

# A Reverse Auction Based Allocation Mechanism in the Cloud Computing Environment

Xingwei Wang<sup>1</sup>, Jiajia Sun<sup>1</sup>, Hongxing Li<sup>2</sup>, Chuan Wu<sup>2</sup> and Min Huang<sup>1</sup>

<sup>1</sup>College of Information Science and Technology, Northeastern University, Shenyang, P.R. China

<sup>2</sup>Department of Computer Science, The University of Hong Kong, Hong Kong

Received: 3 Oct. 2012, Revised: 20 Nov. 2012, Accepted: 1 Dec. 2012

Published online: 1 Feb. 2013

**Abstract:** In the cloud computing, idle resources can be integrated and allocated to users in the form of service. A resource allocation mechanism is in need to effectively allocate resources, motivate users to join the resource pool and avoid fraud among users. Unfortunately, there is little literature addressing this issue. In this paper, we tackle this issue by introducing microeconomic methods into the resource management and allocation in the cloud environment. With the combination of batch matching and reverse auction, a reverse batch matching auction mechanism is proposed for resource allocation. On that basis, we further introduce the strategy of twice-punishment and the pursuit of QoS (Quality of Service) for the purpose of trading fraud prevention. The winner of the auction is then determined by solving an optimization problem that maximizes a weighted sum of three evaluation criteria, i.e., the market efficiency, user satisfaction and QoS. The optimization solution can be readily derived by an improved immune evolutionary algorithm with the application of Vogel's approximation method. We also conduct empirical studies to demonstrate the feasibility and effectiveness of the proposed mechanism.

**Keywords:** Cloud computing, reverse auction, batch matching, Vogel's approximation method, immune evolutionary algorithm

## 1. Introduction

As a novel computational model in Internet computing, cloud computing [1–5] has become a hot topic in both academia and industry around the world. It enables convenient and on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Among all the researches on cloud computing, several significant academic achievements are Nimbus[6], Open Nebula[7], Tsinghua Cloud [8]. Many mainstream IT companies, such as Amazon, Google, Microsoft and IBM, have established new applications for cloud computing, including EC2 (Elastic Cloud Computing) [9], S3 (Simple Storage Service)[10], GAE (Google App Engine)[11], Azure [12], SmartCloud [13]and so on.

Cloud computing has several essential characteristics [14,15]: (1) on-demand: consumers can be provisioned computing or storage capabilities with the quantity that they really need, without any redundancy; (2) pay-as-you-go: consumers only needs to pay for what

they used, and this cost model not only drives down cost but also liberates users from details of the underlying infrastructure; (3) virtualization: kinds of resources from different providers are virtualized to be as a resource pool, and consumers are served through this resource pool; (4) expandability: capabilities can be rapidly and elastically provisioned in any quantity at any time, to quickly scale out and scale in, meeting the needs of applications or users.

Among so many research areas of cloud computing, resource allocation is a hot one. This is not only because that resource allocation mechanism is always the key factor in managing large scale of computational capacity, but also because of the above particular characteristics of cloud resources and the environment.

At present, most IT companies sell cloud resources with fixed-pricing model, but this pricing scheme has a lot of problems: low efficiency, inflexible, less economical, difficulty in forming equilibrium price according to the relation between supply and demand in the market, and so on. Improper allocation methods would cause the above mentioned problems in the system. Fortunately, the

\* Corresponding author e-mail: wangxw@mail.neu.edu.cn

problems can be better addressed by introducing economic models and market-based mechanisms in solving resource allocation in a cloud computing environment.

To the above, we propose a reverse auction based allocation mechanism named Reverse Batch Matching Auction (RBMA) in this paper, targeting on the effective resource allocation in the cloud computing. It is based on a reverse auction, and is improved by batch matching, twice-punishment and QoS (Quality of Service) concerns. To be specific, batch matching improves the performance and efficiency of reverse auction while twice-punishment and QoS constraint prevent fraud and malicious users. Vogel's approximation method and immune evolutionary algorithm are combined to solve the winner determination problem, which is an optimization problem to maximize the weighted sum of market efficiency, user satisfaction and QoS.

The rest of this paper is organized as follows. We discuss related work in Sec. II and present the detailed auction process of RBMA mechanism in Sec. III. The winner determination algorithm is introduced in Sec. IV. In Sec. V, the proposed mechanism is extensively evaluated with empirical study. We conclude the paper in Sec. VI.

## 2. Related work

Several auction-based models were proposed for addressing resource allocation in the cloud computing environment. Lin et al. [16] proposed a second-price auction mechanism which applies the marginal bid to determine the price of the resource for computation capacity allocation with the assistance of pricing and truth-telling mechanism, which ensures the reasonable interests of cloud service providers and effective allocation of computing resource. Prodan et al. [17] proposed a negotiation-based approach for scheduling scientific applications on heterogeneous computing infrastructures such as grids and clouds, and presented a negotiation protocol between the scheduler and resource manager using a market-based continuous double auction model to manage the access to resources in an open market. Shang et al. [18] divided the cloud resource trading market into the futures market and the spot market, and then proposed a knowledge-based continuous double auction trade model and introduced the probability agent based on historical trading information to determine the probability that future bids will succeed, which can achieve higher market efficiency and stable transaction price.

