This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2633288, IEEE Access

A Multi-objective Hybrid Cloud Resource Scheduling Method Based on Deadline and Cost Constraints

Liyun Zuo*†, Lei Shu*‡, Shoubin Dong†, Yuanfang Chen*, Li Yan*
*Guangdong Provincial Key Lab. of Petrochemical Equipment Fault Diagnosis,
Guangdong University of Petrochemical Technology, China
†School of Computer Science and Engineering, South China University of Technology, China
‡School of Engineering, University of Lincoln, UK
Email: {liyun.zuo, lei.shu}@lab.gdupt.edu.cn, sbdong@scut.edu.cn, yuanfang_chen@ieee.org,
li.yan0823@outlook.com

***Abstract*—We propose a task-oriented multi-objective scheduling method based on ant colony optimization (MOSACO) to optimize the finite pool of public and private computing resources in a hybrid cloud computing environment according to deadline and cost constraints. MOSACO is employed to minimize task completion times and costs using time-first and cost-first single-objective optimization strategies, respectively, and to maximize user quality of service and the profit of resource providers using an entropy optimization model. The effectiveness of the MOSACO algorithm based on multiple considerations of task completion time, cost, number of deadline violations, and degree of private resource utilization is verified using simulation and three application examples. Comparisons with similar scheduling methods demonstrate that MOSACO provides the highest optimality, and that the time-first and cost-first strategies provide definite advantages for minimizing completion time and cost, respectively.**

***Index Terms*—Hybrid cloud; Resource utilization; Task scheduling; Deadline; Cost constraint**

I. Introduction

Cloud computing presents a number of challenges regarding the management of tasks and resources, such as cost constraints and completion time requirements. These challenges are even more prominent for a hybrid cloud computing environment that combines relatively inexpensive but low-performance private cloud services with relatively high-cost and high-performance public cloud services. Therefore, it is necessary to determine which tasks are more cost effective in a local private cloud, and which are more cost effective in the public cloud.

With the growth of enterprise applications and the growing volume of high-speed data traffic, particularly during peak hours, existing enterprise clusters have struggled to provide sufficient computing resources to satisfy demand. In turn, it has become necessary to increase hardware investments significantly. The primary problem is in utilization, where most hardware lies idle except at peak time periods. Thus, enterprise management is confronted with both idle resources and increased maintenance costs. To reduce the burdens of hardware investment and maintenance costs, an increasing number of organizations have adopted a hybrid cloud computing environment as the deployment platform of choice for their enterprise applications [1]. As such, these organizations employ their local private cloud, and then directly access the higher computational power of public cloud services when the computing ability of the private cloud cannot meet their computational requirements. The need for public cloud services varies, where some enterprise applications require more computing power and storage capacity than that available from a private cloud, or private computing recourses may be insufficient to meet the demand during peak periods. This mixture of computing resources involves unavoidable scheduling problems. Numerous solutions to these problems have been proposed, such as the management of load [2], or divided and optimized on data [3]. Nonetheless, whether scheduling is based on load or data applications, the scheduling problem has become a key issue. Optimum scheduling must determine those tasks that should be allocated to the public cloud, and those that can best be conducted locally in the private cloud. In addition, optimum scheduling must maximize the utilization of relatively low-cost private cloud resources while minimizing the cost of public cloud services, and meet the time and cost constraints of these two goals. In addition, cloud computing should not only meet the quality of service (QoS) needs of users, but also consider the interests of resource providers such as resource income and low resource idle rate. As such, both sides must be considered when exploring solutions to scheduling problems.

To address these challenges, this paper proposes a task-oriented multi-objective scheduling method based on ant colony optimization (MOSACO) to optimize the finite pool of public and private computing resources in a hybrid cloud computing environment. The proposed method considers task completion times, costs, user QoS, and resource provider interests under cost and deadline constraints. The main contributions of this paper are as follows.

- First, the proposed (MOSACO) method applies task completion time and cost priorities according to cost and deadline constraints, respectively, to maximize the utilization of private cloud resources and minimize the cost of public cloud resources as much as possible. Simultaneously, the method also maximizes the interests of both users and resource providers.

- Second, we propose two single-objective scheduling optimization strategies that respectively prioritize cost and task completion time. For simultaneous consideration of both user deadline and cost constraints, the performance requirements of high-priority users can be met by scheduling to public cloud resources, while the cost requirements of high-demand users can be met by scheduling to private cloud resources.
- Third, the proposed (MOSACO) method applies an improved ant colony optimization algorithm to balance all multi-objective requirements.

The rest of the paper is organized as follows. The related work is introduced in Section II, and the description and definition of system model including hybrid cloud resources and tasks, two single objective optimization scheduling strategies, entropy optimization model and multi-objective optimization scheduling problem are shown in Section III. An improved ant colony algorithm to solve the multi-objective optimization problem is in Section IV. Experiments are shown in Section V, and the paper is concluded in Section VI.

II. Related Work

Here, past research on hybrid cloud scheduling is mainly discussed from the perspectives of single-objective optimization, multi-objective optimization, and multi-objective ant colony optimization.

A. Single-objective Optimization Scheduling

A task scheduling method focused on the scheduling problems of public and private cloud resources has been proposed based on a genetic algorithm, but the method did not take into account the restrictions of resources. Some existing research has focused on the objective optimization problem constraints of bills on deadline. An outsourcing scheduling model has been proposed from the perspective of cloud users to minimize costs, but the model focused on public cloud cost optimization problems, and involved restrictions on the scale of scheduling solutions. Part of the application of the critical path has been proposed for a cost-driven workflow scheduling method, and two types of scheduling optimization strategies were developed based on hybrid cloud architectures, but the study mainly focused on cost analysis. An optimization of the task scheduling system has been proposed that dedicates as few resources as possible to meet user QoS requirements and reduced costs, where public cloud computing resources compensate for insufficient local resources, and transform existing local resources into a hybrid cloud model. Here, the cloud agent sends a task to multiple cloud providers, so as to minimize the users budget costs.

B. Multi-objective Optimization Scheduling

Researchers have also studied multi-objective optimization scheduling based on considerations of QoS constraints, economic costs, energy consumption, and system performance. Considering computational resources, a scheduling method was proposed that divided the resources and budget costs to minimize the length of tasks, and thereby reduce task completion times and improve system resource utilization [20]. One of the earliest completion time replication algorithms was proposed for task-based replication [21]. First, the algorithm adopts fuzzy clustering for preliminary preprocessing of resources, and then applies a directed acyclic graph and task duplication scheduling. Considering task execution times, resource costs and utilization rate have been considered in a cloud environment using a multi-input, multi-output feedback control dynamic resource scheduling algorithm to ensure implementation under time constraints for optimal application performance [22]. Aiming at uncertainty factors in a hybrid cloud environment, two dynamic resource allocation algorithms using the Pareto optimization method have been proposed based on cost and deadline constraints [23]. However, the time complexities of the two algorithms were high, and were both greater than or equal to O(n2). An adaptive workflow scheduling heuristic method that takes into account cost and time constraints has been proposed, although the approach schedules only the analysis of data workflow in a hybrid cloud environment [24]. A multi-objective scheduling method has been proposed based on a time and cost optimization objective with the bandwidth and storage as the constraints [25]. The method focused on increasing the utilization of private cloud resources to achieve a balance between performance and costs. Three studies [1], [26], and [28], have regarded the scheduling problem in a manner similar to the focus of the present work. To address optimization deadlines and infrastructure as a service (IaaS) provider benefits, an adaptive particle swarm optimization hybrid cloud scheduling algorithm was developed [1]. However, this method only considered the benefits of cloud providers without any consideration for cost from the standpoint of users. In addition, researchers have proposed that the focus should be on improving the overall system performance regardless of whether discussing public or private cloud resources [26]. Finally, the outsourcing of tasks to a public cloud model was proposed to minimize the cost of outsourcing while simultaneously maximizing the utilization rate of an internal cloud data center [28]. Accordingly, the study adopted mathematical programming for optimized scheduling; however, this approach was unable to solve scheduling problems involving large-scale data, and its optimization objectives were costly. A cloud resource management program has been proposed based on similar goals of maximizing resource utilization and minimizing costs [29]. However, the approach was mainly used to migrate on and off a virtual machine, and was not applied for the actual case of task scheduling optimization.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2633288, IEEE Access

C. Multi-objective Optimization by Ant Colony Approaches

The ant colony algorithm has been applied to cloud computing in many studies for multi-objective optimization scheduling, taking into account the completion time, QoS constraints, load balancing, availability, and cost. One such study [30] proposed an ant colony optimization scheduling algorithm that, in addition to minimizing the optimal span (Makespan), also considered how to balance the load of a virtual machine, and realized load balance by minimizing the virtual machine during idle times. Another study [31] proposed an improved ant colony algorithm based on the shortest delay time, taking into account fairness and efficiency in scheduling tasks. Another study [32] proposed an enhanced ant colony optimization algorithm based on the shortest completion time and load balancing, which searched for objects to set a task with virtual resource matching. In addition, a study [33] proposed an ant colony optimization algorithm for obtaining the multiple QoS parameters of a big workflow scheduling problem, where users selected a specified service preference and QoS threshold within the scope of budget cost constraints. However, the relevance of this approach was limited to oriented workflow scheduling in grid computing. Finally, another study [34] proposed a method for determining scheduling QoS constraints via a fusion of genetic and ant colony algorithms, taking into account time and cost, but with less attention to optimization constraints. There are other relevant researches [35-41].

