Combinatorial Double Auction Winner Determination in Cloud Computing using Modified Simulated Annealing Algorithm

Farhad Imani and Ali Asghar Pourhaji Kazem*,

Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran
*Corresponding author

Abstract
With the advancement of information technology need to perform computing tasks everywhere and all the time there. In cloud computing environments and heterogeneous users have access to different sources with different characteristics that these resources are geographically in different areas. Due to this, the allocation of resources in cloud computing comes to the main issue is considered a major challenge to achieve high performance. Due to the nature of cloud computing is a distributed system to account comes to business, economic methods such as auctions are used to allocate resources for decentralized. As an important economic bilateral hybrid auction model is the perfect solution for the allocation of resources in cloud computing. On the other hand, providers of cloud resources similarly, their sources of supply combined addressing. One of the problems auction two-way combination with maximum benefit for the parties to the transaction is the efficient allocation of resources to the problem of determining an auction winner is known. Given that the winning auction is NP-Hard It results in a problem, several methods have been proposed to solve it. In this dissertation, taking into account the strength simulated annealing algorithm, a modified version of it is proposed for solving the winner determination in combinatorial double auction problem in cloud computing. The proposed approach is simulated along with genetic and simulated annealing algorithms and the results show that the proposed approach find better solutions than the two mentioned algorithms.
Keywords:
Cloud Computing, Double Auction, Winner Determination, Simulated Annealing.

1. Introduction

Resource management is one of the key challenges in cloud computing and cloud data center management [1]. Most cloud providers use fixed price mechanisms to allocate resources to users. But these mechanisms do not provide an efficient and acceptable allocation of resources, and in fact cannot maximize the profitability of cloud resource providers [2], [3]. In such a situation, cloud-based economic models are suited to tune, deliver, and demand resources. An appropriate option for allocating resources in cloud computing is the use of bidding mechanisms. Among the auctioning mechanisms, the most appropriate mechanism used in cloud computing for allocation and pricing of resources is a combinatorial auction mechanism.

In this way, prices depend on the conditions of demand and supply rather than the fair exchange between cloud providers and users. Considering the above, the use of combinatorial double auction to allocate resources in cloud computing can be a very appropriate model [4]–[7]. The double auction mechanism consists of two steps. The first step is to determine the winning bidder by solving an optimization model that aims to maximize social welfare by taking into account the payment of users and the profit of the providers. The second step is the allocation and pricing of resources among the winners. However, it has been proved that the problem of winner determination of the auctions is a NP-hard problem and hence the researchers are using heuristic, meta-heuristic and greedy methods to solve it [2], [8]. Considering the above issues regarding allocation of resources in cloud computing, in this paper, a modified simulated annealing algorithm has been used to determine the winner of the auction in the allocation of cloud resources. The modified simulated annealing algorithm in each step, instead of using a neighboring solution, uses several neighboring solutions. This change in the base simulated annealing algorithm leads to early convergence and improves the final solution found. In the rest of this paper, in section 2, related works about economic models for resource allocation in cloud computing are introduced. In section 3, formal definition of the problem is provided. The proposed mechanism is outlined in section 4 and the simulation and experimental results are evaluated in section 5. Finally, section 6 concludes the paper.

2. Related Works

Economic models provide different policies and tools for allocating resources in cloud systems. In cloud computing, users compete with other users as well as

Manuscript received August 5, 2025
Manuscript revised August 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.8.12

resource owners with other resource owners [9], [10]. Economic models can be based on the transaction or payment of the resource price. In cloud computing, providers and owners of resources with financial incentives provide their users with cloud resources. Taking into account the points mentioned, the use of decentralized methods is a good way to manage resources in cloud computing. Economic solutions are appropriate because they have a decentralized structure and also motivate the owners of the resources to participate in their resources in the cloud [11], [12]. Another economic model is that both the user's goals and the objectives of the owners of the resource are taken into consideration in the process of resource allocation.

