Kybernetes

Resource scheduling in a private cloud environment: an efficiency priority perspective

Junfei Chu Jie Wu Qingyuan Zhu Jiasen Sun

Resource scheduling in a private cloud environment: An efficiency priority perspective

1.Introduction

Cloud computing (Rosenthal et al., 2010; Rodero-Merino et al., 2013; Taylor et al., 2010; Brumec and Vrček, 2013) is emerging as a new kind of network application mode which aims to provide reliable, customized, and quality of service (QoS)-guaranteed computational environments for cloud users (Ke et al., 2013). Recent studies have shown that a large proportion of companies worldwide use a cloud computing service to achieve new business goals and to provide more efficient services for their customers (Van Do and Rotter, 2012). However, many cloud users are worrying about issues such as data safety, reliability, and nonsupport of legacy tools in the cloud computing environment. As an important branch of cloud computing, the private cloud can successfully avoid the potential safety hazard of transferring data and business to a third-party data center, which makes it favorable for enterprises. Private cloud computing platforms generally run inside of the enterprises, and processing large data tasks is one of their main features. After the private cloud users submit a task to the private cloud computing platform, the resource scheduling center of the platform needs to dispatch the required resources using the resource nodes to complete this task. The resource nodes are virtual machines (VMs) constructed on physical computing resources (computational servers) using virtualization technology (Bourguiba et al., 2012). To achieve the goals of completing the corresponding task with the high quality and ensuring the physical resource nodes on the private cloud platform are utilized efficiently, an efficient resource scheduling approach is needed. Therefore, the resource scheduling problem in private cloud environments has become a hot research topic.

In recent studies, many scholars have used heuristic methods for resource scheduling in private cloud environments. For example, the ant colony algorithm was incorporated into the resource scheduling approach in the cloud computing environment by Yuan et al. (2012), Huang et al. (2013), and Zhu et al. (2012). These studies attempted to develop an optimal resource scheduling scheme utilizing the positive feedback and distributed-cooperation operating mechanisms of this intelligent search algorithm. An improved differential evolution algorithm (IDEA) was proposed by Tsai et al. (2013) to optimize resource scheduling and task allocation in a cloud computing environment. This algorithm combined the Taguchi method (Taguchi et al., 2000) and the differential evolution algorithm (Ho et al., 2010). The objective function in it was built mainly on the basis of the consideration of time and money required to complete the tasks. The resource scheduling approach in Gu et al. (2012) incorporated an genetic algorithm (GA) which can automatically obtain and accumulate information of the resource nodes before finding the optimal resource scheduling scheme. Although GA has good searching and optimizing capabilities, the phenomenon of premature partial optimal solution easily appears (Elbeltagi et al., 2005), which will affect the optimality of the resource scheduling schemes. Laili et al. (2013) presented the idea of combining service composition optimal selection and optimal allocation of the computing resources. They proposed a new ranking chaos algorithm to address the large-scale dual scheduling of a cloud service. Also, they incorporated dynamic heuristics and ranking selection to control chaos evolution. The particle swarm optimization algorithm was introduced into the resource scheduling approach by Gorbenko and Popov (2012). Information of the resource nodes is exchanged between the particles during their process of obtaining the optimal solution. This algorithm also introduced a mechanism involving chaos to prevent the premature emergence phenomenon. Krishna (2013) presented a honey bee behavior inspired load balancing (HBB-LB) algorithm to do resource scheduling in cloud computing environments. The algorithm was used to achieve a well-balanced load across virtual machines for maximizing the throughput. In addition, it was also claimed to have the ability of improving the efficiency of resource utilization and maximizing the computing resource production capacity.

In the consideration of the fitness of the objective functions of the resource scheduling approaches, most of the studies mentioned above focused only on the money required for operating the resource nodes and the time needed to complete the corresponding tasks, which gives no consideration to the various calculative indexes in each computational resource node. Therefore, the main shortcomings of the present resource scheduling approaches in the cloud computing environment lie in two aspects. Firstly, the objective of the traditional resource scheduling approaches considers only money and time required when completing the tasks. More indexes of the resource nodes, such as the CPU performance, hard drive capacity, and so on, should be taken into consideration. Secondly, money and time required for completing the tasks are not available in the previous methods; they can only be estimated from the historical data in the database. The estimated time and money will make the resource scheduling results inaccurate.

To avoid these shortcomings, we propose a resource scheduling approach from an efficiency priority perspective, which aims to accomplish the tasks with resource scheduling schemes that have the highest total computational efficiency scores. Our main work and contributions are as follows. Firstly, we incorporate DEA into the resource scheduling problem in the private cloud environment and propose a suitable DEA model to evaluate the efficiencies of resource node decision making units (RN-DMUs). Secondly, based on the efficiency scores we get for the RN-DMUs, we introduce a 0-1 programming technique and build a simple resource scheduling model which can be used to obtain optimal resource scheduling schemes, i.e. schemes that have the highest total computational efficiencies for the calculation tasks. In addition, based on the proposed models, the workflow of resource scheduling using our approach is presented to ensure its practical application in the private cloud environment. Finally, the proposed approach is applied in an experiment of resource scheduling in a private cloud environment.

The rest of this paper is organized as follows. Section 2 provides a formal description of the resource scheduling problem in the private cloud environment.

Section 3 gives a suitable DEA model for measuring computational efficiencies of the RN-DMUs. Section 4 discusses the DEA-based resource scheduling model and the workflow of the resource scheduling in the private cloud environment. An experiment is done and its results are discussed in Section 5. Finally, conclusions and further research directions are given in Section 6.

2. Formal description of the resource scheduling problem

In the private cloud environment, the calculation task submitted by a user is a task class. A task class is defined as a set of tasks that have the same task type and can be executed concurrently. We assume that each task class has t tasks, every task should be executed on only one resource node, and each resource node can run only one task at a time. We also assume that the resource quantity demanded by each task and the resource quantity that can be provided by each resource node are known in advance. The resource scheduling problem is how to select suitable resource nodes to fulfill the resource requirements of the tasks. Here we denote the resource scheduling problem as the following nine tuple,

$$DRSM = \langle R, P, Y, \theta, A, B, F, Z, S\rangle \quad (1)$$

We let R be the resource requirement set of the tasks, which represents the resource demands of tasks. R is defined as $R=\{R_l \mid l=1,2,...,h\}$. R_l denotes the resource requirement set of task *l*, and it can be defined as $R_l=\{r_{il} \mid i=1,2,...,m; l=1,2,...,h\}$, where r_{il} denotes the quantity of the i^{th} computing resource required by the l^{th} task.