The auction models used in the three literatures mentioned above all belong to single-item auction, and with the development of the research on auction-based models, combinatorial auction as a kind of multi-item auction that can deal with the combinatorial requirements of buyers has widely applied to allocate resources in the

cloud environment. Zaman et al. [19] formulated the problem of virtual machine allocation in clouds as a combinatorial auction problem and proposed three mechanisms: FIXED-PROCE, CA-LP (Combinatorial Auction - Linear Programming), CA-GREEDY (Combinatorial Auction - Greedy) to solve it, and the experimental results showed that CA-GREEDY is better for general purpose VM instance allocation problem while CA-LP can be served for special scenarios. Fujiwara et al. [20] proposed a market-based resource allocation mechanism that allows participants to trade services by means of a double-sided combinatorial auction. This mechanism enables users to order a combination of services for workflows and co-allocations and providers to reserve future/current services in a forward/spot market, which is a little similar to [18], and the two markets run independently to make predictable and flexible allocations at the same time.

Actually, the auction-based model was used as an allocating method in the grid computing earlier and then applied in the cloud environment. Furthermore, auction-based resources allocation has been a hot topic in grid literature for a decade, so many research works in grid computing have great reference value and can be used in cloud computing. Liang et al. [21] proposed a resource allocation model based on reverse auction to allocate grid resources, which can satisfy user's QoS demand on deadline and budget and have better performance than a commodity market-based allocation model. Grosu et al. [22] proposed and investigated first-price auction protocol, Vickrey auction protocol and double auction protocol, and they found the double auction protocol favors both users and resources. Das et al. [23] proposed a resource allocation agreement based on combinatorial auction, in which users bid for every combination of resources and use approximation algorithm to solve the auction problem. Schnizler et al. [24] improved the combinatorial auction, proposed multi-attribute combinatorial auction and the effectiveness of mechanism is proved from the aspects of economy, computing performance and practicality.

However, most of the works rarely take fraud behaviors of malicious users into account and lack the corresponding punishments. In addition, they also rarely take the comprehensive aspects from buyer, seller and market into consideration. In our opinion, the allocation mechanism should be efficient to market and be convenient and fair to the buyer and seller. So we proposed RBMA (Reverse Batch Matching Auction) based on the reserve auction, in which batch matching is introduced to improve the efficiency of reverse auction and twice-punishment mechanism is added to prevent fraud and malicious users. Vogel's approximation method is applied to improve the performance of immune evolutionary algorithm, and then, the paper uses the improved immune evolutionary algorithm to find the optimal resource allocation solution based on the three criteria.

### 3. Auction process

In this section, we present the system model and the detailed auction process of our proposed mechanism. The winner determination algorithm will be introduced in Sec. IV.

#### 3.1. System model

There are three types of participants in the system, Cloud Resource Consumer (CRC), Cloud Resource Provider (CRP) and Auction Intermediary (AI). CRP is an individual or a datacenter that owns resources, while CRC is an individual or company that wants to lease resources. Resources traded in the market can be CPU, memory, storage or bandwidth. Our system targets at IaaS, which is one of the three famous cloud service models [14], and is based on infrastructure.

The system proposed in this paper is suitable for both distributed cloud [25] and social cloud [26]. In distributed cloud, providers do not rely on large and consolidated datacenters, while application developers can selectively lease geographically distributed resources. In social cloud, users can discover and trade storage and computing services contributed by their friends in an online social network.

Fig. 1 shows the system framework. The resource market is composed of providers, consumers and AI. AI controls the whole process and records information of each CRP. Batch-matching mechanism allows AI to wait for more CRCs to send tenders, improving Reverse Auction from the case of 1 CRC with N CRP to that of N CRC with N CRP. Twice-punishment mechanism is used to punish CRCs and CRPs whose offered or bidding prices are not reasonable or even malicious. Resource allocation mechanism is based on the three optimization goals utilizing Immune Evolutionary Algorithm (IEA) and Vogel's approximation method to compute the optimal allocation. Together with the above components, i.e., batch-matching, twice-punishment, IEA and Vogel's approximation method, RBMA is proposed.

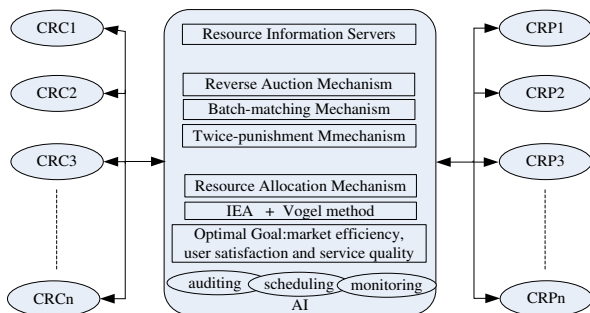


Figure 1 Framework of the resource allocation mechanism

#### 3.2. The buyer tender and behavior

A CRC can be expressed with the following parameters:  $ID_c^i$  represents the unique identity of  $CRC_i$  (the  $i$ -th cloud resource consumer) in the auction platform;  $RP_i$  represents the upper bound of average price that can be accepted by  $CRC_i$ ;  $T_i$  represents the amount of task that  $CRC_i$  submits, and can be regarded as the total amount of resources that the task demands;  $EQ_i$  represents the amount of resources that  $CRC_i$  expects to get in the auction;  $T_{ij}$  represents the amount of resources that  $CRC_i$  obtains from  $CRP_j$  (the  $j$ -th cloud resource provider), i.e., the number of tasks that  $CRP_j$  allocates to  $CRC_i$ ;  $RR_i$  represents the magnitude of  $CRC_i$  increasing reserve price;  $t_{id}^i$  represents the execution deadline of  $CRC_i$ 's task;  $E_i$  represents the total cost that  $CRC_i$  needs to pay for buying the resources in the auction.

