

Hybrid Genetic Algorithms for Solving Winner Determination Problem in Combinatorial Double Auction in Grid

Farhad Gorbantzadeh*, Ali Asghar Pourhaji Kazem**

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

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

Article Info

Article history:

Received May 16, 2012

Revised May 30, 2012

Accepted Jun 10, 2012

Keyword:

Grid

Resource allocation

Auction

WDP

ABSTRACT

Nowadays, since grid has been turned to commercialization, using economic methods such as auction methods are appropriate for resource allocation because of their decentralized nature. Combinatorial double auction has emerged as a major model in the economy and is a good approach for resource allocation in which participants of grid, give their requests once to the combination of resources instead of giving them to different resources multiple times. One problem with the combinatorial double auction is the efficient allocation of resources to derive the maximum benefit. This problem is known as winner determination problem (WDP) and is an NP-hard problem. So far, many methods have been proposed to solve this problem and genetic algorithm is one of the best ones. In this paper, two types of hybrid genetic algorithms were presented to improve the efficiency of genetic algorithm for solving the winner determination problem. The results showed that the proposed algorithms had good efficiency and led to better answers.

Copyright © 2012 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Farhad Gorbantzadeh,
Departement of Computer Engineering,
Islamic Azad University of Tabriz,
Pasdaran highway, Tabriz, Iran, Islamic Azad University.
Email: f.gorbantzadeh@iaut.ac.ir

1. INTRODUCTION

Grid is a rapidly developing computing structure that allows for the components of the information technology infrastructure, computational capabilities, databases, sensors, and people to be shared flexibly as true collaborative tools [1]. It enables virtual organizations and enterprises to share, exchange, select, and aggregate geographically distributed heterogeneous resources. One important problem in such environments is the efficient allocation of resources.

Over the past years, economic approaches to resource allocation have been developed [2] and one of the best economic approaches is auction model. In the auction model, each provider and consumer acts independently and they agree privately on the selling price. Auctions are used for the products that have no standard values and the prices are affected by the supply and demand at a specific time. Auctions require little global price information and are decentralized and easy to implement in a grid setting.

Combinatorial auction, as a new auction model, has satisfying characteristics in grid. In the combinatorial auction, participants can place bids on combinations of discrete items or "packages" rather than just individual items or continuous quantities. This can improve efficiency while maximizing revenue in the grid. However, the existing combinatorial auction-based resource allocation [3,4] usually focuses only on the users' side and does not take providers' price requirements into consideration. To gain better performance, the double auction is proposed.

The double auction model has a high potential for grid computing [5]. In a double auction model, consumers and providers submit their requests at any time. If there are requests that match or are compatible with a price at any time, then a trade is executed immediately. Double auctions, in which both sides submit demand or supply bids, are considerably more efficient than several combined one-sided auctions. Moreover, compared with the one-side auction, where multiple buyers compete for the commodities sold by one seller or multiple sellers compete for the right to sell to one buyer, the double auction can prevent monopoly or monopsony.

The combinatorial double auction [6] not only has the advantages of the combinatorial auction but also considers the requirements of both buyers and sellers and is more suitable for grid resource allocation. The objective of the combinatorial double auction is to maximize the total trade surplus while satisfying the constraint that the number of units selected by buyer bundles does not exceed the number provided by the selected seller bundles for each item. This is denoted as winner determination problem (WDP) which is an NP-hard problem, on which considerable studies have emerged recently.

One of the best approaches for WDP problem is genetic algorithm. In many problems, because of stochastic characteristic of GA's operators (crossover, mutation), GA could not find optimal solutions and may have a tendency to converge towards the points near the global optimum. So, in this paper, two hybrid genetic algorithms were proposed to get better solutions for solving winner determination problem. GA was combined with two local search algorithms, hill-climbing and simulated-annealing. The results showed that hybrid GAs have better performance rather than GA.