We let *P* be the set of resource nodes, which can be defined as $P=\{P_j \mid j=1,2,...,n\}$, where P_j denotes the set of quantities of resources that can be provided by RN-DMU$_j$ which can be defined as $P_j=\{p_{ij} \mid i=1,2,...,m; j=1,2,...,n\}$. p_{ij} denotes the quantity of the i^{th} computing resource that can be provided by RN-DMU$_j$.

We let *Y* denote the set of QoS indexes of RN-DMUs, defined as

$Y=\{Y_j \mid j=1,2,...,n\}$. Y_j denotes the QoS indexes set of RN-DMU$_j$ and can be defined as $Y_j=\{y_{kj} \mid k=1,2,...,s\}$, where y_{kj} denotes the quantity of the k^{th} QoS index that can be provided by RN-DMU$_j$.

We let E be the efficiency score set, defined as $E=\{E_j \mid j=1,2,...,n\}$. E_j denotes the computational efficiency of RN-DMU$_j$, and this score is obtained from the DEA model.

We let A be the set of maximum resource use rates of the RN-DMUs, defined as $A=\{A_j \mid j=1,2,...,n\}$. A_j denotes the set of the maximum use rates of the resources in RN-DMU$_j$ and can be defined as $A_j=\{a_{ij} \mid i=1,2,...,m\}$ where a_{ij} denotes the maximum use rate of the i^{th} resource in RN-DMU$_\text{j}$.

We let B be the set of the occupied resource rates of RN-DMUs. It is similar to A and can be defined as $B=\{B_j \mid j=1,2,...,n\}$ with $B_j=\{b_{ij} \mid k=1,2,...m\}$ where b_{ij} denotes the occupied rate of the i$^{\text{th}}$ resource in RN-DMU$_j$.

We let Z be the decision variable set, defined as $Z=\{z_{lj} \mid l=1,2,...,h; j=1,2,...,n\}$, in which z_{lj} reflects whether RN-DMU$_\text{j}$ is selected as the resource node to complete task l.

We let F be the fitness function. It is used for calculating the total efficiency of the various resource scheduling schemes.

S, the testing software, is installed on the resource nodes to get the parameters of the RN-DMUs. *P*, *Y*, *A*, and *B* of the RN-DMUs can be obtained by running this software on the corresponding resource nodes. Thus, this testing software can obtain the data required for the models.

3. Methodology for measuring efficiencies of the RN-DMUs

In this section, firstly, we select suitable input and output indicators for the RN-DMUs. Then, the production possibility set is analyzed and a suitable model is given for computational efficiency evaluation of the RN-DMUs.

3.1. Input and output selection for RN-DMUs

The first problem faced in this research is how to make the RN-DMUs available for evaluation by the DEA models. It is obvious that the resource nodes are homogeneous. Therefore, if we can establish the input and output vectors for the RN-DMUs, we can then evaluate them using the DEA models.

The inputs of the resource nodes should have the ability of reflecting their computational capacities. Wu et al. (2016) proposed to use the CPU frequency, number of CPUs, memory capacity, and hard drive capacity as the indicators for measuring the computational capability of the computational servers when they studied a technology selection problem. Here, similar to Wu et al.'s (2016) study, we regard the available resources that can be provided by a RN-DMU as its inputs. Therefore, we define the input vector of RN-DMU$_j$ in formula (2).

$$\mathrm{X}_j = \begin{bmatrix} a_{1j}-b_{1j} & 0 & \cdots & 0 \\ 0 & a_{2j}-b_{2j} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & a_{mj-}b_{mj} \end{bmatrix} \times \begin{bmatrix} p_{1j} \\ p_{2j} \\ \cdots \\ p_{mj} \end{bmatrix} \overset{\Delta}{=} (x_{1j}, x_{2j}, \ldots, x_{mj})^T \qquad (2)$$

The considered input indicators (resources) of each RN-DMU$_j$ can be presented as follows.

x_{1j}: CPU frequency (unit: million instructions per second, MIPS) (abbreviation: CPU)

x_{2j} : Internal memory capacity (unit: gigabytes, Gb) (IMC)

x_{3j}: Hard drive capacity (unit: gigabytes, Gb) (HDC)

x_{4j} : Bandwidth (unit: megabytes per second, Mb/s) (BW)

In this paper, the computational efficiencies of the resource nodes are considered. Therefore, the quality of service indexes of the resource nodes should be regarded as the outputs. In real computational situations, the calculation time for a given computational task is the indicator of most concern. Therefore, in this paper, we select the calculation time of the test program as the output of the RN-DMUs. The output of a RN-DMU*j* is denoted as follows.

y_{1j}: Time required (unit: seconds, s) (TR)

As mentioned above, the inputs and output of the RN-DMUs can be automatically obtained by the software in a private cloud computing platform. Therefore, the resource scheduling problem on the private cloud platform transforms into the problem of efficiency evaluation and optimal selection of the RN-DMUs. The number of RN-DMUs is *n*, and each RN-DMU has five inputs and one output.

3.2. The production possibility set

Data envelopment analysis (DEA) is a non-parametric method that has been used widely for measuring the relative efficiencies of a given set of operating entities commonly called decision making units (DMUs) in which similar resources are consumed to create similar products or services (Parkan et al., 2012; Wu et al., 2009). Assume that there are *n* DMUs to be evaluated, and each DMU$_j$ (j=1,2,…,n) has *m* input(s) and *s* output(s), which are denoted $x_{ij}\,(i=1,2,\ldots,m)$ and $y_{rj}\,(r=1,2,\ldots,s)$, respectively. Then, the original production possibility set proposed by Charnes, Cooper, and Rhodes (CCR) (Charnes et al., 1978) under the constant returns to scale assumption can be shown as the following T^0.

$$T^0=\{(X,Y)\mid \sum_{j=`1}^{n}\alpha_j X_j \le X, \sum_{j=`1}^{n}\alpha_j Y_j \le Y, \alpha_j \ge 0, \forall j\} \tag{3}$$

However, this traditional production possibility set cannot be used when evaluating the efficiency of the RN-DMUs in a private cloud environment, since it cannot handle the nondiscretionary inputs (CPU performance and internal memory capacity) and undesirable output (time required) of the RN-DMUs. To surmount this problem, firstly, we distinguish the inputs of a DMU into discretionary inputs and nondiscretionary inputs as in (4).