The buyer tender which  $CRC_i$  submits to AI is a 5-tuple about the behavior parameters of  $CRC_i$ , i.e.,  $\langle ID_c^i, RP_i, EQ_i, t_{id}^i, RR_i \rangle$ .

CRC will send the buyer tender to AI in the following two conditions: 1, when the entity is initialized, the initial target is sent automatically; 2, after updating information upon the reception of the auction results.

At the end of each auction,  $CRC_i$  calculates the cost based on the amount of allocated resource in the auction as follows,

$$E_i = \sum_{j=1}^J T_{ij} \times BP_j, \tag{1}$$

where,  $BP_j$  is the final bidding price of  $CRP_j$ , and is also the last transaction price.

#### 3.3. The seller tender and behavior

A CRP can be expressed with the following parameters:  $ID_p^j$  represents the unique identity of  $CRP_j$  in the auction platform;  $CP_j$  is the cost of the rented resource, and is the lower bound of average price that can be accepted by  $CRP_j$ ;  $EP_j$  represents the resource transaction price that  $CRP_j$  wants to be sold with, but is also the first asking price of  $CRP_j$  in the auction process;  $BP_j$  represents the bidding price of  $CRP_j$  in the auction;  $TQ_j$  represents the total amount of resources that  $CRP_j$  wants to sell;  $RQ_j$  represents the remaining amount of resources that  $CRP_j$  possesses and wants to sell as the auction continues;  $t_{ss}^j$  represents the time after which the auction task will be serviced;  $t_{se}^j$  represents that  $CRP_j$  will no longer bid after the time;  $t_{sa}^j$  represents the time when the system receives the tender, and is expressed as the time of simulation platform;  $DR_j$  represents the magnitude of  $CRP_j$  lowering bid price;  $I_j$  represents the income that  $CRP_j$  earns through the sale of resources in the auction;  $P_j$  represents the profits that  $CRP_j$  earns through the sale of resources in the auction, and it is the net income.

The seller tender which  $CRP_j$  submits to AI is 8-tuple about behavior parameters of  $CRP_j$ , i.e.,  $\langle ID_p^j, CP_j, EP_j, BP_j, RQ_j, t_{ss}^j, t_{se}^j, DR_j \rangle$ .

The same as CRC, CRP will send tender to AI in the two cases mentioned previously.

At the end of each auction,  $CRP_j$  calculates the income and profit obtained from this auction with Eqn. (2) and (3), respectively, based on the amount of rented resources.

$$I_j = \sum_{i=1}^I T_{ij} \times BP_j, \quad (2)$$

$$P_j = \sum_{i=1}^I T_{ij} \times (BP_j - CP_j). \quad (3)$$

### 3.4. Behavior of auction intermediary

According to the features of different periods in the auction process, an auction can be divided into three stages, i.e., waiting period, preparation period and auction period, and AI has the corresponding behaviors respectively in the three stages.

#### (1) Waiting Period

In this period, both the buyer tenders and seller tenders are received and stored with specific order. Among them, the seller tenders are stored in the ascending order of  $t_a^j$ , and the buyer tenders are stored in ascending order in accordance with  $t_{id}^j$ . In the waiting period, the most important behavior of AI is to start an auction when the system receives the first buyer's tender, which means the start of the auction. AI uses a timer to control the beginning and end of the waiting period based on the length of waiting period, and the duration of waiting period can be preset and changed in the simulation.

#### (2) Preparation Period

In the preparation period, AI needs to deal with the tenders which are received in the waiting period. The seller tender can participate in this auction upon the satisfaction of the following three conditions concurrently,

$$\begin{cases} AP_s + AP > t_{ss}^j, \\ AP_s + AP < t_{se}^j, \\ RQ_j > 0, \end{cases} \quad (4)$$

where,  $AP_s$  is the start time of auction period, and  $AP$  is the maximum duration of auction period, so  $AP_s + AP$  is the latest end time of auction.

So the three inequalities can be respectively interpreted that (a) the start time of service is earlier than the end time of auction; (b) the end time of service is later than the end time of auction; (c) the amount of remaining resource is greater than zero.

The buyer tender cannot take part in the auction until both of the following conditions are met,

$$\begin{cases} AP_s + AP < t_{id}^j, \\ EQ_i > 0, \end{cases} \quad (5)$$

where, they respectively represent the task deadline is later than the end time of auction; the amount of resource that is still desired is greater than zero.

AI will respectively traverse the two queues, deleting the tenders which don't meet the requirements from the queues, and move the qualified tenders to the two queues maintained by AI.

#### (3) Auction Period

In the auction period, AI is responsible for controlling the bidding process and determining the final transaction price. Then, the resource allocation scheme is optimized based on the transaction price. In addition, AI needs to coordinate CRCs to grade the service of CRPs. For the whole process, AI is just responsible for sending the appropriate message labels, while the corresponding entities complete the specific operations.

After the resource allocation, every CRC needs to grade the service of the CRP which has transacted with him/her. This is to enable CRC to get better service, improve the overall QoS, and QoS is a basis of optimizing resource allocation scheme.

The initial score of each CRP, upon its first participation in the auction, is set to be 0.5. At the end of each auction, AI updates all the scores as follows,

$$Score_{j\_new} = \theta_1 \times Score_{j\_old} + \theta_2 \times \left( \sum_{i=1}^I score_{ij}/m \right), \quad (6)$$

where,  $I$  is the total number of CRCs participating this auction, and  $m$  is the number of CRCs that have graded  $CRP_j$ ,  $score_{ij}$  is the score that  $CRC_i$  grades the service of  $CRP_j$ , and it ranges from 0 to 1. So this formula represents the updated score of  $CRP_j$  is the weighted sum of  $Score_{j\_old}$  (historical score) and  $\left( \sum_{i=1}^I score_{ij}/m \right)$  (new average score);  $\theta_1$  and  $\theta_2$  are the positive weights.