The above discussion illustrates the limitations associated with existing research in hybrid cloud scheduling. These limitations can be given as follows. (1) Some studies investigated scheduling problems from the perspective of only private cloud or public cloud resources. (2) Many existing studies focused exclusively on the cost of hybrid clouds or the performance of a single index of optimization. (3) Some studies focused on both cost and performance balancing with consideration for the cost benefits accruing to either resource providers or users, while neglecting the collective interests of both. (4) The consideration of mission cost tends to underestimate the importance of mission resource requirements. However, mission costs tend to vary considerably owing to the diversity in the costs associated with cloud resources and tasks. Therefore, mission resource requirements must be considered when evaluating mission cost.

III. System Model of a Hybrid Cloud Environment

To clearly describe the scheduling problems inherent in hybrid cloud environments, this paper proposes the system model given in Fig. 1. According to the model, users submit requests for task and resource requests to the request manager through a user interface. Request information includes task size, required amount of data, deadline, and budget cost. Then, the task manager transmits the request information to the scheduling manager. The scheduling manager is the primary component of this model. The scheduler deploys all scheduling strategies, such as all types of application performance priority strategies, under deadline and cost constraints. The scheduling manager has public and private cloud resource information, such as the computing power of public and private resources, data transport capacity, and computing and data transport costs. The scheduling manager then allocates a task to public or private cloud resources according to the mission requirements, combining resource information according to the scheduling method employed. The computational results are then returned to the user.

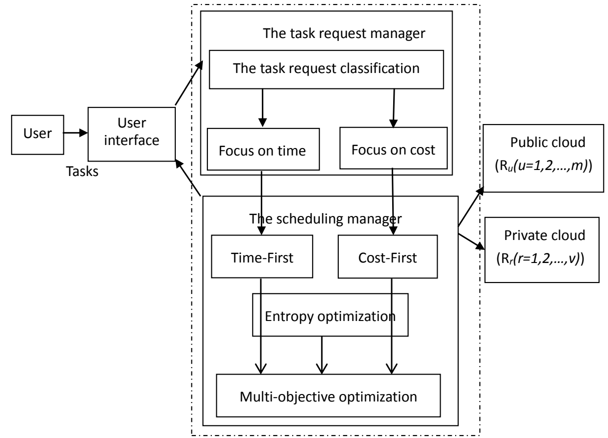

Fig. 1. The system structure model.

TABLE I
MAIN NOTATION DEFINITIONS

Symbol	Definitions
R_u	The public cloud resource, $u = 0, 1, 2, , m - 1$
C_u	The computing power of R_u
L_u	The transmission capacity of R_u
P_u	The computing price of R_u
S_u	The store price of R_u
Q_u	The transmission price of R_u
R_r	The public cloud resource, $r = 0, 1, 2, , k - 1$
C_r	The computing power of R_r
L_r	The transmission capacity of R_r
T_i	A unit of the task request
TD_i	The deadline of T_i
TC_i	The size of T_i
TL_i	The data amount of T_i
TM_i	The budget cost of T_i
t_iR_u, t_iR_r	The completion time of T_i in public and private cloud
F_i	The cost of T_i in public cloud
λ	The threshold of similarity
D_v	The deadline violation
n_v	The number of tasks violated the deadline

A. Description of the Scheduling Problem Based on the Model

To clearly understand the models functions, it is necessary to describe and define the resources, tasks, and constraint conditions related to scheduling in the hybrid cloud. The nomenclature employed in this paper is listed in Table I.

B. Description of Task and Constraint Conditions

We first present definitions of public and private cloud scheduling resources in the hybrid cloud. Here, the virtual machine is the primary resource.

Definition 1, public cloud resource: According to the model of Amazon's cloud service, the public cloud resource R_u is defined as follows.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2633288, IEEE Access

$$R_u =< C_u, L_u, P_u, S_u, Q_u > \tag{1}$$

Here, C_u is the computing power of R_u, which affects the computing time; L_u is the transmission capacity, which influences the transmission time of a task; P_u is the computing cost; S_u is the data storage cost; Q_u is the transmission cost, which is primarily the outside transmission cost because inside transmission is free in Amazon's cloud service.

Definition 2, private cloud resource: A private cloud resource R_r only considers the computing power C_r and the transmission capacity L_r, without considering the computing, storage, and transmission costs P_r, S_r, and Q_u, respectively, and is defined as follows.

$$R_r =< C_r, L_r > \tag{2}$$

This is very similar to definition 1.

Definition 3, It is assumed that a task T_i is a unit of a task request, and each resource considers only one task at a time. Then, T_i is defined as follows.

$$T_i =< TD_i, TC_i, TL_i, TM_i > \tag{3}$$

Here, TD_i is the deadline of T_i, where T_i must be completed and the results returned prior to TD_i; otherwise, T_i will be regarded as a failure; TC_i is the size of T_i, which represents the code length associated with T_i, and affects its completion time; TL_i is the amount of data associated with T_i, which directly affects its transmission time; TM_i is the budget cost of T_i, which sets the limit on the cost of T_i when completed on a public cloud.

According to definition 3, two constraint conditions are employed in task scheduling, which are TD_i and TM_i. To meet these two constraint conditions, it is necessary to calculate the task completion time and TM_i in the scheduling model of the hybrid cloud. The task completion time consists of two parts, the computing time and transmission time. Here, the task completion time is not only the completion time in the public cloud, denoted as t_iR_u, but also in the private cloud, denoted as t_iR_r. According to the definitions of resources and task, these two completion times are respectively calculated as formulas 4 and 5.

The constraint conditions about deadline are $t_iR_u \le TD_i$ and $t_iR_r \le TD_i$.

$$t_iR_u = \frac{TC_i}{C_u} + \frac{TL_i}{L_u} \tag{4}$$

$$t_iR_r = \frac{TC_i}{C_r} + \frac{TL_i}{L_r} \tag{5}$$

So the constraint conditions about deadline are $t_iR_u \le TD_i$ and $t_iR_r \le TD_i$.

The constraint conditions regarding TD_i are $t_iR_u \le TD_i$ and $t_iR_r \le TD_i$. The cost related to scheduling in the hybrid cloud is mainly the cost of public cloud services, which consists of P_r, S_r, and Q_u. The cost F_i of completing T_i in a public cloud is calculated as formula 6.

$$F_i = TC_i \times P_u + TL_i \times Q_u + TL_i \times S_u \tag{6}$$

So the constraint condition about cost is $F_i \le TM_i$.

C. Entropy Optimization Model

In addition to considering task deadlines and cost constraints, the interests of both users and resource providers must be considered, which requires evaluation of the relationship between tasks and resources, and meeting the QoS standards established by users. This study employed an entropy optimization model to maximize the interests of both users and resource providers [42]. Firstly, all users and resource providers are respectively associated with a user satisfaction objective function F_{user} and a service provider benefit objective function F_{prov} [42].

F_{user} is mainly a combination of service price a and service time b, where service time includes the task completion time in addition to the delay time, which is related to the current load status of a given resource. We therefore define F_{user} as follows.

$$a\cdot\sum_{i=1}^{n} t_iR_r + c\cdot\frac{\lambda}{h_r} \le F_{user} \le a\cdot\sum_{i=1}^{n} F_i + b\cdot\sum_{i=1}^{n} t_iR_i + c\cdot\frac{\lambda}{h_u} \tag{7}$$

Here, c denotes the delay time of the weight parameters. In addition, h_r denotes the resources of private cloud services and h_u denotes the resources of public cloud services [42].

We define F_{prov} as follows.

$$F_{prov} = h_u \cdot p(r_u, h_u) - W \cdot v(r_u, h_u) \tag{8}$$

Here, $p(r_u, h_u)$ denotes resource income, W denotes the cost per unit time for the resource paid by a user, and $v(r_u, h_u)$ denotes the resource idle rate [42].