$$X=\begin{bmatrix} X^D \\ X^N \end{bmatrix} \tag{4}$$

In (4), X^D represents the discretionary inputs and X^N denotes the nondiscretionary inputs. When dealing with non-discretionary inputs in DEA, there are generally three kinds of approaches. Banker and Morey (1986) given the first DEA model that considered the non-discretionary inputs. In their model, both the discretionary and non-discretionary inputs are assumed to have the property of convexity. However, the non-discretionary inputs are handled differently by not allowing them to be radially reduced. The second approach is an extension of Banker and Morey's (1986) approach. Ruggiero (1996) removed the convexity constraints with respect to non-discretionary inputs. Instead, the non-discretionary inputs were used to form restrictions to remove DMUs with higher levels of non-discretionary inputs from the reference set. The third approach is a two-stage approach proposed by Ray (1991). In the first stage the non-discretionary inputs are removed in the evaluation. Then, in the second stage, regressions are used to control the non-discretionary inputs to allow adjustment of technical efficiency. Further extensions of Ray's (1991) approach can be seen in Ruggiero (1998), Yu (1998), and Ruggiero (2001). Later, Ruggiero (2004) pointed out that the above approaches ignored the possibility that technical efficiencies of the DMUs may be correlated with their non-discretionary inputs. He further extended Ruggiero's (1996) model by relaxing the constraints and allowing DMUs with higher levels of non-discretionary input into the reference set as long as the exceeded level is not greater than a given value. Ruggiero's (2004) approach is the most up-to date and comprehensively considered one, and therefore, in this study we adopt their approach for handling the non-discretionary inputs. After considering the non-discretionary inputs, the production possibility set can be reformulated as the following T^1

$$
\begin{aligned}
T^1 = \{(X^D, X^N, Y) \mid & \sum_{j=1}^{n} \alpha_j X_j^D \le X^D \\
& \sum_{j=1}^{n} \alpha_j Y_j \ge Y \\
& X_j^N - (X^N + \delta(X^N)) \le M b_j, \forall j \\
& \alpha_j \le M(1 - b_j), \forall j \\
& b_j \in \{0,1\}, \alpha_j \ge 0, \forall j\}
\end{aligned}
\tag{5}
$$

In (5), $\delta(X^N)$ is the given maximum exceeded level vector in non-discretionary inputs for DMUs in the reference set when evaluating the DMU (X^D, X^N, Y). As can be seen from the third and fourth constraint groups in (5), the λ_j is forced to be zero if $X_j^N > X^N + \delta(X^N)$, which guarantees that only the DMUs with non-discretionary inputs that are no larger than $X^N + \delta(X^N)$ can be included in the reference set. In this study, we set $\delta(X^N)$=$0.15 \times X^N$.

Further, we distinguish the outputs of a DMU into desirable outputs and undesirable outputs as in (6).

$$Y = \begin{bmatrix} Y^G \\ Y^B \end{bmatrix} \tag{6}$$

In (6), Y^G represents the desirable outputs and Y^B denotes the undesirable outputs. When dealing with the undesirable outputs in DEA, four kinds of approaches are generally considered. The first category is to ignore the undesirable outputs (Hua and Bian, 2007). This kind of treatment is not appropriate because undesirable and desirable outputs are produced simultaneously during production. The second kind of approach is to take the undesirable outputs of the DMUs as inputs when evaluating them (Dyckhoff and Allen, 2001), but this method fails to reflect the real production process (You and Yan, 2011). The third kind of approach transforms the undesirable output data first, then the DMUs are evaluated by traditional DEA models using the transformed data (Golany and Roll, 1989; Seiford and Zhu, 2002). The problems here are that the effects of transforming the data on the production set are not clear (Färe and Grosskopf), and the data transformation approach in Seiford and Zhu (2002) can be only used for the BCC and additive DEA models because the CCR model is not translation invariant (Cooper et al., 2007). Another approach for handling undesirable outputs in DEA was proposed in Färe et al. (1989). They incorporated the weak and strong disposability assumptions of production technology into the DEA model. And the weakly disposable factors (undesirable outputs) are treated differently by using strict equal constraints on them. Further discussions of handling weak disposability factors in DEA can be seen in Hailu and Veeman (2001), Färe and Grosskopf (2003), Hailu (2003), and Kousmanen (2005). Among the above four approaches, many high-quality studies (for instance, Chen et al., 2015; Liu et al., 2015; Zanella et al., 2015) used the last of the four mentioned above when undesirable outputs appear. In this paper, we also adopt the strong and weak disposability assumptions for outputs. Since Kuosmanen and Podinovski (2009) proved that the approach of Kuosmanen (2005) is the best for handling weak disposability factors, we incorporate Kuosmanen's (2005) approach for handling the undesirable outputs in this paper.

With the above analysis, the production possibility set can be shown as the following T^2 after considering the undesirable outputs.

$$\begin{aligned} T^2 = \{(X^D, X^N, Y^G, Y^B,) \mid & \sum_{j=1}^{n} \alpha_j X_j^D \le X^D \\ & \sum_{j=1}^{n} \phi_j \alpha_j Y_j^G \ge Y^G \\ & \sum_{j=1}^{n} \phi_j \alpha_j Y_j^B = Y^B \\ & X_j^N - (X^N + \delta(X^N)) \le M \times b_j, \forall j \\ & \alpha_j \le M \times (1 - b_j), \forall j \\ & b_j \in \{0,1\}, \alpha_j \ge 0, 0 \le \phi_j \le 1, \forall j\} \end{aligned} \tag{7}$$

As can be seen in formulation (7), it contains non-linear constraints. To transform it into a linear formulation, let $\alpha_j = \lambda_j + \mu_j$ where $\lambda_j = \phi_j \alpha_j$ and $\mu_j = (1 - \phi_j)\alpha_j$. Then the production possibility set T^2 can be transformed into the following T^3.

$$\begin{aligned} T^3 = \{(X^D, X^N, Y^G, Y^B,) \mid & \sum_{j=1}^{n} (\lambda_j + \mu_j) X_j^D \le X^D \\ & \sum_{j=1}^{n} \lambda_j Y_j^G \ge Y^G \\ & \sum_{j=1}^{n} \lambda_j Y_j^B = Y^B \\ & X_j^N - (X^N + \delta(X^N)) \le M \times b_j, \forall j \\ & \lambda_j + \mu_j \le M \times (1 - b_j), \forall j \\ & b_j \in \{0,1\}, \lambda_j, \mu_j \ge 0, \forall j\} \end{aligned} \tag{8}$$

It can be seen in formulation (8) that all the constraints in it are linear. In addition, we can identify that in (8), the variables μ_j become redundant, specifically, all the feasible output vectors can be obtained when setting μ_j=0 for all *j*. Therefore, the above formulation (8) is equivalent to the following (9).

$$
\begin{aligned}
T^4 = \{(X^D, X^N, Y^G, Y^B,) \mid & \sum_{j=1}^{n} \lambda_j X_j^D \le X^D \\
& \sum_{j=1}^{n} \lambda_j Y_j^G \ge Y^G \\
& \sum_{j=1}^{n} \lambda_j Y_j^B = Y^B \\
& X_j^N - (X^N + \delta(X^N)) \le M \times b_j, \forall j \\
& \lambda_j \le M \times (1 - b_j), \forall j \\
& b_j \in \{0,1\}, \lambda_j, \mu_j \ge 0, \forall j\}
\end{aligned}
\tag{9}
$$

It is easy to identify that null-jointness is also imposed in T^4, which means that if no undesirable output is produced, there will also be no desirable output produced.