The historical score  $Score_{j\_old}$  is stored in a database, and the primary key is  $ID_p^j$ , which is unique for every CRP. When we need to compute the score of  $CRP_j$ , we can get this score from database in terms of its  $ID_p^j$ .

### 3.5. Auction process

A complete cloud resource auction process may be executed for multiple consecutive auctions until there is no untreated demand, while each auction is composed of several rounds but with bounded maximum round number.

For a particular auction, the beginning of auction period means the beginning of this auction.

The end condition of the auction is that the current auction round is greater than the default maximum round, or the price meets the requirement specified as follows,

$$\left(\sum_{j=1}^J RQ_j \times BP_j / \sum_{j=1}^J RQ_j\right) \leq \left(\sum_{i=1}^I EQ_i \times RP_i / \sum_{i=1}^I EQ_i\right). \quad (7)$$

It represents that the average price of resource calculated in accordance with CRP's bid price is not more than that calculated in accordance with CRC's reserve price.

In an auction process, when the end conditions of the auction cannot be reached due to the price, the bid price of CRP and the reserve price of CRC will be updated in accordance with the corresponding strategy and the parameters in tenders.

The bid price of CRP will be updated in every round, and the specific updating strategy is specified in Eqn. (8).

$$BP_{j\_new} = \begin{cases} BP_{new} & BP_{new} \geq CP_j, \\ CP_j & BP_{new} < CP_j, \end{cases} \quad (8)$$

where,  $BP_{new} = BP_{j\_old} \times (1 - DR_j)$ .

The reserve price of CRC will be updated in every two rounds, and the specific updating strategy is specified in Eqn. (9).

$$RP_{i\_new} = \begin{cases} RP_{new} & RP_{new} \leq 1, \\ 1 & RP_{new} > 1, \end{cases} \quad (9)$$

where,  $RP_{new} = RP_{i\_old} \times (1 + RR_i)$ .

As discussed above, the detailed steps of the whole auction are as follows,

Step1: Auction enters into waiting period when AI receives the first buyer tender. Then, AI continues to receive tenders from buyers and sellers.

Step2: Auction enters the preparation period when timer sends a message indicating the end of waiting period. The specific process in this period is as follows,

a) AI checks the queue of seller tender, and removes the tenders which do not satisfy the three constraints in formula (4). After that if the queue still has tenders, go to Step 2.b; otherwise, go to Step 4;

b) AI checks buyer tender queue, and removes the tender which do not satisfy the two constraints in formula (5). After that, if the queue still has tenders, go to Step 3; otherwise, go to Step 4;

Step3: Auction enters into auction period. The specific process in this period is as follows,

a) If the current auction round does not exceed the default maximum value, CRCs and CRPs bid round by round, and update their bid price or reverse price in accordance with the strategy specified in Eqn. (8) and Eqn. (9), respectively; otherwise, go to Step 3.b;

b) Using twice-punishment mechanism to end the auction;

c) Applying IEA to optimize resource allocation scheme (specified in Section 4.2); resources start to be allocated, and users grade the services they get;

Step4: AI checks the seller tender queue. If there has no unprocessed tender, the auction finishes; otherwise, go to Step1.

### 3.6. Twice-punishment mechanism

Fraud behaviors of malicious users may arise during the cloud resource auction. And possibly, CRC and CRP could make inaccurate estimation on the price of resources they want to buy or to sell. These two kinds of circumstances will both lead to the results that final transactions prices can't reach the end of auction condition specified in (7). Twice-punishment mechanism is designed in the paper, not only to avoid the occurrence of these two situations, but also to play the role of incentive compatibility.

The first punishment takes place in the final round of the auction. To be specific, when it is already in the final round of the auction, the auction is no longer based on the original auction strategy. Then, the first punishment will be executed in the way that AI enforces every CRP to bid at its  $CP_j$  and enforces every CRC to increase its  $RP_i$  n times successively in accordance with updating strategy of reserve price.

The second punishment is executed in the additional round. In more detail, if the auction has experienced the first punishment but has not yet reached the end condition of auction, the auction enters the additional round, where the final punishment will be carried out by setting cost price of every CRP to be the unified median price (half of the maximum price set in advance), which will serve as the bidding price; setting the reserve price of every CRC also to be the unified median price. At this point, the price just satisfies the end condition of auction specified in (7).

## 4. Optimization algorithm

### 4.1. Evaluation criteria

Three evaluation criteria, i.e., market efficiency, user satisfaction and QoS, are introduced in order to evaluate an allocation scheme. In more detail, market efficiency is derived from the perspective of market while user satisfaction and QoS are on the users' perspective to respectively characterize the satisfaction levels with the price and service.

#### (1) Market Efficiency

In economics, the market efficiency in auction is defined that the item for sale is finally obtained by the bidder who bids with the highest price [27].