Entropy optimization is based on the principle of a maximum entropy distribution. All known data are considered to achieve the probability distribution of maximum value, which is expressed as the following optimization problem.

$$maxs(x) = maxS_n(P) = -\sum_{i=1}^{n} P_i \ln P_i \tag{9}$$

$$a\cdot\sum_{i=1}^{n} t_iR_r + c\cdot\frac{\lambda}{h_r} \le F_{user} \le a\cdot\sum_{i=1}^{n} F_i + b\cdot\sum_{i=1}^{n} t_iR_i + c\cdot\frac{\lambda}{h_u} \tag{10}$$

$$P_i > 0, i = 1, \cdots, n \tag{11}$$

$$a\cdot\sum_{i=1}^{n} t_iR_r + c\cdot\frac{\lambda}{h_r} \le F_{user} \le a\cdot\sum_{i=1}^{n} F_i + b\cdot\sum_{i=1}^{n} t_iR_i + c\cdot\frac{\lambda}{h_u} \tag{12}$$

$$h_u \cdot p(r_u, h_u) - W \cdot v(r_u, h_u) > 0 \tag{13}$$

Here, P_i represents the probability distribution of the task to achieve the rate. In the model, F_{user} and F_{prov} are employed as the constraint conditions of entropy optimization for simultaneously maximizing the interests of both users and resource providers.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2633288, IEEE Access

IV. The Single-objective Optimization Scheduling strategies

Different users have different priorities for a given task. For example, some users require their task to be completed as soon as possible, while others seek the lowest price. In brief, performance and cost demands differ from user to user and from task to task. All these priorities can be met in the hybrid cloud because of the high-performance computing power of the public cloud resource in conjunction with the very low cost of the private cloud resource. To accommodate the different priorities of users, we propose two scheduling strategies, one of which is a cost-first strategy that gives priority to the cost and the other is a time-first strategy that gives priority to the completion time.

1) Cost-First Scheduling Strategy: The cost-first scheduling strategy prioritizes the scheduling of tasks to resources that can complete the task with the lowest cost under deadline constraints. First, the strategy judges the constraint condition imposed by t_iR_r. According to the deadline constraint, if $t_iR_r \leq TD_i$, T_i will be scheduled to a private cloud; otherwise, it will be scheduled to a public cloud. When scheduled to a public cloud, the constraint condition of cost must be satisfied as well. Then, the cost problem of n tasks is transformed to the following constrained optimization problem.

$$\min \sum F_i = \min \sum TC_i \times P_u + TL_i \times Q_u + TL_i \times S_u \tag{14}$$

S.t.

$$t_iR_u = \frac{TC_i}{C_u} + \frac{TL_i}{L_u} \leq TD_i \tag{15}$$

$$t_iR_r = \frac{TC_i}{C_r} + \frac{TL_i}{L_r} \leq TD_i \tag{16}$$

$$F_i = TC_i \times P_u + TL_i \times Q_u + TL_i \times S_u \leq TM_i \tag{17}$$

The implement process of the scheduling strategy is simply described as Algorithm 1.

Algorithm 1 Cost-First scheduling Algorithm

Input:
T_i, R_u, R_r
Output:
R_x

```
Calculate t_iR_r, min ∑ F_i;
IF t_iR_r ≤ TD_i THEN
    R_x = R_r;
ELSE IF F_i ≤ TM_i THEN
    R_x = R_r;
ELSE
    Divide T_i and Rescheduling;
END IF
END IF
```

Where R_x is the final resource selected for the task T_i.

2) Time-First Scheduling Strategy: Time-first scheduling strategy prioritizes the scheduling of tasks to resources that can complete the task as soon as possible under cost constraints. First, the strategy judges the completion time constraint on all resources, and schedules the task to those resources that can complete the task as soon as possible. According to the cost constraint, when completion times in a public cloud and private cloud are the same or similar, the task will be scheduled to the private cloud; otherwise, it will be scheduled to the public cloud. The degree of similarity is established according to an adjustable threshold λ. Tasks scheduled to a public cloud must satisfy the cost constraint. The optimization and constraint conditions are given as follows.

$$\min \sum t_iR_u, \sum t_iR_r = \min \sum \frac{TC_i}{C_u} + \frac{TL_i}{L_u}, \frac{TC_i}{C_r} + \frac{TL_i}{L_r} \tag{18}$$

S.t.

$$t_iR_u = \frac{TC_i}{C_u} + \frac{TL_i}{L_u} \leq TD_i \tag{19}$$

$$t_iR_r = \frac{TC_i}{C_r} + \frac{TL_i}{L_r} \leq TD_i \tag{20}$$

$$F_i = TC_i \times P_u + TL_i \times Q_u + TL_i \times S_u \leq TM_i \tag{21}$$

The implement process of the scheduling strategy is simply described as Algorithm 2.

Algorithm 2 Cost-First scheduling Algorithm

Input:
T_i, R_u, R_r
Output:
R_x

```
Calculate t_iR_u, t_iR_r, min ∑ t_iR_u, ∑ t_iR_r;
IF (t_iR_u)/(t_iR_r) ≤ λ THEN
    R_x = R_r;
ELSE IF F_i ≤ TM_i THEN
    R_x = R_r;
ELSE
    Divide T_i and Rescheduling;
END IF
END IF
```

Where R_x is the final resource selected for the task T_i. In this paper λ is set as 1.85.

A. The Multi-objective Optimization Scheduling Strategies

In addition to considering deadline tasks and budget constraints, attention should be paid to the relationship between tasks and resources of the state, meeting both user QoS and resource provider interests. Toward that end, this paper uses an entropy optimization model to meet the interests of both users and resource providers.

Firstly, the formula of the entropy optimization model is combined with the resource provider's benefit formula to get the user satisfaction F_{user} and the service provider's benefit objective function F_{prov}.

User satisfaction F_{user} is mainly the combination of service price and service time; service time includes task completion time and delay time; delay time is related to the current load status of resources, so the definition of the entropy optimization model and the task and resources can get user satisfaction F_{user}.

$$a \cdot \sum_{i=1}^{n} t_iR_r + c \cdot \frac{\lambda}{h_r} \leq F_{user} \leq a \cdot \sum_{i=1}^{n} F_i + b \cdot \sum_{i=1}^{n} t_iR_i + c \cdot \frac{\lambda}{h_u} \tag{22}$$

where a denotes the cost of user to pay, b denotes the service time and c denotes the waiting time of the weight parameters. And h_r denotes the resources of private cloud services and h_u denotes the resources of public cloud services.

$$F_prov = h_u \cdot p(r_u, h_u) - W \cdot v(r_u, h_u) \tag{23}$$

Where $p(r_u, h_u)$ denotes resource income, W denotes the cost per unit time resource needed to be paid and $v(r_u, h_u)$ denotes resource idle rate [42].

$$maxs(x) = maxS_n(P) = -\sum_{i=1}^{n} P_i \ln P_i \tag{24}$$

$$s.t. \sum_{i=1}^{n} P_i = 1 P_i > 0, i = 1, \cdots, n \tag{25}$$

A multi-objective scheduling model is proposed to achieve optimal hybrid cloud scheduling efficiency, including optimal task completion time and cost, and to maximize the interests of users and resource providers. The definitions of relevant symbols and parameters are given as follows.

(1) the definition of relevant symbols and parameters:

T_i : represents task, $i = 1, 2, \cdots, n$;

R_j : represents resource, Including m public cloud resources and V private cloud resources, $j = 1, 2, \cdots, m + v$;

f_{ij} :represents the ask resource allocation of cost function $f(x)$;

c_{ij} : represents the task resource allocation of time function $c(x)$;

s_{ij} : represents the task resource allocation that satisfies the interests of users and resource providers;

(2) decision variable:

$$Min \Sigma_{i=1}^{n} \Sigma_{i=1}^{n} T_i j \cdot f_{ij} \tag{26}$$

$$Min \Sigma_{i=1}^{n} \Sigma_{i=1}^{n} T_i j \cdot c_{ij} \tag{27}$$

$$Min \Sigma_{i=1}^{n} \Sigma_{i=1}^{n} T_i j \cdot s_{ij} \tag{28}$$

$$MinF(x) = (f(x), c(x), -s(x)) \tag{29}$$

$$S.t.t_i R_u \leq TD_i \tag{30}$$

$$t_i R_r \leq TD_i \tag{31}$$

$$F_i \leq TM_i \tag{32}$$

$$\sum_{i=1}^{n} P_i = 1 \tag{33}$$

$$P_i > 0, i = 1, \cdots, n \tag{34}$$

$$a \cdot \sum_{i=1}^{n} t_i R_r + c \cdot \frac{\lambda}{h_r} \leq F_{user} \leq a \cdot \sum_{i=1}^{n} F_i + b \cdot \sum_{i=1}^{n} t_i R_i + c \cdot \frac{\lambda}{h_u} \tag{35}$$

$$h_u \cdot p(r_u, h_u) - W \cdot v(r_u, h) > 0 \tag{36}$$

The optimization problem includes all three optimization objectives, i.e., cost-first, time-first, and maximum entropy. Users and resource providers benefit from constraints for this type of combinatorial optimization problem, particularly for obtaining the optimal solution when the optimization process involves a high degree of difficulty. The method employed in this study for solving this combinatorial optimization problem is discussed in the following section.