The remainder of the paper is organized as follows. In Section 2, the works related to WDP are discussed. In Section 3, WDP is explained in details. In Section 4, the proposed methods were explained. Section 5 reports simulation and experimental results. Finally, Section 6 concludes the work.

2. RELATED WORKS

Winner determination problem is an NP-hard problem which was first studied in [7]. Thus far, most researches have focused on developing heuristics, studying the complexity of the problem and applying some integer programming techniques.

In [8], two randomized methods were proposed. The first was based on the Cross-Entropy (CE) method and the other was a new adaptive simulation approach by Botev and Kroese, which evolved from the CE method and combined the adaptability and level-crossing ideas of CE with Markov Chain Monte Carlo techniques. In [9], three heuristic bid ordering schemes were presented for solving WDP; the first two schemes took into account the number of goods shared by conflicting bids and the third one was based on a recursive application of such local heuristic functions. In [10], a new class of parallel branch-and-bound (B&B) schemes was proposed which focused on the functional parallelism instead of conventional data parallelism to support such a heterogeneous and irregular parallelism using a collection of autonomous agents distributed over the network.

A hill-climbing greedy algorithm and an SA-like random search algorithm and their enhancements for searching multiple key parameter values were proposed in [11]. In [12], an efficient approximate searching algorithm IAA was proposed for the problem, which used the Ant Colony Optimization algorithm based on heuristic rules; the proposed algorithm not only gave the way for identifying feasible bids with a given partial solution but also avoided the unnecessary trials that would not lead to an optimal solution. In [13], the authors considered the set packing formulation of the problem, studied its polyhedral structure and then proposed a new and tighter formulation and presented new valid inequalities which were generated by exploiting peculiarities of combinatorial auctions and implemented a branch-and-cut algorithm demonstrating its efficiency in a big number of instances. A differential evolution algorithm (DE) was also studied in [14].

Genetic algorithm (GA) is one of the best methods for solving WDP and has many different types. A simple GA was presented in [15]. In [16], the authors focused on optimal winner determination in combinatorial auctions with XOR-bids and OR-bids and proposed a partheno-genetic algorithm, with partheno-genetic operators and the fitting-first heuristic rules. The lower-layer Orthogonal Multi-Agent Genetic Algorithm (OMAGA) was applied for searching the optimal solution for the given combinatorial auctions optimization problem in [17]. In [18], the use of sub-populations (Parallel genetic algorithms PGA), and a hybridization of a PGA with SLS (stochastic local search) which can be implemented on a parallel architecture were considered. An improved hybrid ant genetic algorithm was adopted to solve the problem in [19].

3. PROBLEM DEFINITION

The central problem arising from combinatorial auctions is winner determination, which is described as follows. Suppose an auctioneer has a collection of items to auction to a number of bidders, who submit bids on every combination of items (bundles). Given the set of bids, the auctioneer then determines the allocation of items to bidders that maximizes their revenue under the constraint that the number of units selected by buyer bundles does not exceed the number provided by the selected seller bundles for each item. This problem can be formally stated as a combinatorial optimization problem in the following way:

Suppose there is an item set K , in which there are k items. The model is as follows:

$$\max \sum_{j=1}^n p_j x_j \quad (1)$$

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

$$x_j \in \{0,1\}, \quad \forall j \in \{1,2, \dots, n\} \quad (3)$$

The set of bid bundles is $\bar{B} = \{B_1, B_2, \dots, B_j, \dots, B_n\}$ in which there are n bundles. A bid B_j can be specified as (a_j, p_j) , where $a_j = (a_{1j}, \dots, a_{ij}, \dots, a_{kj})$, and a_{ij} is the units of item i requested (when $a_{ij} > 0$) or supplied (when $a_{ij} < 0$) by bundle j . p_j is the amount that the bidder is willing to pay for bundle j : if $p_j > 0$, it is regarded as a buyer bid; otherwise, it is regarded as a seller bid. if $x_j = 1$ it means that the bundle j wins and $x_j = 0$ means that bundle j does not win. Finding an optimal case of x_j s for maximizing the revenue Eq. (1) with the restrictions Eq. (2), Eq. (3), is the winner determination problem and it can be seen that the model can be solved as the 0-1 programming problem and is an NP-hard problem.