In the cloud resource auction mechanism, if CRP bids at a lower price, it indicates that the resource is more

available and/or the sale is more urgent. If CRC sets a higher reserve price, it represents that he/she is willing to spend more money on the resource and the task is more urgent. Therefore, the market efficiency of the cloud resource auction mechanism is that more available resource is allocated to the more urgent customers. The average market efficiency is defined as in Eqn. (10).

$$Eff = \sum_{i,j=1}^{I,J} (RP_i - BP_j) \times T_{ij} / \sum_{i,j=1}^{I,J} T_{ij}, \quad (10)$$

where,  $RP_i$  represents the urgency degree of  $CRC_i$ 's task;  $BP_j$  represents the degree of availability of  $CRP_j$ 's resource. Therefore,  $(RP_i - BP_j)$  represents the market efficiency in  $T_{ij}$ .

#### (2) User Satisfaction

User satisfaction represents the CRC's satisfaction on the bidding price of CRP, which is jointly determined by reserve price and bid price. The average satisfaction is defined as in Eqn. (11).

$$Sat = \sum_{i,j=1}^{I,J} \left( \frac{RP_i - BP_j}{RP_i} \times T_{ij} \right) / \sum_{i,j=1}^{I,J} T_{ij}, \quad (11)$$

where,  $\frac{RP_i - BP_j}{RP_i}$  represents  $CRC_i$ 's satisfaction on the bidding price of  $CRP_j$ .

#### (3) QoS

QoS represents CRC's satisfaction degree with the resource service when CRC gets the right to use the resource after each auction. In this paper, it is the score that CRC grades CRP in accordance with QoS. The average QoS is defined as follows,

$$Qua = \sum_{i,j=1}^{I,J} Score_j \times T_{ij} / \sum_{i,j=1}^{I,J} T_{ij}, \quad (12)$$

where,  $Score_j$  is the comprehensive historical score of  $CRP_j$ , and after each auction,  $Score_j$  will be updated in accordance with Eqn. (6).

## 4.2. Improved IEA

Based on the three evaluation criteria, we can have that the optimization objective of the resource allocation scheme is to maximize the three evaluation criteria.

#### (1) Optimal Goal

In this paper, the optimization objective is a weighted sum of the three evaluation criteria and can be defined as follows,

$$f(T) = \alpha \times Eff + \beta \times Sat + \gamma \times Qua, \quad (13)$$

where,  $Eff$ ,  $Sat$  and  $Qua$  are specified in Eqn. (10), (11) and (12), respectively.  $\alpha$ ,  $\beta$  and  $\gamma$  are non-negative weights of these criteria.

Up to now, the problem has been transformed into finding an allocation scheme  $T$  that maximizes  $f(T)$ . The immune evolutionary algorithm [28,29] is adopted to solve the optimization problem here.

#### (2) Antibody Encoding

Each antibody stands for a resource allocation scheme and is characterized by a matrix (two-dimensional variable-length array). The matrix size depends on the number of CRC and CRP. The element  $T_{ij}$  in the matrix represents the task amount that  $CRP_j$  allocates to  $CRC_i$ , the value of each element (gene) uses real coding from  $[0, 1]$ .

#### (3) Initialization

Considering the fact that the quality of initial population has great influence on the algorithm's convergence and stability, and that the initial population of the traditional immune algorithm is completely randomly generated leading to the great uncertainty about its coverage area, it is difficult to effectively find the optimal solution.

An initial population of high quality should maintain the diversity in order to improve the global convergence of algorithm, and also should include several representative individuals to accelerate the convergence rate and avoid blind search. Based on the analysis above, this paper abandons the traditional method of initial population generation, but utilizes the Vogel's approximation method which is used to resolve the problem on assignment and transportation in operational research to generate several excellent individuals to join in the initial population.

The process of applying Vogel's approximation method to generate the initial population of immune evolutionary algorithm is as follows,

Step1: Initialize every element in allocation scheme  $T$  to zero, and define the evaluation of point  $T_{ij}$  with Eqn. (14).

$$\varphi(i, j) = \alpha \times Eff_{ij} + \beta \times Qua_{ij} + \gamma \times Sat_{ij}. \quad (14)$$

Step2: Calculate the difference between the maximum and the second maximum value of  $\varphi(i, j)$ ,  $\forall i \in [1, I]$  or  $\forall j \in [1, J]$ ;

Step3: Find the maximum one among the  $(I + J)$  differences, and the maximum difference is corresponding to a certain line or row in  $T$ , and then choose the point  $(i, j)$  with the maximum value of  $\varphi(i, j)$  in that line or row we just mentioned. Then, we can determine the amount of resource that  $CRC_i$  can get from  $CRP_j$ , i.e.,  $T_{ij} = \min(EQ_i, RQ_j)$ ;

Step4: If  $\min(EQ_i, RQ_j) = EQ_i$ , which means  $CRP_j$  meets the demand of  $CRC_i$ , we let the  $\varphi(i, j)$  of elements in the  $i$ -th line of matrix  $T$  to be  $-\infty$ ; otherwise, i.e.,  $\min(EQ_i, RQ_j) = RQ_j$ , which means all the resource of  $CRP_j$  has been sold to  $CRC_i$ , we set the  $\varphi(i, j)$  of elements in  $j$ -th row of matrix  $T$  to be  $-\infty$ ;

Step5: If all CRPs' resource have been sold out or all CRCs' demand have been satisfied, a resource allocation

scheme (matrix  $T$ ) is derived, and go to Step6; otherwise, go to step2;

Step6: Generate  $N_{vogel}$  solutions in the neighborhood of scheme  $T$  we just get from Vogel's approximation method, and then randomly generate other solutions to make up the whole initial population.

From the steps above, we can see that the initial population is composed of three parts: 1, the solution directly got from Vogel's approximation method; 2, the solutions generated near the first one; 3, the solutions generated randomly.