V. Improved Ant Colony Algorithm to Solve Multi-objective Optimization Scheduling Problems

An ant colony algorithm is employed in this study for solving this combinatorial optimization problem. The ant colony algorithm has definite advantages. The method has been widely used in a large variety of scheduling problems, and has achieved good results. However, because the standard algorithm becomes easily trapped in local optimal solutions, we propose an improved ant colony optimization algorithm to avoid local optimal solutions using a suitable cost function for solution quality assessment and feedback.

A. Algorithm Overview

The multi-objective scheduling problem combined with an ant colony optimization algorithm is represented as a two figure $G = (T, R, E)$, where T is a collection of tasks $T_i (i = 1, 2, \cdots, n)$, R is a collection of public cloud and private cloud resources $R_j (j = 1, 2, \cdots, m + v)$, and E is the collection of paths for a task of T to a node of R, where $E = e_{ij} | i = 1, 2, \cdots, n, j = 1, 2, \cdots, m + v$. An ant from the collection of ants K selects path e_{ij} to express tasks T_i assigned to resources R_j, as illustrated in Fig. 2.

The algorithm implementation process is given as follows.

(1) Ant K is randomly deployed on a task node T_i.

(2) Ant K selects a path e_{ij} to resource node R_j with a specific probability η_{ij}, and determines whether the constraints of the optimization model are satisfied. If so, the resource node is added to the ants list of solutions $Table_k$; otherwise, ant searches for the next resource node.

(3) Ants randomly move to the next task node for allocation of the next task.

(4) The assignment of all tasks is considered as an iterative process, and the algorithm terminates when the maximum number of iterations is reached.

B. Heuristic Information

Heuristic information allows the ant to give priority to paths meeting the three optimization objectives, even if T_i is assigned to the expected degree of resource R_j. This

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2633288, IEEE Access

heuristic information is defined as follows, according to the three optimization objectives.

The first is the cost-first optimization objective, and its heuristic information is η_{ij}^1.

$$\eta_{ij}^1 = \frac{1}{\varepsilon + \Sigma_{i=1}^{n}\Sigma_{j=1}^{m} T_{ij} \cdot f_{ij}} \tag{37}$$

Where ε is a positive number. It is very small.

The second optimization objective is the minimum time, and its heuristic information is η_{ij}^2.

$$\eta_{ij}^2 = \frac{1}{\varepsilon + \Sigma_{i=1}^{n}\Sigma_{j=1}^{m+v} T_{ij} \cdot c_{ij}} \tag{38}$$

The third is the maximum entropy, and its heuristic information is η_{ij}^3.

$$\eta_{ij}^3 = \Sigma_{i=1}^{n}\Sigma_{j=1}^{m+v} T_{ij} \cdot s_{ij} \tag{39}$$

Therefore task T_i is assigned to the expected degree of resource R_j, $\eta_{ij} = \eta_{ij}^1 \cdot \eta_{ij}^2 \cdot \eta_{ij}^3$.

C. Evaluation of Solutions

A path formed by the traversal of all tasks is a feasible solution to the multi-objective optimization problem. A fitness function was employed to evaluate the quality of the feasible solution according to the three optimization objectives to ensure that the best possible solution is achieved.

$$Fit(x) = \gamma \cdot e^{-f(x)} + \delta \cdot e^{-c(x)} + \varpi \cdot e^{s(x)} \tag{40}$$

Among them, γ, δ and ϖ are the weight coefficients of the three optimization objectives, $\gamma > \delta > \varpi$, and γ, δ, $\varpi \in (0,1)$. $c(x)$ and $s(x)$ are cost, time and entropy optimization objective function, respectively, the smaller of $f(x)$, $c(x)$, the bigger of $s(x)$.

D. Pheromone Update

With each ant choosing a path between tasks and resources, we must update the pheromone information to strengthen those paths leading to higher performance, so that more ants can find the path. Simultaneously, updating also increases the number of path choices to avoid stagnation of the algorithm. The update rules are given as follows.

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij} \tag{41}$$

Here, $\tau_{ij}(t)$ represents the pheromone level on e_{ij} at time t, $\tau_{ij}(t+1)$ is the updated pheromone level, ρ is a pheromone volatility factor, which ensures that the updated information is maintained between updating processes, and $\Delta\tau_{ij}$ is the pheromone level increment on e_{ij} in the current iteration process, where its value is related to the optimization objective. Among these, X is a feasible solution to the optimization objective, which only updates the current optimal solution and suboptimal solution for improving the efficiency of the algorithm.

$$\Delta\tau_{ij} = \begin{cases} \frac{1}{f(X)}, & \text{if } X \in \text{The best optimal or sub optimal solution.} \\ 0, & \text{otherwise.} \end{cases} \tag{42}$$

Among them, X is a feasible solution to the optimization objective, and it only updates the current optimal solution and sub optimal solution, which can improve the efficiency of the algorithm.

E. Transition Probability

Ant k selects a new path adjacent to the current path based on heuristic information τ_{ij}. The transfer probability that ant k selects a path of pheromone level τ_{ij} is given as follows.

Algorithm 3 MOSACO

Input:
$T_i, R_j, f_x, c(x), s(x)$
Output:
$E = e_{ij} | i = 1, 2, \cdots, n, j = 1, 2, \cdots, m+v$
Begin
Initialize n, m, v, K, N_{max};
For k= 1 to K
Initialize η_{ij} and τ_{ij};
Set ant k to T_i;
For r = 1 to N_{max}
For i = 1 to n
For j = 1 to $m+v$
If T_{ij} and satisfy $f(x)$ $c(x)$ $s(x)$ and their constraints, then
$Table_k \leftarrow e_{ij}$;
END IF
Next j
Next i
Record the best path e_{ij};
Next r
Output the set of all best paths E;
END

F. Algorithm Complexity Analysis

The time complexity resides mainly in the use of the ant colony algorithm for optimizing the MOSACO approach. Cost-first and time-first scheduling strategies are simple, and require only conditional judgments that do not involve the use of integral differential operations. As such, the complexity of optimization for these strategies is linear, namely, $O(n)$. However, the complexity of optimization for the MOSACO algorithm is greater because it involves three optimization objectives and multiple constraints. For K ants, if the maximum number of iterations is N_{max}, the time complexity of the algorithm is $O(N_{max} \cdot K \cdot n \cdot (m+v))$.

VI. Experiments

The experiments comprise two parts. The first part seeks to verify the proposed method via simulation data, and the second part seeks to verify the method via application examples obtained from the relevant literature, which include three IaaS application examples [27],[28],[29].

A. Experiments Setup

The simulation system is designed to verify the performance of the proposed scheduling method. The cloud simulation software Cloudsim3.0 [44] created the A, B, C, three data

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2633288, IEEE Access

TABLE II
PARAMETERS SETUP OF VM IN THREE DATA CENTERS.

Parameter	VM setup in A and C	VM setup in B
The number of CPU	1	2
CPU computing capacity	200 MIPs	400 MIPs
RAM	1G	2G
BW	2M/s	4M/s
Storage	2G	4G

TABLE III
PARAMETERS SETUP OF TASKS.

Parameter	setup of tasks
Length	[400,1000]MIPs
File Size	[200,400]MB
Output Size	[20,40]MB

centers that used the information of VM and ECs in three IaaS application examples of Ref. [27] [28] [29].

The cloud simulation software Cloudsim3.0 [44] was employed to create three data centers, denoted as A, B, and C. Data center A is a private cloud, whereas data centers B and C are public clouds. To fully simulate the difference between private and public cloud services in a hybrid cloud environment, the parameter settings of the three data centers included some differences, where A and C employed 20 virtual machines, B employed 200 virtual machines, and the resource configuration of B differs from those of A and C. The parameters are listed in Table II.

The Cloudlet function in Cloudsim was employed to create 100C600 random cloud tasks, according to the number of virtual machines in the three data centers. The specific task parameters are listed in Table III. The related experimental parameters for the MOSACO algorithm are listed in Table IV.

The cloud tasks are randomly created by Cloudlet of Cloudsim. According to the number of virtual machines in three data centers, the number of tasks is 100 -600. The parameter setup is shown in Table V.