4. PROPOSED ALGORITHMS

Genetic algorithm is one of the best methods for solving WDP [17]. But, as we know [20], while GA is good at rapidly identifying good areas of the search space (exploration), it is less good at the endgame of fine-tuning solutions (exploitation), partly as a result of the stochastic nature of the variation operators. A more efficient method is to incorporate a more systematic search of the vicinity of good solutions by adding a local search improvement step to the evolutionary cycle. So, in this paper, GA was combined with simulated-annealing and hill-climbing local searches in order to solve winner determination problem and obtain more revenue.

4.1 Main components of genetic algorithm

The main components of the proposed algorithms adapted to the winner determination problem are given in the following.

A. Individual Representation

An individual is represented by a binary vector A having a length due to the number of bids n . The components of the vector are 0 or 1. Here 1 denotes that bid is accepted and 0 denotes that bid is not accepted. An instance of individual for 8 bids is shown in Fig 1.

0	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

Figure 1. Individual representation

B. Fitness Function

Fitness function is one of the most important concepts in the proposed algorithms. The quality of an individual is given by the sum of the price of winning bids as shown in Eq. (4).

$$\sum_{j=1}^n price_j x_j \quad \text{where } x_j \in \{0,1\} \quad (4)$$

C. Parents Selection

The selection operator is used to select the two parents for crossover operation. Selection operator is tournament selection in which, each parent is the best one of a random set. Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. Selection pressure is easily adjusted by changing the tournament size. If the tournament size is larger, weak individuals have smaller chance of being selected.

D. Crossover

Crossover, the process whereby a new individual solution is created from the information contained with two parent solutions, is considered as one of the most important features in genetic algorithms. Uniform crossover was used here in which, for each gene, a random value between 0 and 1 is generated. If the random value is less than 0.5, the gene is inherited from the first parent; otherwise, from the second one. The second offspring is created using the inverse mapping. This is illustrated in Fig. 2.

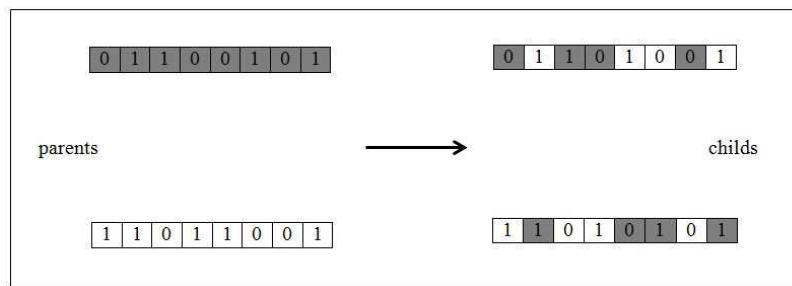


Figure 2. Uniform Crossover

E. Mutation

Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence, GA can come to better solution using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set high, the search will turn into a primitive random search. The mutation operator used in this paper is bit flip. This mutation operator takes the chosen genome and inverts the bits. (i.e. if the genome bit is 1, it is changed to 0 and vice versa).

F. Replacement

After the offspring is made, a choice has to be made on which individuals will be allowed in the next generation. In the proposed algorithm, the worst solution was replaced with the best solution of the previous generation.