To date, several market-based resource allocation models and algorithms have been proposed for cloud computing environments. In the rest of this section some of them have been discussed. Wang et al. [13] conducted a study for resource allocation using an English combinatorial auction in cloud computing environments. The resource marketing price was resolved by an English combinatorial auction model, which is concentrated principally on maximizing the seller's profit and reducing the execution time for the winner determination. In 2013 [14], a virtualized resource allocation mechanism to assign CPU resources in virtualized machines was proposed. This work tried to overcome the unfairness issue of resource allocation in cloud computing. This work is essentially concentrated on enhancing the system resource utilization, and it is not considered to be of benefit to the user and service provider.

Another deficiency of this model is that it was restricted to virtual machine and different types of resources was not considered.

Xu presented CDA-CCRA, a new cloud computing resource allocation model based on combinatorial double auction mechanism for more effective resource utilization in cloud computing [15]. The CDA-CCRA model can simultaneously satisfy the users and providers requirements and significantly reduce transactions. Sabzevari et. Al have been proposed one double combinatorial auction based resource allocation approach for cloud computing environments [2]. The main goal of their study is to allocate economic resources in a way that lead to increase social welfare. Their proposed approach uses imperialist competitive algorithm for winner determination and genetic algorithm for resource allocation and payment schemes.

3. Problem Definition

Auctions that bidders can offer a combination of resources has recently been taken into consideration. Compared to a non-trading auction, the combinatorial suction has a high performance. In the cloud computing environment distributed resources, including computing resources, storage resources, network bandwidth, and so on, compete with each other to execute user's work, and as a result, a combinatorial auction is appropriate for allocating resources. In double auction, both buyer and seller can submit their offers. Compared to a one-way auction in which several buyers compete for goods sold by a vendor, a double auction prevents of monopolies. The combinatorial double auction offers not only the benefits of a combinatorial auction, but also the needs of both buyers and sellers and therefore is suitable for cloud resource allocation. The purpose of the combinatorial double auction is to maximize the overall profit by taking into account this limitation that the number of units selected from the resources in the buyer's combined packages does not exceed the number of units provided by the vendors.

Suppose there is a R resource set containing k resources. After both parties offer their offers to the broker, the broker must perform the auction which is known as the winner determination of the auction problem. This problem is described in formulas 1, 2 and 3.

$$\text{Max}\sum_{j=1}^{n} P_j x_j \tag{1}$$

$$\sum a_{ij} x_j \leq 0 \quad ,\forall i \in K \tag{2}$$

$$x_j \in \{0,1\}, \quad \forall j \in \{1,2,...,n\} \tag{3}$$

The set of proposed packages is $\overline{B} = \{B_1, B_2, ..., B_j, ..., B_n\}$, where n is the number of combined packets of resources. Each bid B_j is (a_j, p_j), where $a_j = (a_{1j}, ..., a_{ij}, ..., a_{kj})$ and a_{ij} represents the number of units requested from resource i. Also, p_j is the offered price for package j. If $p_j > 0$, it is a buyer's offer, and if $p_j < 0$, then it is considered as the seller's offer. Also, if $x_j = 1$, that is, the packet is assigned and if $x_j = 0$, that is, not assigned. Finding the best x_j values to maximize formula (1) with the constraints (2) and (3) is the same as determining the winner of the auction, which is considered as programming 0-1 and is an NP-hard problem.

4. Proposed Approach

In this section, the various stages of the proposed approach, which uses the enhanced simulated annealing algorithm to determine the winner of the auction in the allocation of cloud computing resources, is presented.

4.1 Encoding

An important step in evolutionary algorithms is how to encode and display a solution. Each problem solution in the proposed approach is an array of binary numbers of length *n*, in which *n* is the number of offers. Array members as mentioned, there are binary numbers in which 1 means acceptance of the offer and 0 means the denial of the relevant offer. Figure 1 illustrates an example of a solution for eight offers.