B. Evaluation Metrics

In the simulations, the performances of the proposed cost-first, time-first, and MOSACO strategies are compared with those of the Min-Min [13][14] and first in, first out (FIFO) [15] algorithms. The Min-Min algorithm is a classic heuristic algorithm that focuses on optimizing the completion time. First, the algorithm selects the minimum completion time of each task. Second, it selects the combination of resources and

TABLE IV
PARAMETERS SETUP OF MOSACO.

Parameter	K	N_{max}	α	β	ρ	γ	δ	ϖ
Value	50	50	0.6	0.3	0.3	0.4	0.35	0.2

TABLE V
PRICE OF TWO PUBLIC CLOUD.

The computing price	0.085$ hour	0.12$hour
The storage price	0.06$GB	0.1$GB
The transmission price	0.05$GB	0.1$GB

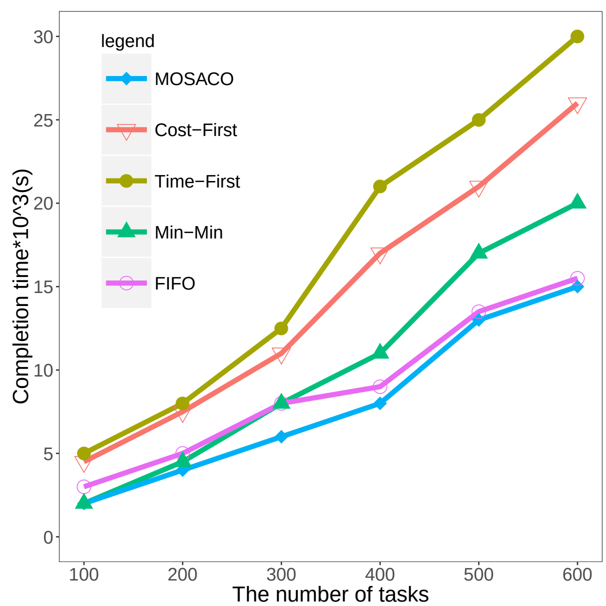

Fig. 2. The completion time.

tasks whose completion time is the minimum of all minimums. FIFO is a standard scheduling algorithm that advocates fair scheduling, where the first incoming task is scheduled first. This method employs no optimization and does not consider the cost.

Four metrics are used to evaluate the performance of these scheduling strategies: the completion time, cost, the deadline violation, and the utilization of private cloud resources.

The cost is calculated by the above formulas. The prices of public cloud resources refer to Amazon's cloud services: the computing price is $0.12 per hour, the store price is 0.1 $/ GB, and the I/O price is $ 0.1/MIPs. The computing cost is calculated by the amount of computation according to the computing time in the experiment.

The deadline violation D_v is calculated by formula 12.

$$D_v = \frac{n_v}{n} \times 100\% \tag{43}$$

Where n_v is the number of tasks violated the deadline, n is the total of tasks.

C. Result of Simulation Experiments

1) The Completion Time: The results represent the average over 10 experiments. The task completion times of the five methods with different numbers of tasks are shown in Fig. 3.

The performance of the MOSACO algorithm was the best, the performances of time-first and Min-Min algorithms were second, and the performance of the FIFO algorithm was the worst. MOSACO performed best because of the number of comprehensive optimization objectives and the use of the ant colony optimization. While Min-Min has obvious advantages in job execution time and average response time, time-first was slightly better as the number of tasks increased because time-first can balance the use of resources better for a large number of tasks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2633288, IEEE Access

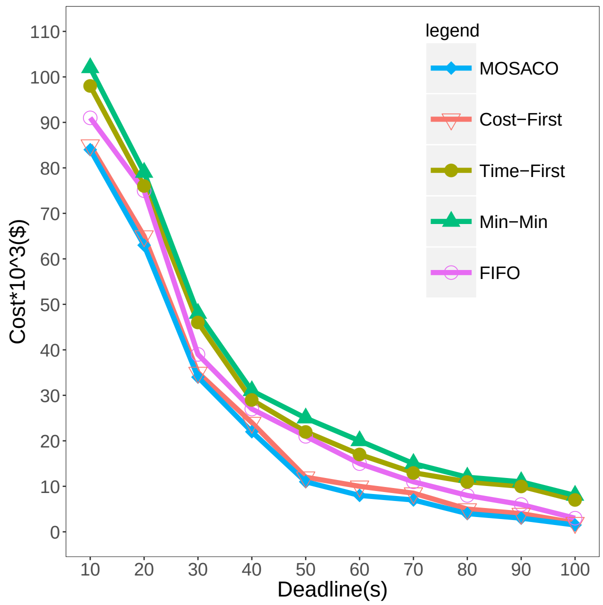

Fig. 3. The cost of 200 tasks.

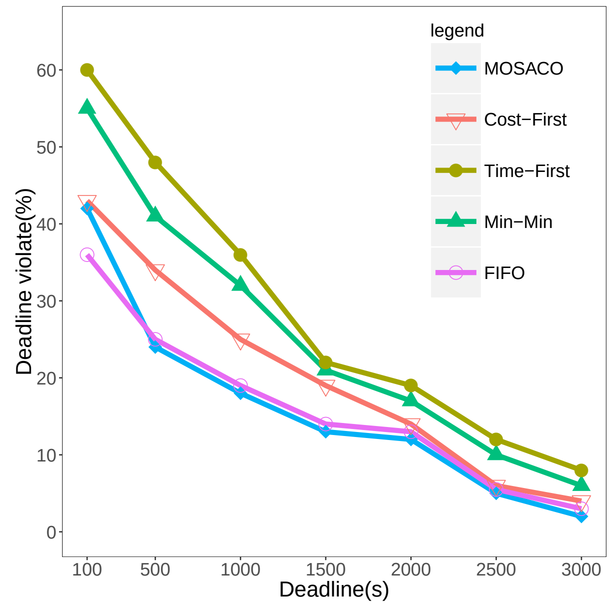

Fig. 5. The utilization of private cloud resources.

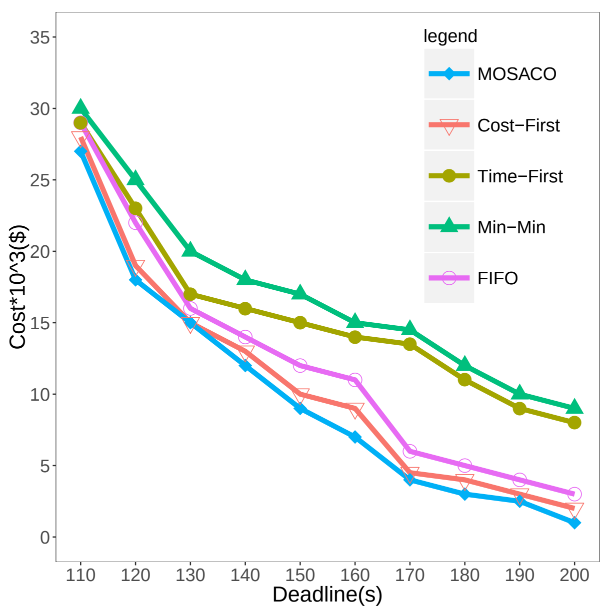

Fig. 4. The cost of 500 tasks.

2) The Cost: To verify the costs of the methods in a hybrid cloud (except for FIFO), 200 and 500 tasks with different deadlines were selected, and the results are shown in Figs. 4 and 5, respectively.

It is clear that the cost of MOSACO is the least because it takes the cost and other objectives into account. Cost-first was the second, and time-first and Min-Min had the highest costs because both pursue a minimum completion time, which gives priority to relatively high-performance and expensive public cloud resources. Cost-first provided lower costs because it gives priority to private cloud resources and considers cost constraints when selecting public cloud resources.

3) Number of Deadline Violations: We selected 500 tasks, and changed the deadline time from 100 s to 3000 s. The deadline violations of the five scheduling methods are shown in Fig. 6.

MOSACO had the best performance, followed by time-first and cost-first in that order, while the performance of FIFO was the worst. The results reflect the fact that MOSACO, time-first, and cost-first consider deadline constraints. The pursuit of the optimal completion time in Min-Min tends to lead to breaches in the deadlines of major tasks, particularly when the number of tasks is very large.

4) Degree of Private Cloud Resource Utilization: We selected 10 tasks to evaluate the degree of utilization of private cloud resources obtained for the five scheduling algorithms. Each deadline was once again randomly generated. The experimental results are shown in Fig. 7.

Similar to the case of cost, MOSACO had the best private cloud resource utilization because it considered the interests of public cloud providers. Its total scheduling times of 10 resources was close to 100. The performance of cost-first was second because it gives priority to the cost, and prefers to select private cloud resources. Min-Min was the worst because it gives priority to public cloud resources.