4.2 Hybrid genetic algorithm with Hill-Climbing

The solutions which do not satisfy the Eq. (2), mean that the number of units selected by buyer bundles is more than the number provided by the selected seller bundles for each item; the solutions whose fitness value is smaller than zero means they do not have any revenue and are called infeasible solutions. In the proposed algorithm, after applying crossover and mutation operators, if the new solution is an infeasible solution, it passes to a hill-climbing function. Hill-climbing (HC) is a mathematical optimization technique which belongs to the family of local search. It begins with one initial solution (here, with infeasible solution), then the solution is mutated and, if the mutation result has higher fitness for the new solution than for the previous one, the new solution is kept; otherwise, the current solution is retained. Here, the hill-climbing function starts with an infeasible solution and continues until a feasible solution is obtained; then it returns the feasible solution to GA. The algorithm is as follows:

```

1. create an initial population randomly
2. for i from 1 to generation number
3.   for j from 1 to population size
4.     select parents
5.     createnew_solution with crossover and mutation operators
6.     ifnew_solution is infeasible
7.       new_solution = Hill-Climbing(new_solution)
8.     endIF
9.   endFOR
10.  create next population
11.  if stop condition is met
12.    stop the algorithm
13.  endIF
14. endFOR

15. Function Hill-Climbing (current_solution)
16.  whilecurrent_solution is infeasible do
17.    next_solution = expand current_solution
18.    if fitness of next_solution > fitness of current_solution
19.      current_solution = next_solution
20.    endIF
21.  endWHILE
22.  returncurrent_solution
23. endFUNC

```

Figure 3. GA with hill-climbing

```

1. Initialize the variables of GA and SA
2. create an initial population randomly
3. for i from 1 to generation numeber
4.   for j from 1 to population size
5.     select parents
6.     createnew_solutions with applying crossover and mutation on parents
7.      $\Delta f = \text{fitness}(\text{parents}) - \text{fitness}(\text{new\_solutions})$ 
8.     if  $\Delta f < 0$ 
9.       new_solutions accept to new generation
10.    else
11.      if  $\exp(\Delta f / T) > \text{rand}(0 \sim 1)$ 
12.        new_solutions accept to new generation
13.      else
14.        parents go to new generation
15.      endIF
16.    endIF
17.  endFOR
18.  decrease T
19.  if the stop condtions are satisfied stop the algorithm
20. endFOR

```

Figure 4. GA with simulated- annealing

4.3 Hybrid genetic algorithm with Simulated-Annealing

Simulated Annealing (SA) is a Meta heuristic which has been successfully applied for solving a variety of difficult optimization problems. The term annealing refers to the process of cooling after heating, in order to make the material tough and temper. Each step of the SA algorithm attempts to replace the current solution by a random solution (chosen according to a *candidate distribution*, often constructed to sample from the solutions near the current solution). The new solution may then be accepted with a probability that depends both on the difference between the corresponding function values and on a global parameter T

(called the *temperature*), that is gradually decreased during the process. Eq. 4 shows this probability in which f and f^* are the fitness values of the current and new solutions respectively.

$$P = \begin{cases} 1 & \text{if } f^* \geq f \\ \exp\left(\frac{f-f^*}{T}\right) & \text{if } f^* < f \end{cases} \quad (5)$$

The dependency is such that the choice between the previous and current solutions is almost random when T is large; but it increasingly selects the better or "uphill" solution as T goes to zero. The allowance for "downhill" moves potentially and saves the method from becoming stuck at local optima. In the proposed algorithm, after selecting the parents and applying the crossover and mutation operators, the new solutions were accepted to the next generation according to Eq. (5); otherwise, they were not accepted and the parents passed to the next generation without any changes. The parameter T decreased in every generation of GA. The proposed algorithm is as Figure 4.

5. SIMULATION AND EXPERIMENTAL RESULTS

The proposed approaches were implemented in MATLAB environment and were executed by the personal computer with a dual core processor and a 4 GB RAM in order to solve the winner determination problem of combinatorial auctions. Table 1 shows the parameters of the proposed methods.

Table 1. Simulation Parameters

Parameters	Value
Population number	200
Chromosome type	Binary array
Chromosome size	Same as number of bids
Parent selection	Tournament selection
Crossover	Uniform crossover (rate:0.95)
Mutation	Bit flip (rate 0.06)
Stop condition	no improvement in fitness of best solution over 20 generations
Temperature T	90 $T = 0.9 \times T$

Three resources (A, B, C) were supposed. Every user and seller determined the number of units of each resource and proposed its total price for that bundle. The units of demand or supply of each user and seller and the price for the resource combination can be seen in Table 2. There are 16 bidders (participants) in this table, six of them are users and ten are sellers. For each item of the resource, a reference resource can be chosen as the unit. Take storage resource for example, a unit can be 1 GB storage space.