(4) Adjustment

Adjusting operation is used to adjust every antibody in  $q(0)$  is adjusted so that they can meet the three requirements as specified in Eqn. formula (15), formula (16) and formula (17);

$$T_{ij} > 0 \iff RP_i - BP_j \geq 0, \tag{15}$$

$$\sum_{i=1}^I T_{ij} \leq RQ_j, \tag{16}$$

$$\sum_{j=1}^J T_{ij} \leq EQ_i. \tag{17}$$

For the gene which does not satisfy formula (15), reset its value  $T_{ij}$  as 0; for the one which fails to satisfy formula (16) or formula (17), reset its value according to Eqn. (18).

$$T_{ij} = \min[(T_{ij} / \sum_{i=1}^I T_{ij}) \times RQ_j, (T_{ij} / \sum_{j=1}^J T_{ij}) \times EQ_i]. \tag{18}$$

By adjusting operation, every antibody is qualified to represent a resource allocation scheme.

(5) Selection, Clone and Mutation

Selection operation is to select the antibodies whose stimulation is higher than the average in the population. To get stimulation of antibody, we must compute affinity, similarity and density in advance. The result of Eqn. (13) is the affinity of antibody. Euclidean distance is used to represent the similarity of two antibodies, as specified in Eqn. (19). Then, we can get the density, as specified in Eqn. (20). Stimulation of antibody is proportional to affinity, but inversely proportional to density, as specified in Eqn. (21).

$$sim(T^a, T^b) = \sqrt{\sum_{i,j=1}^{I,J} (T_{ij}^a - T_{ij}^b)^2}, \tag{19}$$

$$den(T^a) = \frac{\sum_{b=1}^N sim(T^a, T^b)}{N}, \tag{20}$$

$$sti(T^a) = \omega_1 \times aff(T^a) + \omega_2 \times den(T^a). \tag{21}$$

Clone operation makes antibodies to replicate themselves according to the cloning size as follows,

$$num_a = n_c \times (aff(T^a) / \sum_{b=1}^N aff(T^b)), \tag{22}$$

where,  $num_a$  is the number that antibody  $T^a$  replicates itself;  $n_c$  is the default total number of antibodies gotten by cloning.

Mutation operation replaces the gene value with uniformly or normally distributed random numbers within  $[0, \min(EQ_i, RQ_j)]$  according to the mutation probability as follows,

$$P_m(T^a) = \{2 - (N \times aff(T^a)) / \sum_{b=1}^N aff(T^b)\} \times P_{m,ave}. \tag{23}$$

(6) Process of IEA

The overall process of immune evolutionary algorithm is as follows,

Step1: Set the initial population size as  $N$ . Follow the steps specified in 4.2.3 (Initialization) and 4.2.4 (Adjustment) to initialize and adjust antibodies in  $q(0)$ .

Step2: For the population  $q(t)$ , iterate the following operations until the loop reaches the maximum number of iterations ( $T_{max}$ ) or memory cell has not been updated for several continuous generations ( $T_{con}$ ).

a) Conduct selection, clone and mutation, as specified in 4.2.5.

b) Use adjusting operation to adjust the mutated antibodies.

c) Substitute poorer individuals with mutated individuals with higher affinity in  $q(t)$ ; generate new antibodies to substitute antibodies with lower stimulation.

Step3: Regard the antibody with the highest affinity in the population as the result of immune evolutionary algorithm.

### 5. Simulation and evaluation

The reserve batch matching auction mechanism is implemented and evaluated based on Simjava2.0 toolkit [30] on the Eclipse platform. Our simulation results demonstrate the superiority of the proposed algorithm over its counterparts in literature [21].

The buyer data, i.e.,  $EQ_i, t_{td}^i$  and  $RR_i$ , and the seller data, i.e.,  $CP_j, EQ_j, RQ_j, t_{ss}^j, t_{se}^j$  and  $DR_j$ , are generated randomly. In addition, the parameters of immune evolutionary algorithm are detailed in Table.1.

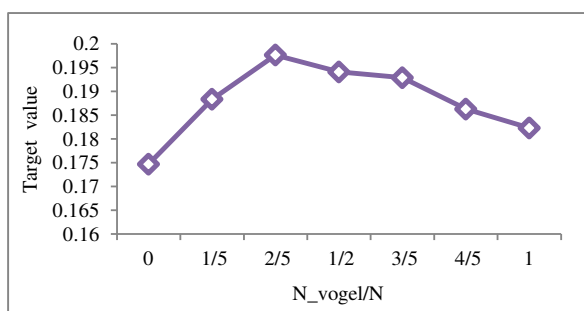
Each datum in the following figures is the average of 10 trials under the corresponding simulation conditions.

Table 1 Relative Parameter to IEA

| Relative parameter  | valuation |
|---|-----------|
| Population size $N$   | 20        |
| Maximum iteration number $T_{max}$  | 50        |
| Maximum continuous iteration number $T_{con}$                                       | 10        |
| Average mutation probability $P_{m\_ave}$   | 0.03      |
| Proportion of individuals generated from Vogel's approximation method $N_{vogel}/N$ | 2/5       |

Fig. 2 shows target values of resource allocation scheme when  $N_{vogel}/N$  varies. The ordinate (target value) is the mean of market efficiency and user satisfaction. In the simulation, the value of  $score_{ij}$  is randomly generated with normal distribution, so it has no meaning to evaluate the third evaluation criteria (QoS). So, it is not included in the target value here and in the comparison of two mechanisms to be discussed.