D. Result of Application Examples

As discussed, three IaaS application examples [27],[28],[29] were employed for experiments to further demonstrate the performance of the proposed method. It is noted that researchers in [27] also employed optimization scheduling methods in a hybrid cloud, with research objectives and methods similar to those employed in the present work. Accordingly, MOSACO, cost-first, and time-first were compared with the previously proposed SLPSO-SA and SPSO-SA methods [27]. The application examples and their cost setups are listed in Table VI.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2633288, IEEE Access

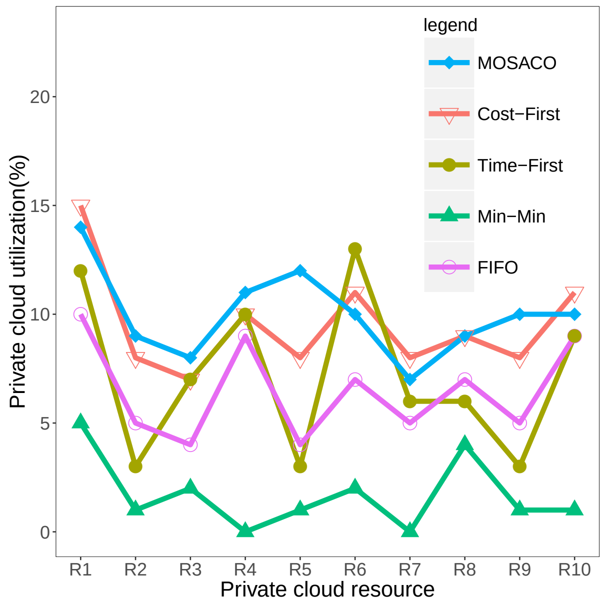

Fig. 6. The utilization of private cloud resources.

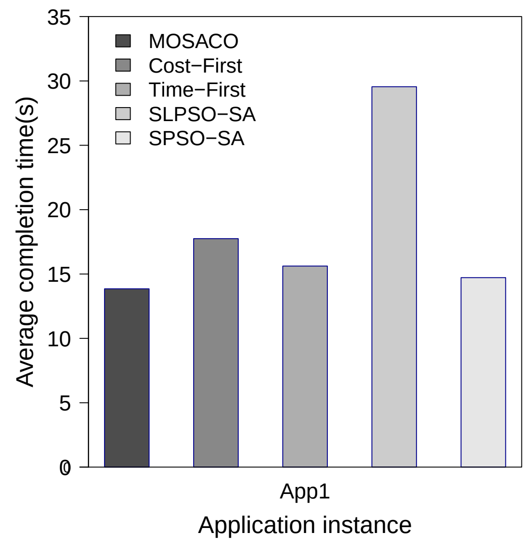

Fig. 7. The completion time.

TABLE VI
PARAMETERS SETUP AND PRICE OF APPLICATION EXAMPLES.

Name	CPU	Memory(GB)	ESs-A($/hour)	ESs-B	ESs-C
Small	1	1.7	0.085	0.070	0.100
Large	4	7.5	0.34	0.30	0.40
Xlarge	8	15	0.68	0.70	0.72

Example 1 includes eight small applications, and Examples 2 and 3 include five and eight large applications, respectively. The three examples and their parameters as listed in Tables VII and VIII. Each virtual machine type of application request must choose from the three types in the above list. Because of limitations in the search space of MOSACO, SLPSO-SA, and SPSO-SA, deadlines and run times of every application must be an integer number obtained from a uniform distribution.

The same four evaluation metrics were employed, including completion time, cost, the number of deadline violations, and the degree of private cloud resource utilization. The researchers in [27] focused on the interests of IaaS providers; otherwise, the work was developed from the perspective of user costs, so cost was employed in the present study as an evaluation index rather than the evaluation metric employed in the original study (i.e., provider interest). The average utilization of CPUs and memory in the private cloud were employed to evaluate the degree of private cloud resource utilization, which is defined as follows.

$$U_{CPU} = \frac{\sum_{i=1}^{n} T_{iCPU}}{total_{CPU}} \tag{44}$$

$$U_{Memory} = \frac{\sum_{i=1}^{n} T_{iMemory}}{total_{Memory}} \tag{45}$$

Here, n is the number of task, $total_{CPU}$ and $total_{Memory}$ are the total number of CPUs and memory employed, respectively.

1) Completion Time: The average completion time of three application examples can be obtained in the experiment, and the results for the average completion times of Example 1 and Examples 2 and 3 are shown in Figs. 8 and 9, respectively. Owing to the large differences between the scales of Example 1 and Examples 2 and 3, the results for each evaluation metric are reported below using separate figures (Figs. 8 and 9).

As can be seen from Figs. 8 and 9, for Example 1, MOSACO had the best performance, followed by SPSO-SA, time-first, and cost-first. The performance of SLPSO-SA was the worst. In Examples 2 and 3, MOSACO and time-first were obviously better than the other three methods. Completion time is the primary optimization goal of time-first, while MOSACO also takes completion time as one of its main objectives; however, the primary optimization goal of SLPSO-SA is provider interests. Therefore, the relative performances of the various methods are understandable, and illustrate the effectiveness of the time-first approach.

2) Cost: To unify the methods employed in [27] with the methods employed in the present study, the costs for completing the corresponding tasks are used in the five methods.

TABLE VII
TASKS PARAMETERS SETUP OF THREE APPLICATIONS.

Parameters	App1	App2	App3
Number of tasks	unif[1,5]	unif[1,50]	unif[1,50]
VM type	unif[1,3]	unif[1,3]	unif[1,3]
Deadline(hours)	unif[1,5]	unif[1,168]	unif[1,168]
Runtime(hours)	unif[1,Deadline]	unif[1,Deadline]	unif[1,Deadline]

TABLE VIII
RESOURCES PARAMETERS SETUP OF THREE APPLICATIONS.

Cloud resources	App1	App2	App3
CPU	20	512	512
Memory	40GB	1024GB	1024GB

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2633288, IEEE Access

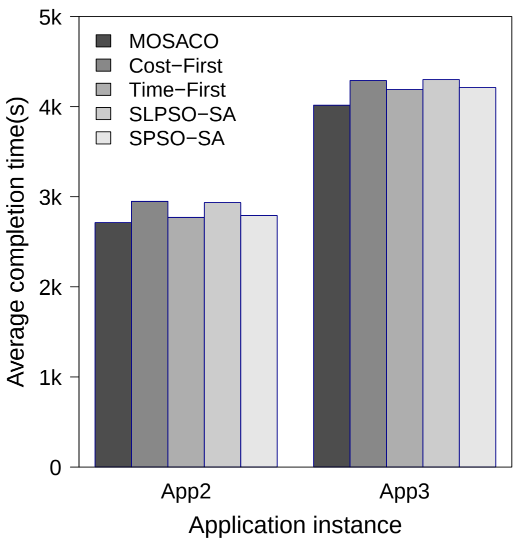

Fig. 8. The completion time.

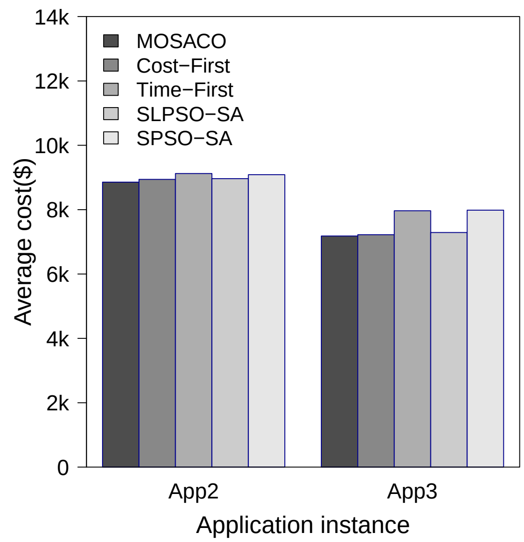

Fig. 10. The cost of 500 tasks.

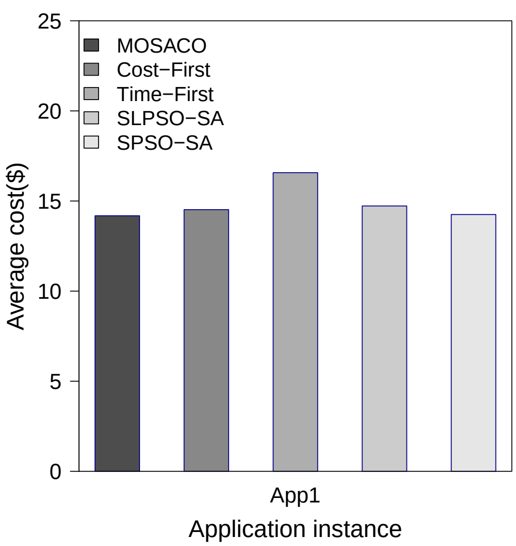

Fig. 9. The cost of 200 tasks.