Table 2. The Parameters of Each Participant

NO	A	B	C	Price	NO	A	B	C	Price
1	0	3	3	104	9	-2	-3	-1	-80
2	4	3	3	136	10	-1	-3	0	-36
3	3	3	4	144	11	-2	-2	0	-38
4	2	0	1	33	12	0	-1	-3	-70
5	4	3	2	125	13	-3	-3	-1	-93
6	1	5	1	93	14	-2	0	-3	-71
7	0	-2	-2	-59	15	0	-3	-3	-76
8	-3	-2	-3	-110	16	-3	-1	-1	-54

By solving the combinatorial double auction represented in Eq. (1), it can be seen that the participants 1, 2, 3, 5, 6, and 7, 8, 10, 11, 12, 13, 15, 16 were the winning bidders. The bids of 4, 9, and 14 were rejected.

To evaluate the proposed hybrid genetic algorithms, first each algorithm was executed with 400 bidders in order to see how the algorithms converge to the answers. Fig. 5 shows the convergence of the best

solution fitness value over subsequent generations. It can be seen that when GA stopped near the best answers, hybrid GAs found better answers.

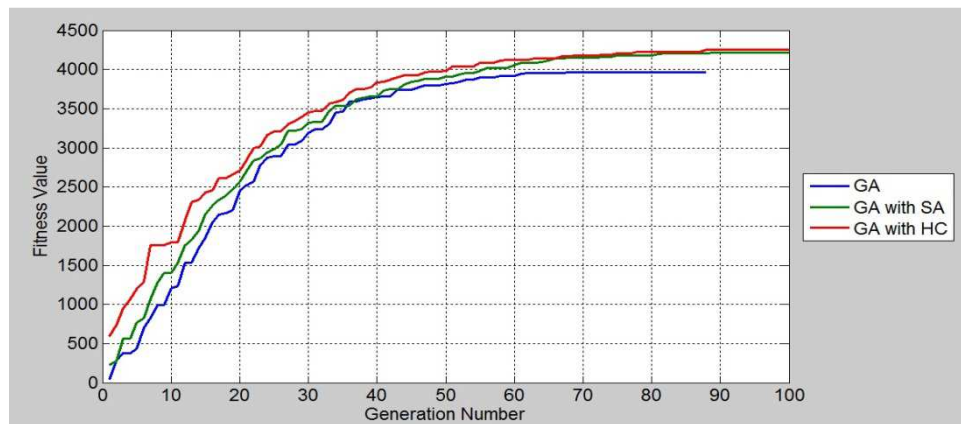


Figure 5. Convergence of best solution's fitness value over subsequent generations



Figure 6. Stability of genetic algorithm in 10 times execution

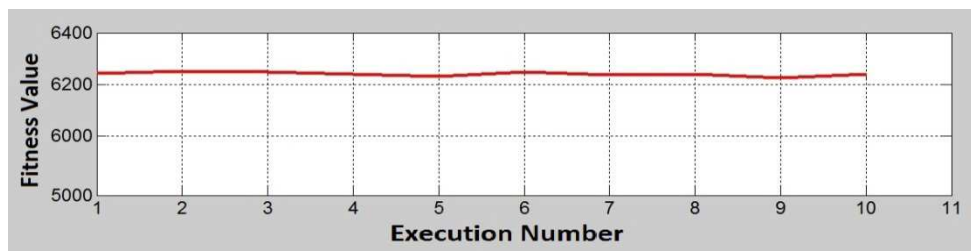


Figure 7. Stability of genetic algorithm with SA in 10 times execution



Figure 8. Stability of genetic algorithm with HC in 10 times execution

In the second experimental test, the stability of the proposed algorithms was compared with that of the genetic algorithm. For this purpose, each algorithm was executed 10 times for the same 400 bidders. As shown in Fig. 6, Fig 7 and Fig 8, the difference between the answers in the proposed methods was less than the difference between the answers in GA and they had better stability compared with the GA.

