiments besides random data. This will increase the credibility of the results of the experiment and be more scientific to reflect the performance of the DPAM mechanism. In addition, besides completion time and monetary cost, we will consider adding other QoS

criteria such as reliability, response time, and service providers' reputation.

## Competing Interests

There is no conflict of interests related to this paper.

## Acknowledgments

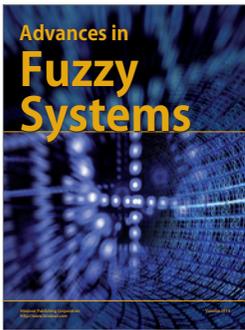
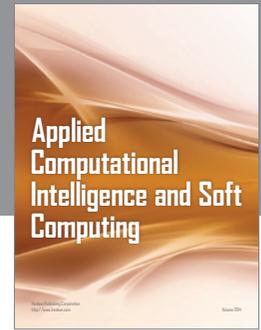
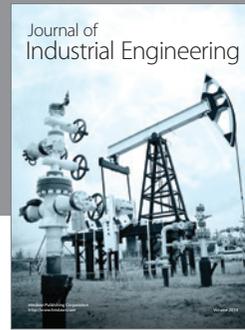
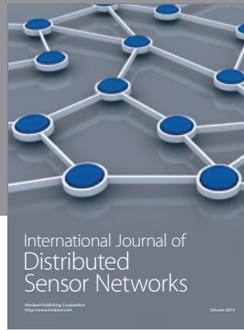
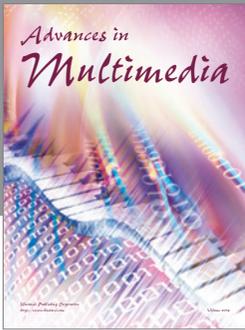
This work is partially supported by Natural Science Foundation of China under nos. 61672034, 61300042, and 61300169, MOE Project of Humanities and Social Sciences under no. 16YJCZH048, and the Key Natural Science Foundation of Education Bureau of Anhui Province Project KJ2016A024. The authors are grateful for Professor Yun Yang from Swinburne University of Technology, Australia, for providing constructive feedback to improve this paper. The price reduction rate is set by empirical knowledge. Therefore, the rational rate deserved to be researched.

## References

- [1] J. Wang, M. AbdelBaky, J. Diaz-Montes, S. Purawat, M. Parashar, and I. Altintas, "Kepler + cometcloud: dynamic scientific workflow execution on federated cloud resources," *Procedia Computer Science*, vol. 80, pp. 700–711, 2016.
- [2] G. Juve and E. Deelman, "Scientific workflows and clouds," *Crossroads*, vol. 16, no. 3, pp. 14–18, 2010.
- [3] A. Prasad, P. Green, and J. Heales, "On governance structures for the cloud computing services and assessing their effectiveness," *International Journal of Accounting Information Systems*, vol. 15, no. 4, pp. 335–356, 2014.
- [4] C. Lin and S. Lu, "Scheduling scientific workflows elastically for cloud computing," in *Proceedings of the 2011 IEEE 4th International Conference on Cloud Computing (CLOUD '11)*, pp. 746–747, Washington, DC, USA, July 2011.
- [5] T. T. Huu and C. K. Tham, "An auction-based resource allocation model for green cloud computing," in *Proceedings of the IEEE International Conference on Cloud Engineering (IC2E '13)*, pp. 269–278, San Francisco, Calif, USA, March 2013.