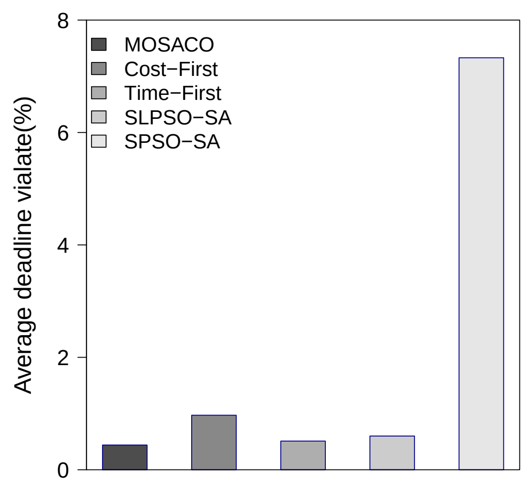

Fig. 11. The utilization of private cloud resources.

Thus, Amazons cost and standards are used in the experiment to obtain the average costs for completing Examples 1, 2, and 3. The results of the costs of the four methods are shown in Figs. 10 and 11 for Example 1 and Examples 2 and 3, respectively.

As seen from Figs. 10 and 11, for Example 1, MOSACO had significantly better performance than the other methods. The performance of cost-first was worse than that of SPSO-SA, but slightly better than those of SLPSO-SA and time-first. However, in Examples 2 and 3, which include a large amount of data, MOSACO was the best, followed by cost-first, which was nearly equivalent to SLPSO-SA. These three methods have obvious advantages in terms of cost because the main optimization goals are all cost/benefit.

3) Number of Deadline Violations: The number of deadline violations for SLPSO-SA and SPSO-SA were obtained from the previously reported results [27], whereas those for MOSACO, cost-first, and time-first were obtained using Eq. (12). The average number of deadline violations for the three application examples are shown in Fig. 12.

MOSACO had the best performance, followed by time-first and SLPSO-SA, while the performance of SPSO-SA was the worst. Both the proposed method and SLPSO-SA consider deadline constraints, such that their deadline violation rates were relatively low, and within an acceptable range.

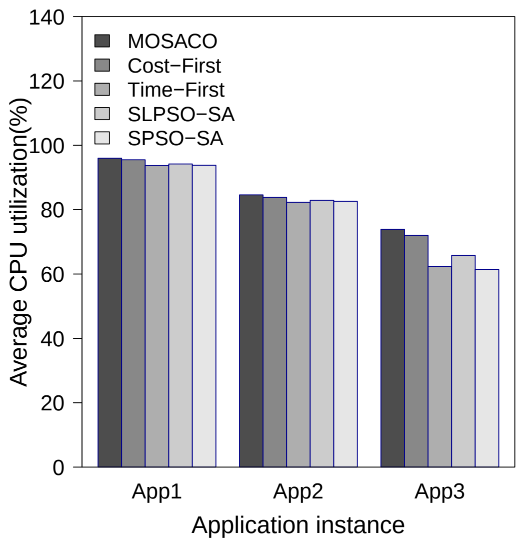

Fig. 12. The utilization of private cloud resources.

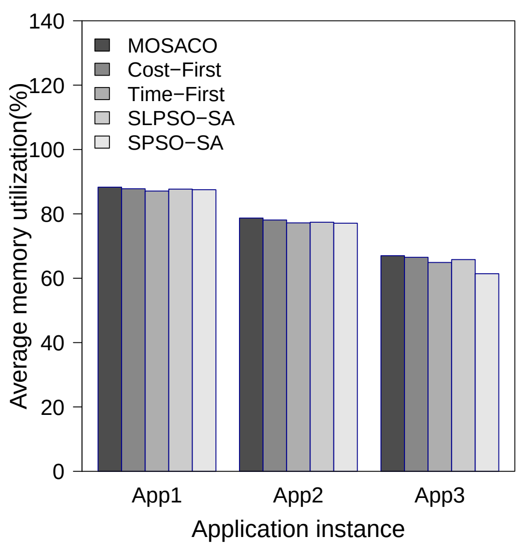

Fig. 13. The utilization of private cloud resources.

4) Degree of Private Cloud Resource Utilization: This evaluation metric was calculated for SLPSO-SA and SPSO-SA using the previously reported method for the average utilization of CPU and memory in the private cloud [27], while MOSACO, cost-first, and time-first were calculated according to Eqs. (13) and (14). The results are shown in Figs. 13 and 14 for Example 1 and Examples 2 and 3, respectively.

The CPU utilization of MOSACO was the highest of the five methods, followed by cost-first and SLPSO-SA; however, time-first was the worst. This ordering of the results reflects the fact that MOSACO employs a variety of optimization objectives, while the focus of cost-first places the priority on private cloud use, and the focus of time-first places the priority on public cloud use. In terms of memory utilization, all five methods provided nearly equivalent utilization. MOSACO and cost-first exhibited a slight advantage only for Examples 2 and 3.

VII. Conclusion

In this paper, we addressed resource utilization in hybrid cloud computing with multiple considerations of cost, task completion time, user QoS, and resource provider interests, and developed the MOSACO algorithm to optimize resource utilization according to deadline and cost constraints. Firstly, we proposed the cost-first and time-first single-objective hybrid cloud scheduling optimization strategies, which respectively prioritize cost with deadline constraints and task completion time with cost constraints. To maximize user QoS and resource provider interests, we employed an entropy optimization model based on user and resource provider objective functions. The MOSACO algorithm was then developed by combining the two single-objective scheduling optimization approaches with entropy optimization to establish a multi-objective scheduling model optimized using an improved ant colony algorithm. Compared with other similar resource scheduling methods, MOSACO demonstrated the highest optimality according to considerations of task completion time, cost, number of deadline violations, and the degree of private cloud resource utilization based on simulation and three application examples. In addition, the cost-first approach demonstrated a definite advantage in terms of cost, while the time-first approach demonstrated considerable advantage in terms of task completion time, validating the effectiveness of the scheduling method proposed in this paper.

Acknowledgement

This work was supported by the Natural Science Foundation of Guangdong Province, China (Project No. 2014A030313729), 2013 top Level Talents Project in “Sailing Plan” of Guangdong Province, 2014 Guangdong Province Outstanding Young Professor Project, Science and Technology Key Project of Guangdong (2014B010112006), Natural Science Fund of Guangdong (2015A030308017). Lei Shu and Shoubin Dong are the corresponding authors.

References

[1] N. Chopra , S. Singh . “Survey on scheduling in hybrid clouds,” in *Proc. 2014 International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, July 2014, pp. 11-13.

[2] H. Zhang , G. Jiang , K. Yoshihira , H. Chen . “Proactive Workload Management in Hybrid Cloud Computing, ” in *Proc.IEEE Transactions on Network and Service Management*, vol. 11, no. 1, pp. 90-100, 2014.

[3] R. Duan , R. Goh , Q. Zheng , Y. Liu . “Scientific Workflow Partitioning and Data Flow Optimization in Hybrid Clouds,” in *Proc.IEEE Transactions on Cloud Computing*, no.99, pp. 1-12, 2014.

[4] R. Van den Bossche, K. Vanmechelen, J. Broeckhove. “Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads,” in *Proc. 2010 IEEE 3rd International Conference on Cloud Computing*, Miami, FL, 14-18 April 2010, pp. 228-235.

[5] C. Zhao, S. Zhang, Q. Liu, J. Xie, J. Hu. “Independent tasks scheduling based on genetic algorithm in cloud computing,” in *Ptoc. 2009 International Conference on Wireless Communication, Network, Mobile Computer*, Marrakech, Morocco, 2009, pp. 1-4.

[6] R. Van den Bossche, K. Vanmechelen, J. Broeckhove. "Cost-efficient scheduling heuristics for deadline constrained workloads on hybrid clouds," in*Proc. 2011 IEEE Third International Conference on Cloud Computing Technology and Science*, Baltimore, MD, 7-10 Nov., 2011, pp.320-327.

[7] X. Qiu, W. L. Yeow, C. Wu, F. C. Lau. "Cost-minimizing preemptive scheduling of mapreduce workloads on hybrid clouds," in *Proc. 2013 IEEE/ACM 21st International Symposium on Quality of Service*, Miami, FL, 14-18 April 2013, pp. 1-6.

[8] N. Chopra, S. Singh. "Deadline and cost based workflow scheduling in hybrid cloud," . in *Proc. 2013 International Conference on Advances in Computing, Communications and Informatics*, Miami, FL, 14-18 April 2013, pp. 840-846.

[9] M. R. Hoseinyfarahabady, H. R. D. Samani, L. M. Leslie. "Handling Uncertainty: Pareto-Efficient BoT Scheduling on Hybrid Clouds," in *Proc. 2013 42nd International Conference on Parallel Processing*, Reno, NV, USA , 10-16 Nov. 2013, pp. 419-428.