In the third experimental test, the optimization results and computation time of the proposed approaches were compared with those of the GA. Therefore, each method was executed 10 times for different numbers of bids (100,200,...,1000) and then the mean values of their fitness and execution time were calculated. Fig. 9 shows the average fitness of different numbers of bids in 10 times of execution. It is seen that the proposed methods found better answers, especially when the number of participants increased.

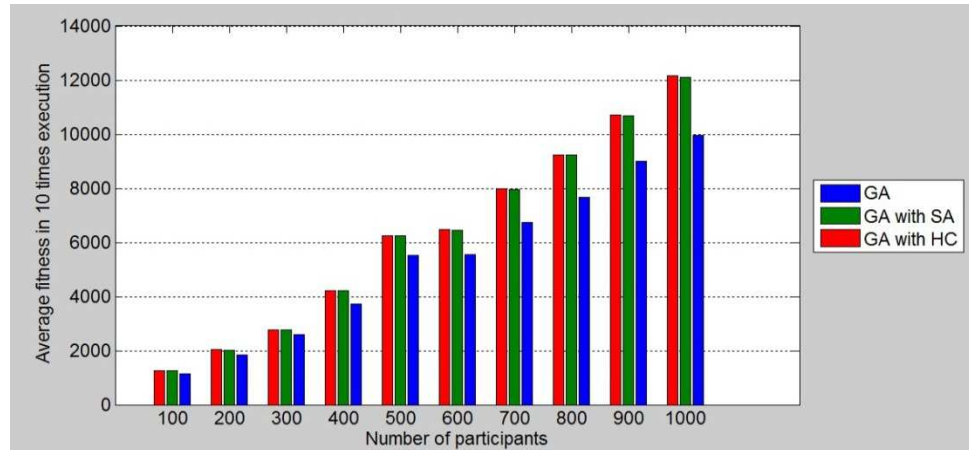


Figure 9. Average fitness in 10 times execution of each method

Table 3 shows average execution time in 10 times of executing each method. The values are in seconds. In proposed methods, because a new step was added to the genetic algorithm, so execution times of the proposed methods were a little longer than those of GA. But the purpose of winner determination problem is to reach the maximum benefit and because grid is a competitive environment, so the increment in execution time is negligible.

Table 3. Average execution time in 10 times execution of each method

Participants number	100	200	300	400	500	600	700	800	900	1000
GA	9	30	56	76	101	118	140	164	189	203
GA with SA	25	79	137	195	245	307	336	391	475	484
GA with HC	40	93	145	215	255	315	345	411	496	501

6. CONCLUSION

Winner determination problem in combinatorial double auction is an NP-hard problem. In this paper, two hybrid genetic algorithms were proposed for solving this problem. Since genetic algorithm was not good at the end of finding good solutions, in this paper hill-climbing and simulated annealing local searches were added to GA. The proposed methods were tested with different instances of participants. Simulation results showed that hybrid GAs had better efficiency with acceptable time execution, especially when the number of participants increased.

REFERENCES

- [1] Foster, C. Kesselman (Eds.), "The Grid: Blueprint for a Future Computing Infrastructure," Morgan Kaufmann Publishers, San Francisco, USA, 1999.
- [2] D. F. Ferguson, C. Nikolaou, J. Sairamesh, and Y. Yemini, "Economic models for allocating resources in computer systems. In Market-Based Control," A Paradigm for Distributed Resources Allocation. World Scientific, 1996.
- [3] Schwind M, Gujo O, Stockheim T, "Dynamic resource prices in combinatorial grid system," Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, Jun 26-29, 2006, San Francisco, CA, USA. Piscataway, NJ, USA: IEEE, 2006: 49-54