- [6] V. Prasad G, S. Rao, and A. S. Prasad, "A combinatorial auction mechanism for multiple resource procurement in cloud computing," in *Proceedings of the 12th International Conference on Intelligent Systems Design and Applications (ISDA '12)*, pp. 337–344, Kochi, India, November 2012.
- [7] M. A. Rahman and R. M. Rahman, "CAPMAuction: reputation indexed auction model for resource allocation in Grid computing," in *Proceedings of the 7th International Conference on Electrical and Computer Engineering (ICECE '12)*, pp. 651–654, IEEE, Dhaka, Bangladesh, December 2012.
- [8] X. Weng, X. Wang, C.-L. Wang, K. Li, and M. Huang, "Resource allocation in cloud environment: a model based on double multi-attribute auction mechanism," in *Proceedings of the 6th IEEE International Conference on Cloud Computing Technology and Science (CloudCom '14)*, pp. 599–604, December 2014.
- [9] C. N. Boyer and B. W. Brorsen, "Implications of a reserve price in an agent-based common-value auction," *Computational Economics*, vol. 43, no. 1, pp. 33–51, 2014.
- [10] H. Qu, I. O. Ryzhov, and M. C. Fu, "Learning logistic demand curves in business-to-business pricing," in *Proceedings of the 43rd Winter Simulation Conference: Simulation: Making Decisions in a Complex World (WSC '13)*, pp. 29–40, Washington, DC, USA, December 2013.
- [11] A. S. Prasad and S. Rao, "A mechanism design approach to resource procurement in cloud computing," *IEEE Transactions on Computers*, vol. 63, no. 1, pp. 17–30, 2014.
- [12] H. M. Fard, R. Prodan, and T. Fahringer, "A truthful dynamic workflow scheduling mechanism for commercial multicloud environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1203–1212, 2013.
- [13] B. Sharma, R. K. Thulasiram, P. Thulasiraman, S. K. Garg, and R. Buyya, "Pricing cloud compute commodities: a novel financial economic model," in *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid '12)*, pp. 451–457, IEEE, Ottawa, Canada, May 2012.
- [14] X. Li, X. Liu, and E. Zhu, "An efficient resource allocation mechanism based on dynamic pricing reverse auction for cloud workflow systems," in *Proceedings of the Asia-Pacific Conference on Business Process Management*, pp. 59–69, 2015.
- [15] H. Xu and B. Li, "Resource allocation with flexible channel cooperation in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 957–970, 2013.
- [16] T. Wood, P. J. Shenoy, A. Venkataramani, and M. S. Yousif, "Black-box and gray-box strategies for virtual machine migration," in *Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation*, pp. 229–242, 2007.
- [17] K. Görlach and F. Leymann, "Dynamic service provisioning for the cloud," in *Proceedings of the IEEE 9th International Conference on Services Computing (SCC '12)*, pp. 555–561, June 2012.
- [18] X. Shi and Y. Zhao, "Dynamic resource scheduling and workflow management in cloud computing," in *Proceedings of the International Conference on Web Information Systems Engineering*, pp. 440–448, 2010.
- [19] M. Mao and M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '11)*, pp. 1–12, ACM, Seattle, Wash, USA, November 2011.
- [20] J. Wang, P. Korambath, I. Altintas, J. Davis, and D. Crawl, "Workflow as a service in the cloud: architecture and scheduling algorithms," *Procedia Computer Science*, vol. 29, pp. 546–556, 2014.
- [21] L. Wang, J. Shen, and J. Yong, "A survey on bio-inspired algorithms for web service composition," in *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD '12)*, pp. 569–574, Wuhan, China, May 2012.
- [22] L. Wang and J. Shen, "Multi-phase ant colony system for multi-party data-intensive service provision," *IEEE Transactions on Services Computing*, vol. 9, no. 2, pp. 264–276, 2016.
- [23] S. A. Ludwig, "Particle swarm optimization approach with parameter-wise hill-climbing heuristic for task allocation of workflow applications on the cloud," in *Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '13)*, pp. 201–206, IEEE, Herndon, Va, USA, November 2013.
- [24] D. Li, C. Chen, J. Guan, Y. Zhang, J. Zhu, and R. Yu, "DCloud: deadline-aware resource allocation for cloud computing jobs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 8, pp. 2248–2260, 2016.
- [25] H. Wang, Z. Kang, and L. Wang, "Performance-aware cloud resource allocation via fitness-enabled auction," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1160–1173, 2016.
- [26] M. M. Nejad, L. Mashayekhy, and D. Grosu, "Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 594–603, 2015.
- [27] F. Teng and F. Magoules, "Resource pricing and equilibrium allocation policy in cloud computing," in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology*, pp. 195–202, 2010.
- [28] M. Mihalescu and Y. M. Teo, "On economic and computational-efficient resource pricing in large distributed systems," in *Proceedings of the 10th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, pp. 838–843, Melbourne, Australia, May 2010.
- [29] L. Pham, J. Teich, H. Wallenius, and J. Wallenius, "Multi-attribute online reverse auctions: recent research trends," *European Journal of Operational Research*, vol. 242, no. 1, pp. 1–9, 2015.
- [30] M. Takeda, D. Takahashi, and M. Shobayashi, "Collective action vs. conservation auction: lessons from a social experiment of a collective auction of water conservation contracts in Japan," *Land Use Policy*, vol. 46, pp. 189–200, 2015.
- [31] C. Xu, L. Song, Z. Han et al., "Efficiency resource allocation for device-to-device underlay communication systems: a reverse iterative combinatorial auction based approach," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 348–358, 2013.
- [32] P. Setia and C. Speier-Pero, "Reverse auctions to innovate procurement processes: effects of bid information presentation design on a supplier's bidding outcome," *Decision Sciences*, vol. 46, no. 2, pp. 333–366, 2015.
- [33] J. R. Fooks, K. D. Messer, and J. M. Duke, "Dynamic entry, reverse auctions, and the purchase of environmental services," *Land Economics*, vol. 91, no. 1, pp. 57–75, 2015.
- [34] W. Depoorter, K. Vanmechelen, and J. Broeckhove, "Advance reservation, co-allocation and pricing of network and computational resources in grids," *Future Generation Computer Systems*, vol. 41, pp. 1–15, 2014.

- [35] Y. Zhao, Y. Li, I. Raicu, S. Lu, W. Tian, and H. Liu, “Enabling scalable scientific workflow management in the Cloud,” *Future Generation Computer Systems*, vol. 46, pp. 3–16, 2015.
- [36] M. Mihailescu and Y. M. Teo, “Strategy-proof dynamic resource pricing of multiple resource types on federated clouds,” in *Algorithms and Architectures for Parallel Processing*, C.-H. Hsu, L. T. Yang, J. H. Park, and S.-S. Yeo, Eds., vol. 6081 of *Lecture Notes in Computer Science*, pp. 337–350, Springer, Berlin, Germany, 2010.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