3.3. The model

Assume that each DMU$_d$ has *m* discretionary inputs, *k* non-discretionary inputs, *s* desirable outputs, and *t* undesirable outputs. Based on the production possibility set T^4, the model considering discretionary inputs, non-discretionary inputs, desirable outputs and undesirable outputs can be expressed as the following model (10).

$$
\begin{aligned}
\min\ & \theta_d - \varepsilon\Big(\sum_{i \in D} s_i^- + \sum_{r \in G} s_r^+\Big) \\
s.t.\ & \sum_{j=1}^{n} \lambda_j x_{ij}^D + s_i^- = \theta_d \times x_{id}^D, \forall i \\
& \sum_{j=1}^{n} \lambda_j y_{rj}^G - s_r^+ = y_{rd}^G, \forall r \\
& \sum_{j=1}^{n} \lambda_j y_{qj}^B = y_{qd}^B, \forall q \\
& x_{pj}^N - (x_{pd}^N + \delta(x_{pd}^N)) \le M \times b_j, \forall p, j \\
& \lambda_j \le M \times (1 - b_j), \forall j \\
& b_j \in \{0,1\}, \lambda_j \ge 0, \forall j \\
& s_i^- \ge 0, s_p^- \ge 0, and\ s_r^+ \ge 0\ \forall i, p, r
\end{aligned}
\tag{10}
$$

Model (10) can successfully handle the non-discretionary inputs and undesirable output of the DMUs. Therefore, it can be directly used to measure the efficiencies of the RN-DMUs. Assuming that the optimal solution of model (10) for a DMU$_d$ is $\{\theta_d^*, b_j^*, \lambda_j^*, s_i^{-*} s_p^{-*}, s_r^{+*}, \forall i, r, p, j\}$, we give the following definitions.

Definition 1. DMU$_d$ is said to be *DEA efficient* if (1) $\theta_d^* = 1$, and (2) $s_i^{-*} = s_r^{+*} = 0, \forall i, r$.

Definition 2. DMU$_d$ is said to be *weakly DEA efficient* if (1) $\theta_d^* = 1$, and (2) $s_i^{-*} \neq 0$ and/or $s_r^{+*} \neq 0$ for some *i* or *r*.

In this paper, we aim to do resource scheduling in a private cloud platform using the efficiency scores of the RN-DMUs. This goal requires that the efficiency scores for the RN-DMUs discriminate all the RN-DMUs, i.e. produce no ties in the ranking. But the efficiency scores generated from model (10) can only distinguish the DMUs which are neither DEA efficient nor weakly DEA efficient. This method cannot make any further distinction among the DEA efficient and weakly DEA efficient DMUs because the generated efficiency sores for them are all equal to 1. To further discriminate the DEA efficient and weakly DEA efficient DMUs, we incorporate the super-efficiency technique, which was proposed by Andersen and Petersen (1993), into model (10) and propose the following model (11).

$$
\begin{aligned}
\min\ & \theta_d - \varepsilon\left(\sum_{i \in D} s_i^- + \sum_{r \in G} s_r^+\right) \\
s.t.\ & \sum_{j \neq d, j=1}^{n} \lambda_j x_{ij}^D + s_i^- = \theta_d \times x_{id}^D, \forall i \\
& \sum_{j \neq d, j=1}^{n} \lambda_j y_{rj}^G - s_r^+ = y_{rd}^G, \forall r \\
& \sum_{j \neq d, j=1}^{n} \lambda_j y_{qj}^B = y_{qd}^B, \forall q \\
& x_{pj}^N - (x_{pd}^N + \delta(x_{pd}^N)) \leq M \times b_j, j \neq d, \forall p, j \\
& \lambda_j \leq M \times (1 - b_j), j \neq d, \forall j \\
& b_j \in \{0,1\}, \lambda_j \geq 0, j \neq d, \forall j \\
& s_i^- \geq 0, s_p^- \geq 0, and\ s_r^+ \geq 0\ \forall i, p, r
\end{aligned}
\tag{11}
$$

Model (11) excludes the column vector corresponding to DMU$_d$ from the coefficients matrix compared with model (10). This model can be used for distinguishing the DEA efficient and weakly DEA efficient DMUs because the efficiency scores generated from model (11) for the DEA efficient DMUs can be greater than 1. However, the discriminating power of model (11) is still not strong enough since the efficiency scores obtained from model (11) for the weakly DEA efficient DMUs are still all equal to 1. In other words, model (11) still cannot distinguish between the weakly DEA efficient DMUs. To solve this problem, we finally propose the following non-radial model (12) for efficiency evaluation of the DMUs.

$$\begin{aligned}
&\min \frac{1}{m}\sum_{i=1}^{m}\theta_{id} - \varepsilon\left(\sum_{i\in D}s_i^- + \sum_{r\in G}s_r^+\right) \\
&s.t. \quad \sum_{j\neq d, j=1}^{n}\lambda_j x_{ij}^D + s_i^- = \theta_{id}\times x_{id}^D, \forall i \\
&\qquad \sum_{j\neq d, j=1}^{n}\lambda_j y_{rj}^G - s_r^+ = y_{rd}^G, \forall r \\
&\qquad \sum_{j\neq d, j=1}^{n}\lambda_j y_{qj}^B = y_{qd}^B, \forall q \\
&\qquad x_{pj}^N - (x_{pd}^N + \delta(x_{pd}^N)) \leq M\times b_j, j\neq d, \forall p, j \\
&\qquad \lambda_j \leq M\times(1-b_j), j\neq d, \forall j \\
&\qquad b_j\in\{0,1\}, \lambda_j \geq 0, j\neq d, \forall j \\
&\qquad s_i^- \geq 0, s_p^- \geq 0, and\ s_r^+ \geq 0\ \forall i, p, r
\end{aligned} \tag{12}$$

In model (12), the parameters are the same as those in model (10). Assume that $\{\theta_{id}^*, b_j^*, \lambda_j^*, s_i^{-*}, s_p^{-*}, s_r^{+*}, \forall i, r, j, s, p\}$ is an optimal solution of model (12) when evaluating a DMU$_d$. It is easy to verify that $s_i^{-*} = 0, \forall i$. Then,

$$E_d = \frac{1}{m}\sum_{i=1}^{m}\theta_{id}^* \tag{13}$$

is denoted as the final efficiency score of DMU$_d$.