0	1	1	0	1	0	1	1

Fig1: An example of an offer (n=8) and the corresponding solution

4.2 Fitness Function

Fitness function is one of the important concepts in every heuristic and meta-heuristic approaches. In determining the winner of the auction, the suitability of a solution is obtained from the sum of the winning bidder's bid. The fitness function is shown in equation 4.

$$\sum_{j=1}^{n} \mathrm{Price}_j x_j \qquad \text{where } x_j \in \{0,1\} \tag{4}$$

4.3 Simulated Annealing Different Stages

Simulated annealing algorithm is a random search method based on the Monte Carlo repeat strategy [16]. That is, it moves by removing steps of a specified length from a point in the solution space to another point. This algorithm is modeled on the process of gradually cooling the molten metals to obtain the highest molecular strength in their crystal. The Simulated annealing algorithm begins its search process with a random point in the space of the bottoms, and each time the temperature drops, it moves from point to point, and if it finds a better solution, replaces the current solution. Like all other meta-algorithms, simulated refrigeration also has its own special approach for passing the optimal locale, which is to accept a bad solution with a certain probability. This suggests that at each step, the reduction of temperature, if the solution obtained is worse than the current solution, takes it with a definite probability, thus allowing the search for unknown resolution spaces and, finally, the provision of a stopping condition provides the best solution.

The change that has been made in the base simulated annealing algorithm is that the algorithm finds several new solutions at each iteration, instead of finding one new solution and switches to the best of them. This change can increase the convergence rate and also increase the suitability of the final solution found by the algorithm. The results of the experiments that will be presented in the next chapter are evidence of this. Considering the above mentioned discussion, the proposed simulated annealing algorithm steps can be summarized as follow:

1. Generate a random solution and compute its fitness value
2. Specify initial temperature value
3. Generate several neighbor solutions using mutation operator
4. If the best of neighbor solutions is better than current solution, replace the current solution with it else replace with a specified probability
5. Decrease the temperature
6. If the termination condition is satisfied then stop else go to step 3

4.3.1 Temperature Initialization and Reduction

Initialization of the temperature as well as its reduction in each repetition of the simulated annealing algorithm are crucial steps in this algorithm [17], [18]. The initial value of the temperature is considered as T_{in} and the equation 5 is used to reduce it.

$$T_{p+1} = \alpha T_p \tag{5}$$

In equation 5, α is the coefficient of temperature reduction which is a real value in the interval (0,1). If the value is close to 1, it causes a slow decrease in temperature, and thus allows the algorithm that searches for a large space of solutions, and accepts many of the displacements to the optimal solution. After several experiments and taking into account the above-mentioned cases, the value of α is considered equal to 0.994 in the proposed approach.

4.3.2 Neighbor Solution Generation

Generation of neighboring solutions in the simulated annealing algorithm is generally carried out using one or more well-known mutation operators. Generation of neighbor solutions in the proposed simulated annealing algorithm is obtained by random selection of one element of the current solution and complementing it. The pseudocode of the proposed approach is shown in Figure 2.

```
Sol1= RandomSolution()
```

```
BestSol = Sol
Repeat the following max_rep times
  f1 = Objective(Sol)
  Repeat the following nb_neighbors times
   NewSol = Neighbor(Sol)
   f2 = Objective(NewSol)
   Diff = f2-f1
   if (f2<f1 or Rand() <= exp(-Diff/Temperature))
      Sol = NewSol
      f1 = f2
      if ( f1 < Objective(BestSol))
        BestSol = Sol
  Temperature = alpha * Temperature
     Final Result=BestSol
```

Fig2: Pseudo code of proposed simulated annealing algorithm

5. Simulation and Experimental Results

To conduct all experiments, a Dell computer with a Core i5, 2 GHz processor and 4 GB of main memory was used. All experiments and simulations were also carried out in the Matlab 2015b software environment. Considering that the proposed approach uses a meta-heuristic algorithm, the results of the convergence as well as stability evaluation are presented. Also, the results of the proposed approach, called the MSA[1] later this section, have been compared with the results of Genetic Algorithm (GA) and Simulated Annealing (SA). Tables 1, 2 and 3 show, respectively, the parameters related to the GA, PSO and MSA.