We can draw two conclusions from Fig.2: 1, when the value of  $N_{vogel}/N$  is 2/5, it is helpful for the process of optimization; 2, the target value is poor when  $N_{vogel}/N$  equals zero.  $N_{vogel}/N$  equals zero means that Vogel's approximation method is not used in the process of optimization, and the traditional method is used to generate initial population. This proves that using Vogel's approximation method to generate several antibodies to join in the population is significant, and Vogel's approximation method improves the performance of immune evolutionary algorithm.

Figure 2 The change of target value when  $N_{vogel}/N$  varies

We also conduct performance analysis between RBMA proposed in this paper and Reverse Auction (RA) [21], which used first-price sealed-bid reverse auction such that the winner provider is the one with the lowest bid.

The comparisons on two evaluation criteria, i.e., market efficiency and user satisfaction, between the two auction mechanisms are presented in Fig. 3 and Fig. 4, respectively.

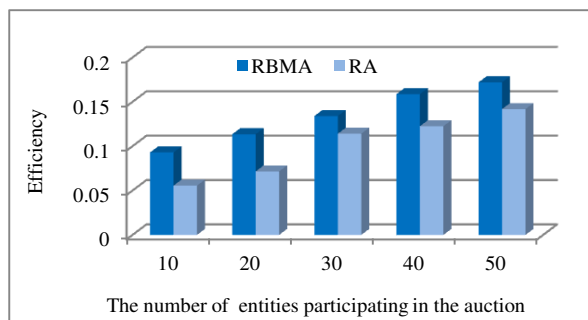


Figure 3 The comparison of market efficiency

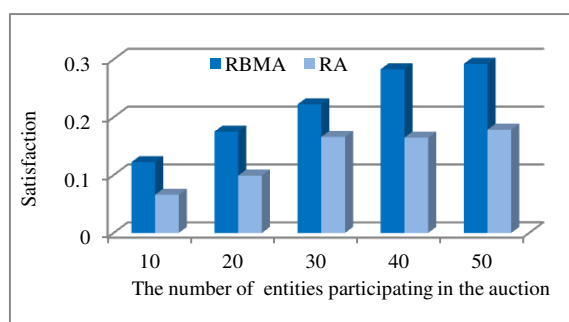


Figure 4 The comparison of user satisfaction

As shown in Fig. 3 and Fig. 4, the resource allocation scheme using RBMA in the auction is superior to RA in both market efficiency and user satisfaction.

It can be mainly explained that, in reverse auction, one auction can only take into account the needs and interests of one CRC, and is similar to a greedy allocation strategy compared with batch matching. That is to say, in reverse auction, the cheapest resources are only allocated to the current user, while ignoring other users and the whole utility, resulting in little possibility to a global optimal result.

Next, we examine both the average and variance of the transaction price acquired under the two auction mechanisms and show the results in Fig. 5 and Fig. 6, respectively.

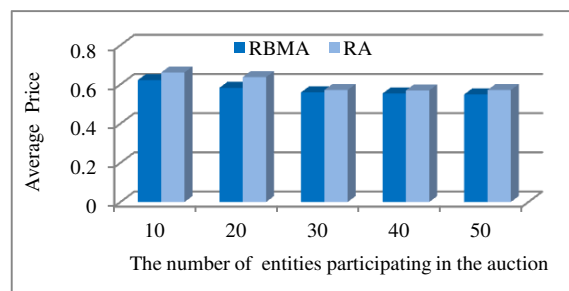


Figure 5 The comparison of average price

As shown in Fig. 5, the average transaction price acquired by RBMA is slightly lower than that using RA,



which indicates that RBMA can get more benefits than RA does for CRC.

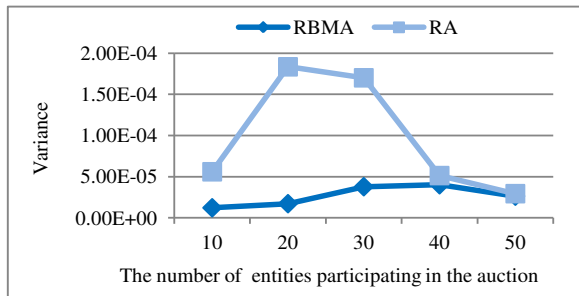


Figure 6 The comparison of variance of price

In Fig.6, we have that the fluctuation of transaction price acquired by RA is larger than that using RBMA. This is because that the resource allocation strategy in RA is closely related to the arrival order of buyer tenders, while the first-come-first-served mode will inevitably lead to obviously different results upon each execution.

From Fig. 3, Fig. 4 and Fig. 5, we can also conclude that with the increased number of entities participating in the auction, the satisfaction and efficiency are improved while the average price decreases. The rationale is that there will be more chances for our model to allocate better resources to users with the increased number of entities. Certainly, the data of every entity (CRC or CRP) also have influence on the results.

## 6. Conclusion

In this paper, methods in economics and operational research are introduced for effective resource allocation in cloud computing. A reverse batch matching auction mechanism is proposed to manage and allocate the resources, while twice-punishment mechanism is additionally implemented to avoid the fraud between buyers and sellers using historical information of QoS. The resource allocation is then the solution to a maximization problem of a weighted sum of three evaluation criteria, and can be calculated with an immune evolutionary algorithm improved by Vogel's approximation method. The simulation results validate the feasibility and demonstrate the superiority of the proposed mechanism on improving both market efficiency and resource utilization. It constitutes our future work to classify the resource in detail, and to record and apply more historical information for better performance.