Theorem 1. Assume that the efficiency score of a DMU$_d$ from models (11) and (12) are θ_d^* and E_d, respectively. Then, we have $E_d \leq \theta_d^*$.

Proof. This Theorem can be proved by reference to the proof of Theorem 1 in Wu et al. (2016). We omit the proof here.

Theorem 1 shows that model (12) generally generates for DMU_d an efficiency score that is no larger than that generated from model (11). This is because the non-radial property of model (12) eliminates the slacks of the DMUs which are not DEA efficient, by reducing their efficiency scores in the objective function. Therefore, the efficiency scores for the weakly DEA efficient DMUs will not always be 1 and they can be further distinguished when the slacks to the discretionary inputs are eliminated.

Consider the constraints of model (12). Firstly, it can handle the nondiscretionary inputs in the DMUs. Secondly, it gives special treatment to the undesirable outputs. Thirdly, it can discriminate and rank all the DMUs. Therefore, model (12) is a suitable model for efficiency evaluation of the RN-DMUs in a private cloud environment.

4.The resource scheduling model and the work flow

In this section, the resource scheduling model is given first, then the work flow of the approach is proposed and the process of resource scheduling in private cloud environment is shown.

4.1. The resource scheduling model based on DEA

By using the proposed DEA model (12), we can calculate the relative computational efficiencies of the RN-DMUs. These scores can reflect the computational abilities of the RN-DMUs in the private cloud environment. Therefore, by combining the resource requirements of the calculation tasks, the available quantity of the resources in the RN-DMUs, and the efficiency scores of RN-DMUs, we can simplify the resource scheduling problem on a private cloud platform into the following decision-making problem. When the efficiency score of each RN-DMU is known, we need to discover a way to select the suitable RN-DMUs to construct the corresponding resource scheduling scheme that not only can fulfill the resource demands of the computational tasks but also has the highest total computational efficiency. Therefore, 0-1 programming can be introduced to solve this problem.

Firstly, we make the following assumption.

$$Z_{lj} = \begin{cases} 1, \ RN-DMU_j \ \textit{is selected to execute task } l; \\ 0, \ RN-DMU_j \ \textit{is not selected to execute task } l. \end{cases} \tag{14}$$

On the basis of this assumption and by combining the efficiency scores of the RN-DMUs obtained from model (12), we have an objective function with an efficiency priority point of view. The objective function can be used for representing the total computational efficiencies of the resource scheduling schemes. The objective function is shown as the following (15).

$$F(\mathrm{z}) = \sum_{l=1}^{h}\sum_{j=1}^{n} z_{lj}E_j \tag{15}$$

Because the DEA model used in this paper is input-oriented, the larger E_j is, the more efficient the DMU_j is. With the goal to maximize the fitness function, the efficiency-based resource scheduling model is proposed as the following model (16).

$$\begin{aligned} \max \quad & \sum_{l=1}^{h}\sum_{j=1}^{n} z_{lj}E_j \\ s.t. \quad & \sum_{j=1}^{n} z_{lj} = 1 \ (l = 1,2,...,h) \\ & \sum_{l=1}^{h} z_{lj} \leq 1 \ (j = 1,2,...,n) \\ & x_{ij} \geq \sum_{l=1}^{h} z_{ij} r_{il} \ (i = 1,...,m; \ j = 1,2,...,m) \\ & z_{lj} = 0 \ or \ 1 \end{aligned} \tag{16}$$

In model (16), the first and second constraint groups ensure that each RN-DMU is assigned at most one task in an allocation procedure. The third constraint group guarantees that the resource requirements of each task can be fulfilled by its assigned RN-DMU. Additionally, we can see that the model not only considers various important indicators in the objective function in the resource scheduling procedure (through the use of computational efficiencies of the RN-DMUs as multipliers), but also it has the ability to obtain an optimal resource scheduling result that has the largest total computational efficiency. Finally, it can be seen that unlike the traditional resource scheduling approaches, there is no need to estimate any data when doing resource scheduling using this approach. All the data used in this model can be obtained before the resource scheduling procedure.

Generally, there need to be some other limits in the constraints of model (16). This kind of constraint can be added based on the practical requirements of each submitted task class. For instance, the time should not exceed the time consumed when half of the resource nodes have completed the testing software. Here we omit this kind of constraint, because model (16) is enough to illustrate the idea of resource scheduling from an efficiency priority perspective.

4.2. The workflow of resource scheduling approach

Based on the proposed models (12) and (16), we give the workflow for resource scheduling in a private cloud environment. The detailed workflow can be shown as the following Figure 1.