Table 1. GA parameters

Parameter	Value
Initial population	50, 100
Selection operator	Roulete wheel
Crossover	Two point
Crossover rate	0.85
Mutation rate	0.04

Table 2. SA parameters

Parameter	Value
Initial population	50, 100
Number of neighbors	1
α	0.994

Table 3. MSA parameters

Parameter	Value
Initial population	50, 100
Number of neighbors	5
α	0.994

5.1 Convergence Experiment

To test convergence, MSA, SA and GA algorithms are executed for three different scenarios, and Figures 3, 4 and 5 show the convergence to the final solution in three scenarios, respectively. In the first scenario, the number of participants is considered 100, in the second scenario the number of participants is considered 200 and in the third scenario, the number of participants is considered 500. In the graphs, the horizontal axis shows the order of repetition of the algorithm and the vertical axis also shows the value of the fitness function.

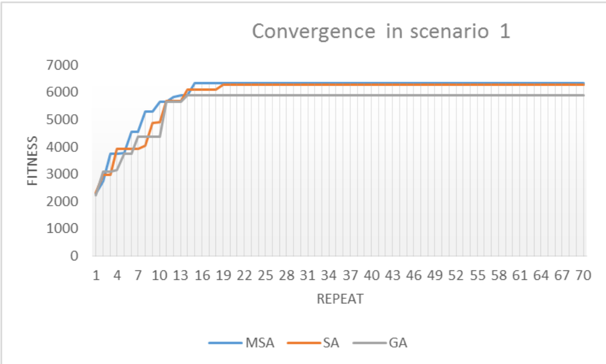

Fig3: Convergence in scenario 1

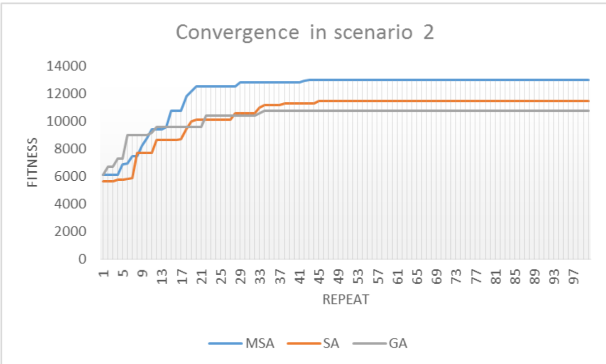

Fig4: Convergence in scenario 2

[1] Modified Simulated Annealing

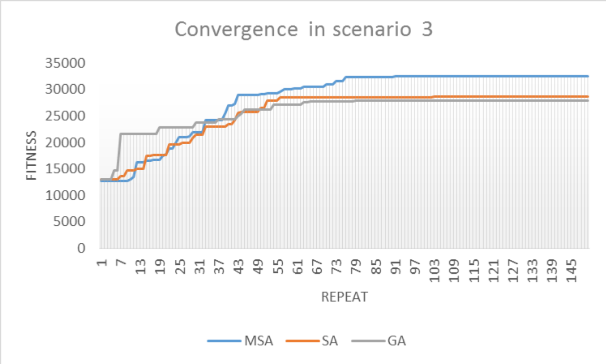

Fig5: Convergence in scenario 3

The results show that the proposed approach for determining the winner of the double auction in allocating resources in cloud computing has a good convergence rate and also found better solutions compared to SA and GA.

5.2 Stability Experiment

Meta-heuristic algorithms have an indeterminate and random nature, so it is necessary to examine the stability of these algorithms. The stability of an algorithm is whether the algorithm generates the same or proximity results for different executions. To investigate the stability of the MSA algorithm for the three scenarios mentioned above, the algorithm is executed 20 times and the value of the fitness function in each run is shown in Figures 6, 7 and 8. The horizontal axis in the diagrams shows the execution number of the algorithm and the vertical axis shows the fitness value.