## Acknowledgement

This work is supported by the National Science Foundation for Distinguished Young Scholars of China

under Grant No. 61225012; the National Natural Science Foundation of China under Grant No. 61070162, No. 71071028 and No. 70931001; the Specialized Research Fund of the Doctoral Program of Higher Education for the Priority Development Areas under Grant No. 20120042130003; the Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20100042110025 and No. 20110042110024; the Specialized Development Fund for the Internet of Things from the ministry of industry and information technology of the P.R. China; the Fundamental Research Funds for the Central Universities under Grant No. N110204003.

## References

- [1] G. Pallis, IEEE Internet Computing **14**, 70-73 (2010).
- [2] Q. Zhong, Appl. Math. Inf. Sci. **6**, 105S-109S (2012).
- [3] Y. Wei and M. B. Blak., IEEE Internet Computing **14**, 72-75 (2010).
- [4] X.H. Wu, M. C. QU, Z.Q. Liu and J.Z Li, Appl. Math. Inf. Sci. **6**, 1S-8S (2012).
- [5] B. Peng, Appl. Math. Inf. Sci. **6**, 99S-103S (2012).
- [6] <http://www.nimbusproject.org/>.
- [7] <http://www.opennebula.org/>.
- [8] <http://grid.tsinghua.edu.cn/hpcgrid/GCD/cloud/cloud.htm>.
- [9] <http://aws.amazon.com/ec2>.
- [10] <http://aws.amazon.com/s3/>.
- [11] <http://appengine.google.com>.
- [12] <http://www.microsoft.com/windowsazure/>.
- [13] <http://www.ibm.com/cloud-computing/>.
- [14] P. Mell and T. Grance, Definition of cloud computing, Technical report, National Institute of Standard and Technology (NIST),2009.
- [15] S. Sakr, A. Liu, D. M. IEEE Communications Surveys & Tutorials **13**, 311-336 (2011).
- [16] W. Y. Lin, G. Y. Lin and H. Y. Wei, IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 591-592 (2010).
- [17] R. Prodan, M. Wiecek and M. Frad, Journal of Grid Computing **9**, 531-548 (2011).
- [18] S. Shang, J. Jiang, Y. Wu, Z. Huang, G. Yang and W. Zheng, IFIP international conference on Network and parallel computing, 155-164 (2010).
- [19] S. Zaman and D. Grosu, International Conference on Cloud Computing Technology and Science, 127-134 (2010).
- [20] I. Fujiwara, K. Adia and I. Ono, International Symposium on Applications and the Internet, 7-14 (2010).
- [21] Z. Liang, Y. Sun, L. Zhang and S. Dong, Pacific Rim International Workshop on Multi-Agents (PRIMA), 150-161 (2006).
- [22] D. Grosu and A. Das, Concurrency and Computation: Practice & Experience **18**, 1909-1927 (2006).
- [23] A. Das and D. Grosu, Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS), (2005).
- [24] B. Schinzler, D. Neumann, D. Veit and C. Weinhardt, European Journal of Operational Research **187**, 943-961 (2006).

- [25] P.T. Endo, A.V. de Almeida Palhares and N.N. Pereira, *IEEE Network***15**, 42-46 (2011).
- [26] K. Chard, S. Caton and O. Rana, *International Conference on Cloud Computing(CLOUD)*, 99-106 (2010).
- [27] K. Vijay, D. M. Luo, and X. C. Xi, *Auction Theory*, Beijing, Chian Renmin University Press, 2010.
- [28] Y. Yu and C. Hou, *Computer Engineering* **32**, 167-168 (2006).
- [29] G. M. Lu and X. B. Wang, *Computer Engineering* **34**, 200-205 (2008).
- [30] F. Howell and R. McNab, *First International Conference on Web-based Modelling and Simulation*, (1998).



**Xingwei Wang** received the PhD degree in computer science from Northeastern University, Shenyang, China, in 1998. He is currently a professor in the faculty of information science and engineering, Northeastern University. His research interests include resource

management for cloud computing, optimized routing algorithms in Future Internet. He has published over 100 scholarly publications in *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Communications*, *IEEE Transactions on Industrial Electronics*, *ACM Operating Systems Review*, *Computer Networks*, *Computer Communications*, etc.



**Jijia Sun** received the bachelor degree in computer science and technology from Northeastern University, Shenyang, China, in 2012. Currently, she is studying as a master degree student of computer application technology in Northeastern University. Her research

interests include resource management for cloud computing and social cloud. She has published several scholarly publications on resource auction schemes for cloud computing and routing mechanisms in Internet. She has won National Scholarship.



**Hongxing Li** received his B.Sc. and M.Eng. degrees in 2005 and 2008 from the Department of Computer Science and Technology, Nanjing University, China. He is currently a PhD candidate in the Department of Computer Science, the University of Hong Kong,

China. His research interests include wireless networks and cloud computing.



**Chuan Wu** received her B. Engr. and M.Engr. degrees in 2000 and 2002 from Department of Computer Science and Technology, Tsinghua University, China, and her Ph.D. degree in 2008 from the Department of Electrical and Computer Engineering, University of Toronto, Canada. She is

currently an assistant professor in the Department of Computer Science, the University of Hong Kong, China. Her research interests include cloud computing, peer-to-peer networks and online/mobile social networks.



**Min Huang** received the PhD degree in system engineering from Northeastern University, Shenyang, China, in 1999. She is currently a professor in the faculty of information science and engineering, Northeastern University.

Her research interests include optimal algorithm, scheduling and inventory control. She has published over 60 scholarly publications.