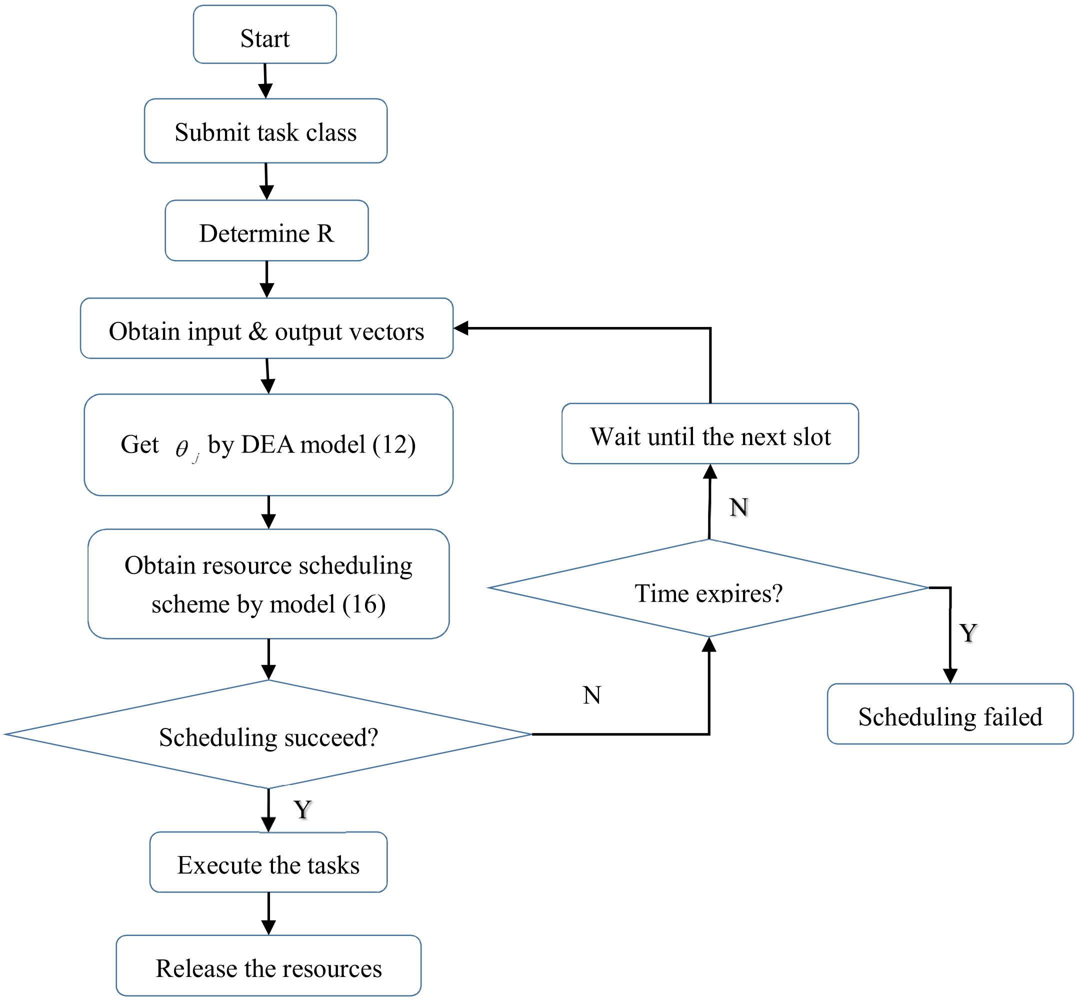

Figure 1. Work flow of the scheduling program

The workflow of this scheduling program can be easily understood through Figure 1, but some further explanation is warranted. Firstly, how do we check whether the scheduling has succeeded? If the tasks are successfully allocated to the resource nodes, the platform then will achieve a successful scheduling. But sometimes the resources of the resource nodes are insufficient to cover the resource requirements of the tasks. In this case, there will be no solution to model (16), which will lead to failure of the scheduling method. Secondly, what will happen if the scheduling fails? In the resource scheduling program, the time is slotted with a slot size equal to Δt. Each scheduling can only start at the beginning of the slot. This means that if the scheduling attempt failed, the algorithm needs to wait until the next slot to restart the scheduling. This process will be repeated until either the scheduling succeeds or its time window expires.

5. Experiment

To test and illustrate the proposed resource scheduling approach, we constructed a private cloud platform with 30 resource nodes (DMUs). In addition, we assumed that 10 task classes, each containing 6 tasks, are submitted to the platform and need to be executed. Firstly, we give descriptive statistical analyses for the data of the DMUs and the 60 tasks. The results are listed in Tables 1 and 2. The detailed data is not provided here but is available from the authors.

Table 1.

Descriptive analysis of the DMUs

Variables	Inputs(N)		Input(D)		Output (B)
	CPU	IMC	HDC	BW	TR
Max	19.75	5	1934	19	4.395
Min	5.58	2	385	5	2.127
Mean	**13.18**	**3.83**	**1109.47**	**12.33**	3.40
Std.dev	4.49	0.95	516.73	4.25	0.76

Table 2.

Descriptive analysis of the Tasks.

Characteristics	Task requirements			
	CPU	IMC	HDC	BW
Max	15.98	4.00	1491.51	14.94
Min	2.39	1.10	210.60	2.25
Mean	**8.76**	**2.41**	**733.39**	**8.08**
Std.dev	3.17	0.74	349.54	3.10

From Tables 1 and 2, we can see that generally the DMUs can fulfill the task requirements of the computational tasks. According to the workflow of the resource scheduling program, we firstly evaluate the DMUs using the proposed model (12) and generate the efficiency scores and ranking positions. Along with these, we also show the evaluation results of model (10) and model (11). The details are listed in Table 3.

Table 3.

Evaluating and ranking results of the DMUs

DMUs	Model (10)		Model (11)		Model (12)	
	Scores	Rank	Scores	Rank	Scores	Rank
1	0.6150	25	0.6150	25	0.5589	25
2	1.0000	1	1.4421	5	1.3926	2
3	0.5869	28	0.5869	28	0.4950	29
4	0.5205	30	0.5205	30	0.4688	30
5	1.0000	1	1.5234	3	1.3869	3
6	1.0000	1	2.8136	1	1.7443	1
7	1.0000	1	1.4815	4	0.9184	10
8	1.0000	1	1.6363	2	1.0028	7
9	0.7727	20	0.7727	20	0.7368	16
10	0.6789	23	0.6789	23	0.6106	24
11	0.5675	29	0.5675	29	0.5151	28
12	0.6123	26	0.6123	26	0.5533	26
13	0.7661	21	0.7661	21	0.6559	21
14	0.8066	19	0.8066	19	0.7328	18
15	1.0000	1	1.0464	8	1.0263	6
16	1.0000	1	1.0335	9	0.8873	11
17	0.8858	15	0.8858	15	0.8049	13
18	0.6620	24	0.6620	24	0.6146	23
19	1.0000	1	1.0000	10	0.9750	8
20	1.0000	1	1.0000	10	0.9597	9
21	0.9808	12	0.9808	12	0.7736	14
22	0.9713	14	0.9713	14	0.7645	15
23	0.9799	13	0.9799	13	0.7117	19
24	1.0000	1	1.3068	6	1.1912	4
25	0.5991	27	0.5991	27	0.5428	27
26	0.6847	22	0.6847	22	0.6484	22
27	0.8154	17	0.8154	17	0.7337	17
28	0.8106	18	0.8106	18	0.6896	20
29	1.0000	1	1.2113	7	1.1691	5
30	0.8447	16	0.8447	16	0.8252	12

Several conclusions can be drawn from the results. Firstly, all these models can effectively evaluate the DMUs and generate efficiency scores. Secondly, model (10) evaluates many DMUs (11 of the 30 DMUs) as DEA efficient or weakly DEA efficient. It cannot make any further distinction among these DMUs because their efficiency scores obtained from model (10) are all equal to 1. Thirdly, compared with model (10), model (11) has better discriminating power and it discriminates almost all the DMUs. For the DEA efficient DMUs, model (11) generates efficiency scores that greater than 1, which ensures the discrimination of the DEA efficient DMUs. Take DMU_2 as an example; its efficiency score from model (10) is 1 but that of model (11) is 1.4421, and its ranking position has changed from 1 to 5. In contrast, for the other DMUs, model (11) generates the same efficiency scores as those generated by model (10). This characteristic has a bad effect on the discrimination of the weakly DEA efficient DMUs. For instance, the efficiency scores of the weakly DEA efficient DMU_{19} and DMU_{20} generated from model (11) are both 1 and their ranking positions are both 10. Thus, model (11) cannot further distinguish them. Fourthly, for all DMUs, the efficiency scores generated from model (12) are no larger than those generated from model (11), which is consistent with the conclusion that was presented in Theorem 1. Finally, we see the evaluation results of model (12): it generates different efficiency scores for all the DMUs and therefore ranks all the DMUs in different positions. Therefore, the proposed model (12) has the best discriminating power among these models and it is a suitable model for the efficiency evaluation of the DMUs in the private cloud environment.