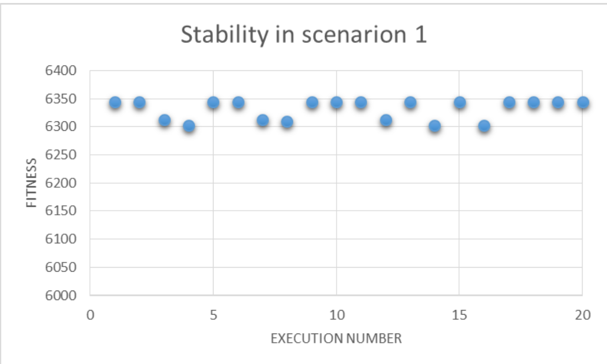

Fig6: Stability in scenario 1

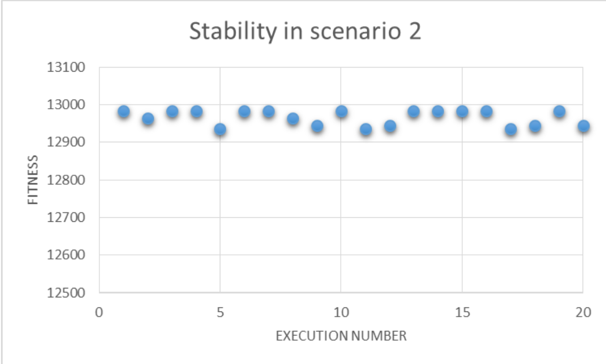

Fig7: Stability in scenario 2

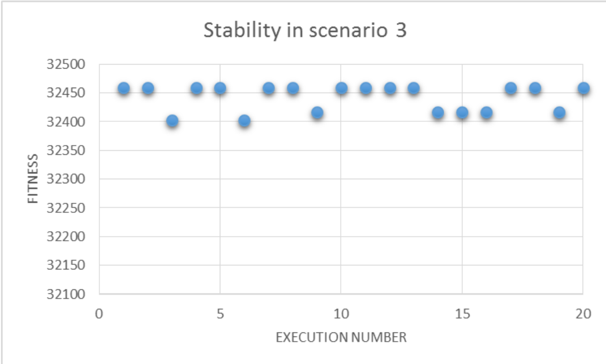

Fig8: Stability in scenario 3

Evaluating the results of the stability test shows good stability of the proposed approach. Also, this experiment shows that the proposed approach converges to optimal solution at all times of the execution of the algorithm.

6. Conclusion

Resource management is one of the key challenges in cloud computing and cloud data center management. Most cloud providers use fixed price mechanisms to allocate resources to users. But these mechanisms do not provide an efficient and acceptable allocation of resources, and in fact can not maximize the profit of cloud providers. In such a situation, cloud-based economic models are suited to tune, deliver, and demand resources.

In this paper, combinatorial double auction method has been used to allocate resources in cloud computing. Considering that the problem of determining the winner of the auctions is in the category of NP-hard problems, in this paper the modified simulated annealing algorithm has been used to solve it. The results of the experiments conducted in the MATLAB environment showed that the proposed approach has a good convergence rate and is also very well in terms of sustainability. Also, the results of experiments using three different scenarios showed that the proposed approach generates more suitable solutions than two genetic and simulated annealing algorithms.

References

[1] A. M. Sampaio and J. G. Barbosa, "Chapter Three - Energy-Efficient and SLA-Based Resource Management in Cloud Data Centers," in *Energy Efficiency in Data Centers and Clouds*, vol. 100, A. R. Hurson and H. Sarbazi-Azad, Eds. Elsevier, 2016, pp. 103–159.