- [4] Das A, Grosu D. , "Combinatorial auction-based protocols for resource allocation in grids," Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium(IPDPS'05), Apr 4-8, 2005, Denver, CO, USA. Piscataway, NJ, USA: IEEE, 2005: 23-30
- [5] R. Buyya, D. Abramson, J. Giddy, H. Stockinger, "Economic models for resource management and scheduling in Grid computing," The Journal of Concurrency and Computation 14 (13_15) (2002) 1507_1542.
- [6] Xia M, Stallaert J, Whinston B A. , "Solving the combinatorial double auction problem," European Journal of Operational Research, 2005, 164(1): 239-251
- [7] M. Rothkopf, A. Pekec, R. Harstad, "Computationally manageable combinatorial auctions," Management Science 44 (1998) 1131-1147.
- [8] Joshua C. C. Chan, Dirk P. Kroese, "Randomized methods for solving the winner determination problem in combinatorial auctions," Proceedings of the 2008 Winter Simulation Conference
- [9] M. Mito, S. Fujita, "On Heuristics for Solving Winner Determination Problem in Combinatorial Auctions," Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03)
- [10] S. Fujita, S. Tagashira, C. Qiao. M. Mito, "Distributed Branch-and-Bound Scheme for Solving the Winner Determination Problem in Combinatorial Auctions," Proceedings of the 19th International Conference on Advanced Information Networking and Applications 2005
- [11] N. Fukuta, T. Ito, "Towards Better Approximation of Winner Determination for Combinatorial Auctions with Large Number of Bids," Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology 2006
- [12] R. Gan, Q. Guo, H. Chang, Yang Yi, "Ant Colony Optimization for Winner Determination in Combinatorial Auctions," Third International Conference on Natural Computation (ICNC 2007)
- [13] L.F. Escudero , M. Landete, A. Marin, "A branch-and-cut algorithm for the Winner Determination Problem," Decision Support Systems 46 (2009) 649-659
- [14] D. Boughaci, "A Differential Evolution Algorithm for the Winner Determination Problem in Combinatorial Auctions," Electronic Notes in Discrete Mathematics 36 (2010) 535-542
- [15] P. Patodi, A.K. Ray, M. Jenamani, "GA Based Winner Determination in Combinatorial Reverse Auction," 2011 Second International Conference on Emerging Applications of Information Technology
- [16] J. CONG BAI, WI-YOU CHANG, YANG YI, "A partheno-genetic algorithm for optimal winner determination in combinatorial auctions," Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2006
- [17] L. ZHANG, R. ZHANG, "The Winner Determination Approach of Combinatorial Auctions based on Double Layer Orthogonal Multi-Agent Genetic Algorithm," 2007 Second IEEE Conference on Industrial Electronics and Applications
- [18] Kahina Achour, louiza slaouti and Dalila Boughaci, "A MetaComputing Approach for the Winner Determination Problem in Combinatorial Auctions," 2010 IEEE
- [19] Li Chen, Shanli Hu, Xiaoyun Chen, Yunguo Lin, "Hybrid Algorithm for Winner Determination Problem in Combinatorial Auctions," 2008 International Conference on Computational Intelligence and Security
- [20] A.E.Eiben, J.E.Smith, "Introduction to evolutionary computing," Springer, Natural Computing Series 2003

BIOGRAPHIES OF AUTHORS



Farhad Gorbazadeh received his B.S. degree in Computer Engineering from Azad university of Shabestar, Iran, in 2007. He has been a M.S. student at the Azad university of Tabriz, Tabriz, Iran, since 2009. His research interests include grid computing and wireless sensor networks.



Ali Asghar Pourhaji Kazem received a B.Sc. degree in computer engineering from University of Isfahn and also a M.S. degree in computer engineering from Shahid Beheshti University in Tehran. He is currently a Ph.D. student of computer engineering in Science and Research branch of Islamic Azad University in Tehran. His current research interests include distributed systems, Grid computing and Cloud computing.