Based on the efficiency evaluation results of the proposed model (11), we then do resource scheduling for the tasks using the proposed resource scheduling model (15). The resource scheduling results are shown in Table 4.

Table 4.

The resource scheduling results

Task class	Resource scheduling scheme (DMUs)						Total efficiency
	task1	task2	task3	task4	task5	task6	
1	16	21	13	8	5	26	5.3549
2	5	27	4	22	21	2	5.5201
3	24	30	5	8	2	17	6.6036
4	14	27	5	8	24	16	5.9347
5	5	26	13	17	9	8	5.2357
6	5	4	21	22	2	16	5.6737
7	5	30	19	6	24	2	7.5152
8	7	2	16	5	24	30	6.6016

9	9	5	30	16	21	8	5.6126
10	8	9	30	21	16	5	5.6126

As can be seen from Table 4, the proposed approach can get each task class the optimal resource scheme which can not only fulfill the resource requirements of the tasks in the task class but also with the highest total efficiency. Taking task class 1 as an example, the proposed resource scheduling approach lets DMU_{16}, DMU_{21}, DMU_{13}, DMU_{8}, DMU_{5}, and DMU_{26} run its tasks 1-6 respectively and the total efficiency for this resource scheduling scheme is 5.3549. Any higher efficiency score for a resource scheduling scheme means that the tasks can be completed with relatively less input and with better output. In other words, less resources are consumed while completing the tasks in a relatively shorter time. Therefore, the proposed resource scheduling approach can be used for resource scheduling in the private cloud computing environment, which provides a new scope and makes a meaningful contribution to the study of resource scheduling in private cloud environments.

6. Conclusion

A resource scheduling approach with an efficiency priority point of view was proposed in this paper for application in a private cloud environment. We established the input and output vectors for the DMUs in a private cloud environment, making choices which ensure that the DEA model can be used for evaluating the DMUs. Then, a suitable DEA model was proposed which can handle nondiscretionary inputs and undesirable output in the DMUs. This model can also effectively measure and completely rank all the DMUs. Based on the efficiency scores obtained from the proposed DEA model for the DMUs, a simple resource scheduling model was proposed to solve the resource scheduling problem in the private cloud environment. The experiment discussed in Section 5 illustrated that the proposed DEA model is suitable for evaluating the DMUs and the resource scheduling model can effectively obtain an optimal resource scheduling scheme for the task classes. Using an efficiency priority point of view, the proposed resource scheduling approach provides a new scope and makes meaningful contributions to resource scheduling in the private cloud environment.

At least two future research directions can be drawn from this research. Firstly, in this paper the DEA method was introduced into the resource scheduling problem for the private cloud environment. Further research efforts may consider whether this effective method can be used in the study of resource scheduling in the environments of public clouds and hybrid clouds. Secondly, we verified the applicability of the proposed approach using simulated data. Further studies may apply the proposed approach to applications where real-world data is available, and compare it with a variety of other resource scheduling approaches. We believe these comparisons will provide some more interesting insights.

References

Andersen, P. and Petersen, N. C. (1993), A procedure for ranking efficient units in data envelopment analysis, *Management science*, Vol. 39 No. 10, pp. 1261-1264.

Banker, R. D. and Morey, R. C. (1986), Efficiency analysis for exogenously fixed inputs and outputs, *Operations research*, Vol.34, No. 4, PP. 513-521.

Bourguiba, M., Haddadou, K. and Pujolle, G. (2012), Packet aggregation based network I/O virtualization for cloud computing, *Computer Communications*, Vol. 35 No. 3, pp. 309-319.

Brumec, S. and Vrček, N. (2013), Cost effectiveness of commercial computing clouds, *Information Systems*, Vol. 38 No. 4, pp. 495-508.

Charnes, A., Cooper, W. W. and Rhodes, E. (1978), Measuring the efficiency of decision making units, *European Journal of Operational Research*, Vol. 2 No. 6, pp. 429-444.

Chen, P. C., Yu, M. M., Chang, C. C., Hsu, S. H. and Managi, S. (2015), The enhanced Russell-based directional distance measure with undesirable outputs: numerical example considering CO 2 emissions, *Omega*, Vol .53, PP. 30-40.

Cooper, W. W., Seiford, L. M. and Tone, K. (2007), Data envelopment analysis: a comprehensive text with models, applications, references and DEA-solver software, Springer Science & Business Media.

Dorigo, M. and Stützle, T. (2003), The ant colony optimization metaheuristic: Algorithms, applications, and advances, In Handbook of metaheuristics (pp. 250-285). Springer US.

Dyckhoff, H. and Allen, K. (2001), Measuring ecological efficiency with data envelopment analysis (DEA), *European Journal of Operational Research*, Vol. 132 No. 2, pp. 312-325.

Elbeltagi, E., Hegazy, T. and Grierson, D. (2005), Comparison among five evolutionary-based optimization algorithms, *Advanced Engineering Informatics*, Vol. 19 No. 1, pp. 43-53.

Färe, R. and Grosskopf, S. (2004), Modeling undesirable factors in efficiency evaluation: comment, *European Journal of Operational Research*, Vol. 157 No. 1, pp. 242-245.

Färe, R., and Grosskopf, S. (2003), Nonparametric productivity analysis with undesirable outputs: comment, *American Journal of Agricultural Economics*, Vol. 85, No.4, PP.

1070-1074.

Färe, R., Grosskopf, S., Lovell, C. K. and Pasurka, C. (1989), Multilateral productivity comparisons when some outputs are undesirable: a nonparametric approach, *The review of economics and statistics*, pp. 90-98.

Golany, B., & Roll, Y. (1989), An application procedure for DEA, *Omega*, 17(3), 237-250.

Gorbenko, A. and Popov, V. (2012), Task-resource scheduling problem, *International Journal of Automation and Computing*, Vol. 9 No. 4, pp. 429-441.

Gu, J., Hu, J., Zhao, T. and Sun, G. (2012), A new resource scheduling strategy based on genetic algorithm in cloud computing environment, *Journal of Computers*, Vol. 7 No. 1, pp. 42-52.

Ho, W. H., Chou, J. H. and Guo, C. Y. (2010), Parameter identification of chaotic systems using improved differential evolution algorithm, *Nonlinear Dynamics*, Vol. 61 No. 1-2, pp. 29-41.