[2] R. A. Sabzevari and E. B. Nejad, "Double Combinatorial Auction based Resource Allocation in Cloud Computing by Combinational using of ICA and Genetic Algorithms," *Int. J. Comput. Appl.*, vol. 110, no. 12, 2015.

[3] H. Wang, H. Tianfield, and Q. Mair, "Auction Based Resource Allocation in Cloud Computing," *Multiagent Grid Syst.*, vol. 10, no. 1, pp. 51–66, 2014.

[4] X. Deng, P. Goldberg, B. Tang, and J. Zhang, "Revenue maximization in a Bayesian double auction market," *Theor. Comput. Sci.*, vol. 539, pp. 1–12, 2014.

[5] I. Fujiwara, K. Aida, and I. Ono, "Applying double-sided combinational auctions to resource allocation in cloud computing," in *Proceedings - 2010 10th Annual International Symposium on Applications and the Internet, SAINT 2010*, 2010, pp. 7–14.

[6] F. Nassiri-Mofakham, M. A. Nematbakhsh, A. Baraani-Dastjerdi, N. Ghasem-Aghaee, and R. Kowalczyk, "Bidding strategy for agents in multi-attribute combinatorial double auction," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3268–3295, 2015.

[7] Y. LIU, Y. LIU, X. MA, and K. LIU, "Solving {WDP} in combinatorial double auction based on trading strategy," *J. China Univ. Posts Telecommun.*, vol. 19, Supple, pp. 148–152, 2012.

[8] F. Gorbanzadeh and A. A. Pourhaji Kazem, "Hybrid Genetic Algorithms for Solving Winner Determination Problem in Combinatorial Double Auction in Grid," *IAES Int. J. Artif. Intell.*, vol. 1, no. 2, 2012.

[9] M. Mihailescu and Y. M. Teo, "Dynamic resource pricing on federated clouds," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, 2010, pp. 513–517.

[10] J. Stößer, D. Neumann, and C. Weinhardt, "Market-based pricing in grids: On strategic manipulation and computational cost," *Eur. J. Oper. Res.*, vol. 203, no. 2, pp. 464–475, 2010.

[11] A. Haque, S. M. Alhashmi, and R. Parthiban, "A survey of economic models in grid computing," *Futur. Gener. Comput. Syst.*, vol. 27, no. 8, pp. 1056–1069, 2011.

[12] L. Rodero-Merino, E. Caron, A. Muresan, and F. Desprez, "Using clouds to scale grid resources: An economic model," *Futur. Gener. Comput. Syst.*, vol. 28, no. 4, pp. 633–646, 2012.

[13] X. w. Wang, X. y. Wang, and M. Huang, "A resource allocation method based on the limited English combinatorial auction under cloud computing environment," in *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, 2012, pp. 905–909.

[14] C. Jiang, L. Duan, C. Liu, J. Wan, and L. Zhou, "VRAA: virtualized resource auction and allocation based on incentive and penalty," *Cluster Comput.*, vol. 16, no. 4, pp. 639–650, 2013.

[15] J. Xu, "A Cloud Computing Resource Allocation Model Based on Combinatorial Double Auction," in *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, 2016, pp. 5–8.

[16] S. P. Brooks and B. J. T. Morgan, "Optimization using simulated annealing," *Stat.*, pp. 241–257, 1995.

[17] A. A. Pourhaji Kazem, A. M. Rahmani, and H. Habibi Aghdam, "A modified simulated annealing algorithm for static task scheduling in grid computing," in *Computer Science and Information Technology, 2008. ICCSIT'08. International Conference on*, 2008, pp. 623–627.

[18] A. A. Pourhaji Kazem, N. Dadashkhani, M. Kargahi, and H. Habibi Aghdam, "A Simulated Annealing Approach for Maximizing the Accrued Utility of an Isochronal Soft Real-Time System," in *Environmental and Computer Science, 2009. ICECS'09. Second International Conference on*, 2009, pp. 191–195.