[10] T. A. L. Genez, L. F. Bittencourt, E. R. M. Madeira. Christopher, R. Hejer. "On the Performance-Cost Tradeoff for Workflow Scheduling in Hybrid Clouds," in *Proc. 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*, Reno, NV, USA , 10-16 Nov. 2013, pp. 411-416.

[11] L. Li. "An optimistic differentiated service job scheduling system for cloud computing service users and providers," in *Proc. Multimedia and Ubiquitous Engineering, 2009. MUE'09. Third International Conference on. IEEE*, 2009, pp. 295-299.

[12] B. Sotomayor , R. S. Montero , I. M. Llorente , F. Ian. "Virtual infrastructure management in private and hybrid clouds," in *Proc. Internet computing*, IEEE, vol. 13, no.5, pp. 14-22, 2009.

[13] S. Chaisiri , B. S. Lee , D. Niyato. "Optimal virtual machine placement across multiple cloud providers," in *Ptoc. Services Computing Conference, 2009. APSCC 2009*, IEEE Asia-Pacific. IEEE, 2009, pp. 103-110.

[14] Y. Wang , W. Shi . "Budget-Driven Scheduling Algorithms for Batches of MapReduce Jobs in Heterogeneous Clouds," in *Proc.IEEE TRANSACTIONS ON CLOUD COMPUTING*, vol.2, no.3, pp. 306-319, 2014.

[15] K. Li, X. Tang. "Energy-Efficient Stochastic Task Scheduling on Heterogeneous Computing Systems," in *Proc. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, vol. 25, no. 11, pp. 2967-2876, 2014.

[16] Z. Xiao, W. Song, Q. Chen. "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment," in *Proc. IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1107-1116, 2013.

[17] R.A.M. Razali, R.A. Rahman, N. Zaini, M. Samad. "Virtual machine migration implementation in load balancing for Cloud computing," in *Proc. 2014 5th International Conference on Intelligent and Advanced Systems (ICIAS)*, Kuala Lumpur, 3-5 June, 2014, pp. 1-4.

[18] S. Di, C. Wang, F. Cappello. "Adaptive Algorithm for Minimizing Cloud Task Length with Prediction Errors," in *Proc. IEEE TRANSACTIONS ON CLOUD COMPUTING*, vol.2, no.2, pp. 194-206, 2014.

[19] R. Duan, R. Prodan , X. Li. "Multi-Objective Game Theoretic Scheduling of Bag-of-Tasks Workflows on Hybrid Clouds," in *Proc. IEEE TRANSACTIONS ON CLOUD COMPUTING*, vol.2, no.1, pp. 29-42.2014.

[20] Z. Liu, W. Qu, W. Liu, Z. Li, Y. Xu. "Resource preprocessing and optimal task scheduling in cloud computing environments," in *Proc. Concurrency and Computation: Practice and Experience*, 2014, pp.1-22.

[21] Q. Zhu, G. Agrawal. "Resource provisioning with budget constraints for adaptive applications in cloud environments," in *Proc.IEEE Transactions on Services Computing*, vol. 5, no. 4, pp: 497-511, 2012.

[22] S. Abrishami, M. Naghibzadeh, D. H. Epema. "Cost-driven scheduling of grid workflows using partial critical paths," in *Proc. IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1400C1414, Aug. 2012.

[23] M. Shifrin, R. Atar, I. Cidon. "Optimal scheduling in the hybrid-cloud," in *Proc. 2013 IFIP/IEEE International Symposium on Integrated Network Management*, Miami, FL, 14-18 April 2013, pp. 51-59.

[24] R. Duan, R. Prodan , X. Li. "Multi-Objective Game Theoretic Scheduling of Bag-of-Tasks Workflows on Hybrid Clouds," in *Proc.IEEE TRANSACTIONS ON CLOUD COMPUTING*, vol.2, no.1, pp. 29-42, 2014.

[25] M. Rahman, X. Li, H. Palit. "Hybrid heuristic for scheduling data analytics workflow applications in hybrid cloud environment," in *Proc. 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*, Reno, NV, USA , 10-16 Nov. 2011, pp. 966-974.

[26] B. Javadi, J. Abawajy, R. Buyya. "Failure-aware resource provisioning for hybrid Cloud infrastructure ," *Journal of parallel and distributed computing*, vol. 72, no. 10, pp. 1318-1331, Oct. 2012.

[27] X. Zuo, G. Zhang, W. Tan. "Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 564-573, Mar. 2014.

[28] R. Van den Bossche, K. Vanmechelen, J. Broeckhove. "Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads," in *Proc. 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD)*, IEEE, 2010, pp. 228-235.

[29] S. He, L. Guo, Y. Guo. "Real time elastic cloud management for limited resources," in *Proc. 2011 IEEE International Conference on Cloud Computing (CLOUD)*, IEEE, 2011, pp. 622-629.

[30] R. Chaukwale, S. S. Kamath. "A modified ant colony optimization algorithm with load balancing for job shop scheduling," in *Proc. 2013 15th International Conference on Advanced Computing Technologies (ICACT)*, IEEE, 2013, pp. 1-5.

[31] Y. Wei, Y. CHEN. "Cloud Computing Task Scheduling Model Based on Improved Ant Colony Algorithm," in *Proc. Computer Engineering*, vol.41, no.2, pp. 12-16, 2015.

[32] Y. Zha, Y. Li. "Task scheduling in cloud computing based on improved ant colony optimization," in *Proc. Computer Engineering and Design*, vol.34, no. 5, pp. 1716-1719, 2013.

[33] R. Van den Bossche, K. Vanmechelen, J.Broeckhove. "Cost-efficient scheduling heuristics for deadline constrained workloads on hybrid clouds," in *Proc. 2011 IEEE Third International Conference on Cloud Computing Technology and Science*, Baltimore, MD, 7-10 Nov. 2011, pp. 320-327.

[34] W. DAUN, X. FU, F. WANG, B. WANG, H. HU. "QoS constraints task scheduling based on genetic algorithm and ant colony algorithm under cloud computing environment," in *Proc. Journal of Computer Applications*, vol.34, no. S2, pp.66-69, 2014.

[35] G. Han , W. Que , G. Jia , L. Shu . "An Efficient Virtual Machine Consolidation Scheme for Multimedia Cloud Computing," in *Proc.Sensors*, Vol.16, No.2, pp. 246-258, 2016.

[36] G. Han , L. Wan , L. Shu, N. Feng . "Two Novel DoA Estimation Approaches for Real Time Assistant Calibration System in Future Vehicle Industrial," in *Proc.IEEE Systems Journal*, 2015. DOI: 10.1109/JSYST.2015.2434822.

[37] G. Han , J. Shen , L. Liu , L. Shu . "BRTCO: A Novel Border Line Recognition and Tracking Algorithm for Continuous Objects in Wireless Sensor Networks," in *Proc.IEEE Systems Journal*, 2016, DOI: 10.1109/JSYST.2016.2593949

[38] L. Gu , D. Zeng , S. Guo , A. Barnawi , I. Stojmenovic . "Optimal Task Placement with QoS Constraints in Geo-distributed Data Centers using DVFS," in *Proc.IEEE Transactions on Computers*, Vol. 64, No. 7, pp. 2049 - 2059, 2015.

[39] Y. Zhang, X. Sun, B. Wang. "Efficient Algorithm for K-Barrier Coverage Based on Integer Linear Programming," in *Proc.China Communications*, vol. 13, no. 7, pp. 16-23, 2016.

[40] Z. Fu, X. Sun, Q. Liu, L. Zhou, J. Shu. "Achieving Efficient Cloud Search Services: Multi-keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing," in *Proc.IEICE Transactions on Communications*, vol. E98-B, no. 1, pp.190-200, 2015.

[41] B. Gu, VS. Sheng, KY. Tay, W. Romano, S. Li. "Incremental Support Vector Learning for Ordinal Regression," in *Proc.IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1403-1416, 2015.

[42] L.Y. ZUO, Z.B. CAO, S.B. DONG. "Virtual Resource Evaluation Model Based on Entropy Optimized and Dynamic Weighted in Cloud Computing," in *Proc. Journal of Software*, vol.24, no.8, pp.1937-1946, 2013.

[43] L.Y. ZUO, S.B. DONG, L. Shu. "Resource scheduling methods based on deadline and cost constraint in hybrid cloud," in *Proc. Application esearch of Computers*, vol.33, no.8, pp. 1-7, 2016.

[44] R. N. Calheiros, R. Ranjan, A. Beloglazov, A. F. Cesar, R. De, R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Software: Practice and Experience, Wiley*, vol. 41, no. 1, pp. 23-50, 2011.