Hua, Z. and Bian, Y. (2007), DEA with undesirable factors, In: J. Zhu and W.D. Cook (eds), Modeling Data Irregularities and Structural Complexities in Data Envelopment Analysis, Spring Science: Boton, MA, pp. 103-121.

Hailu, A. and Veeman, T. S. (2001), Non-parametric productivity analysis with undesirable outputs: an application to the Canadian pulp and paper industry, *American Journal of Agricultural Economics*, Vol. 83, No. 3, PP. 605-616.

Hailu, A. (2003), Nonparametric productivity analysis with undesirable outputs: reply, *American Journal of Agricultural Economics*, Vol. 85, No. 4, PP. 1075-1077.

Huang, L., Chen, H. S. and Hu, T. T. (2013), Survey on Resource Allocation Policy and Job Scheduling Algorithms of Cloud Computing, *Journal of Software*, Vol. 8 No. 2, pp. 480-487.

Ke, C., Huang, Z. and Tang, M. (2013), Supporting negotiation mechanism privacy authority method in cloud computing, *Knowledge-Based Systems*, Vol. 41, pp. 48-59.

Krishna, P. V. (2013), Honey bee behavior inspired load balancing of tasks in cloud computing environments, *Applied Soft Computing*, Vol. 13 No. 5, pp. 2292-2303.

Kuosmanen, T. (2005), Weak disposability in nonparametric production analysis with undesirable outputs, *American Journal of Agricultural Economics*, Vol. 87, No. 4, PP.

1077-1082.

Kuosmanen, T. and Podinovski, V. (2009), Weak disposability in nonparametric production analysis: reply to Färe and Grosskopf, *American Journal of Agricultural Economics*, Vol. 91 No. 2, PP. 539-545.

Laili, Y., Tao, F., Zhang, L., Cheng, Y., Luo, Y. and Sarker, B. R. (2013), A Ranking Chaos Algorithm for dual scheduling of cloud service and computing resource in private cloud, *Computers in Industry*, Vol. 64 No. 4, pp. 448-463.

Liu, W., Zhou, Z., Ma, C., Liu, D. and Shen, W. (2015), Two-stage DEA models with undesirable input-intermediate-outputs, *Omega*, Vol. *56*, PP.74-87.

Ondrich, J. and Ruggiero, J. (2001), Efficiency measurement in the stochastic frontier model, *European Journal of Operational Research*, Vol. 129, No. 2, PP. 434-442.

Parkan, C., Wang, J., Wu, D. and Wei, G. (2012), Data envelopment analysis based on maximin relative efficiency criterion, *Computers & Operations Research*, Vol. 39 No. 11, pp. 2478-2487.

Ray, S. C. (1991), Resource-use efficiency in public schools: a study of Connecticut data, *Management Science*, Vol. 37, No. 12, PP. 1620-1628.

Rosenthal, A., Mork, P., Li, M. H., Stanford, J., Koester, D. and Reynolds, P. (2010), Cloud computing: a new business paradigm for biomedical information sharing, *Journal of biomedical informatics*, Vol. 43 No. 2, pp. 342-353.

Rodero-Merino, L., Caron, E., Muresan, A. and Desprez, F. (2012), Using clouds to scale grid resources: An economic model, *Future Generation Computer Systems*, Vol. 28 No. 4, pp. 633-646.

Ruggiero, J. (1996), On the measurement of technical efficiency in the public sector, *European Journal of Operational Research*, Vol. 90, No. 3, PP. 553-565.

Ruggiero, J. (1998), Non-discretionary inputs in data envelopment analysis, *European Journal of Operational Research*, Vol. 111 No. 3, PP. 461-469.

Ruggiero, J. (2004), Performance evaluation when non-discretionary factors correlate with technical efficiency, *European Journal of Operational Research*, Vol. 159, No. 1, PP. 250-257.

Seiford, L. M. and Zhu, J. (2002), Modeling undesirable factors in efficiency evaluation, *European Journal of Operational Research*, Vol. 142 No.1, pp. 16-20.

Taguchi, G., Chowdhury, S. and Taguchi, S. (2000), Robust engineering, McGraw-Hill Professional.

Takabi, H., Joshi, J. B. and Ahn, G. J. (2010), Security and privacy challenges in cloud computing environments, *IEEE Security and Privacy*, Vol. 8 No. 6, pp. 24-31.

Tsai, J. T., Fang, J. C. and Chou, J. H. (2013), Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm, *Computers & Operations Research*, Vol. 40 No. 12, pp. 3045-3055.

Taylor, M., Haggerty, J., Gresty, D. and Hegarty, R. (2010), Digital evidence in cloud computing systems, *Computer Law & Security Review*, Vol. 26 No. 3, pp. 304-308.

Van Do, T. and Rotter, C. (2012), Comparison of scheduling schemes for on-demand IaaS requests, *Journal of Systems and Software*, Vol. 85 No. 6, pp. 1400-1408.

Wu, J., Liang, L. and Chen, Y. (2009), DEA game cross-efficiency approach to Olympic rankings, *Omega*, Vol. 37 No. 4, pp. 909-918.

Wu, J., Chu, J., Zhu, Q., Yin, P. and Liang, L. (2016), DEA cross-efficiency evaluation based on satisfaction degree: an application to technology selection, *International Journal of Production Research*, DOI: 10.1080/00207543.2016.1148278.

Wu, J., Chu, J., Sun, J., Zhu, Q. and Liang, L. (2016), Extended secondary goal models for weights selection in DEA cross-efficiency evaluation, *Computers & Industrial Engineering*, Vol. 93, pp. 143-151.

Yuan, H., Li, C. and Du, M. (2012), Resource scheduling of cloud computing for node of wireless sensor network based on ant colony algorithm, *Inform. Technol. J*, Vol. 11, pp. 1638-1643.

You, S. and Yan, H. (2011), A new approach in modelling undesirable output in DEA model & star, *Journal of the Operational Research Society,* Vol. 62 No. 12, pp. 2146-2156.

Yu, C. (1998), The effects of exogenous variables in efficiency measurement—a Monte Carlo study, *European Journal of Operational Research*, Vol. 105, No.3, PP. 569-580.

Zhu, Y.X. (2012), Study on the resource allocation algorithm based on ant colony optimization, *Journal of Convergence Information Technology,* Vol. 7 No. 16, pp. 214-223.

Zanella, A., Camanho, A. S. and Dias, T. G. (2015), Undesirable outputs and weighting schemes in composite indicators based on data envelopment analysis, *European Journal of Operational Research*, Vol. *245*, No. 2, PP. 517-530.