International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:8, 2008

Design and Implementation of Optimal Winner Determination Algorithm in Combinatorial e-Auctions

S. Khanpour, and A. Movaghar

Abstract—The one of best robust search technique on large scale search area is heuristic and meta heuristic approaches. Especially in issue that the exploitation of combinatorial status in the large scale search area prevents the solution of the problem via classical calculating methods, so such problems is NP-complete. in this research, the problem of winner determination in combinatorial auctions have been formulated and by assessing older heuristic functions, we solve the problem by using of genetic algorithm and would show that this new method would result in better performance in comparison to other heuristic function such as simulated annealing greedy approach.

***Keywords*—**Bids, genetic algorithm, heuristic, metaheuristic, simulated annealing greedy.

I. INTRODUCTION

THE combinatorial optimization on auction biding problems has been an important matter of studying subjects, and as in the real world the using distributed informatics systems have been extended, resource allocation with a controlled price in IT systems have been common item in the research fields that the number of applications can be computational power allocation, permanent memory, random temporary memory and bandwidth which such a resource allocation with limitation is named combinatorial auctions. Combinatorial auctions of goods and services on internet like shape a number one from the view of negotiation aspects could have different dimensions such as assortment, quality of goods, variation of criteria (price, quality, delivery time and so on etc or an mixed combination of them). So these advanced auctions formats can be categorized by several different dimensions of the goods in negotiation (as illustrated in Fig. 1) [1] :

- the number of different items in negotiation (e.g., a single or multiple line items),
- the negotiable qualitative attributes (e.g., a single or multiple attributes), and
- the quantity for each item (e.g., a single or multiple units).

S. Khanpour is with Islamic Azad University – Babol Branch, Babol, Iran (phone:+98 111 3293149; fax: +98 111 3293149; e-mail: s_khanpour@ yahoo.com).

A. Movaghar is Associate Professor in Sharif University of Technology Department of Computer Engineering, Tehran, Iran (phone: +98 21 66164628; fax: +98 21 66019246; e-mail: movaghar@sharif.edu).

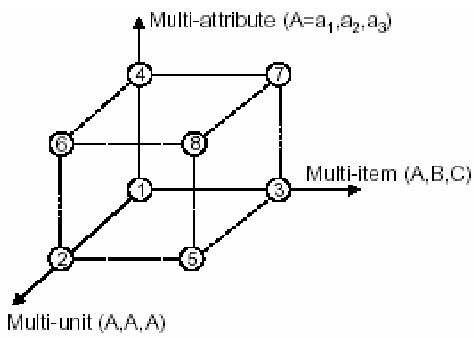

Fig. 1 Dimensions of a trade negotiation

In this paper *Multi Item single Unit Auction* [2] is modeled and implemented. Degree of created search space in such problems is exponential in complexity degree (order time), so computation optimum allocation is from NP-complete problem. Because of avoiding from creation of complete whole search space, it is better to use heuristic and meta heuristic search techniques to find out optimum allocation.

II. REVIEW OF RELATED WORK

There is huge amount of studies on internet auction from theory studies through practical studies have been down till this time. The major parts of these studies have been down mainly in the binding combinatorial auctions. Winner determination in such as auctions is a complex problem with its limitations that formulation could be solved by using of Set Packing Problem. In [3] Fujishima a nearly aspect for finding out optimum oriented allocation, under the name of Virtual Synchronization Algorithm presented that was under interest of market based computing. However the function of V.S.A. was not very satisfying and in comparison with his next method, under the name of Combinatorial Auction Structural Search was weaker. C.A.S.S. has been erected on the bases *depth-first Branch & Bound* search and could improve the efficiency of optimum winner determination algorithm in that time. In [2] Anderson, he observed that a winner determination problem to be modeled as mixed integer programming problem and he should the way that one can solve the problem with to write programming algorithm and today business software (IBM, CEPLEX). In [4] Hoos, Boutilier they presented nearly method under the name of Casanova based on stochastic local search. In [5] Leyton-Brown presented Combinatorial Auction Multi-Unit under the

name of C.A.M.U. that was a new version of C.A.S.S. for Combinatorial Auction Single-Unit. In [6] Mr. Sandholm and Suri presented optimum algorithm on the base of *depth-first branch & bound* for generalization combinatorial auctions, for example Combinatorial Auction Multi-Unit, Double Auction and Auction with Price Reservation possibility. The algorithm of Mr. Sandholm could find a reasonable optimum allocation for 200 items of goods and 200 bids in a time of 10000 second. In the 2003 a team of tree people on university of Singapore succeed to find nearly a optimum solution for this problem by using of Simulated Annealing method, and they could improve the all old methods of heuristic. Further more they presented the results of their work with a 500 sample test files in this address that was a benchmark in our assessing in the performance analysis my approach (designed and implemented algorithm in this paper). URL Address is:
http://logistics.ust.hk/~zhuyi/instance.zip.

The simulated annealing is a meta heuristic method which different from traditional heuristic methods such as *hill climbing.* Because in this method, the movements and replacements are in the current condition that decreasing the optimization of the solution. In fact *simulated annealing* method has two basic parts which are local search and scheduling of initial temperature decrease. This special method has been shaped from hybrid meta heuristic method that basely *simulated annealing* method simultaneity with *greedy search* for selecting of local movements.

III. Problem Model

Assume there are n bids and m jobs and each bid can cover a number of jobs resulting in a profit to the supplier, w_j ($j \in 1, ..., n$) is the profit if bid j is selected, and $[a_{ij}]m \times n$ is a m-row, n-column 0-1 matrix, where $a_{ij} = 1$ if job i is included in bid j. Further, the decision variables, $x_j = 1$ if the supplier selects bid j, and 0 otherwise. An integer programming (IP) model for the brokering problem as a *Set Packing Problem problem*(*S.P.P.)* [7],[8] is then:

$$Maximize \sum_{j \in N} w_j . x_j \tag{1}$$

$$Subject\ \ to:$$

$$\sum_{j \in N} a_{ij} . x_j \leq 1 \ , \ i \in M \tag{2}$$

$$x_j \in \{0,1\} \quad j \in N \, , \ N = \{1,2,...,n\} \, , \ M = \{1,2,...,m\} \tag{3}$$

where $N = \{1, . . . , n\}$, $M = \{1, . . . , m\}$.The first set of constraints ensure that each row is covered by at most one column and the second integrality constraints ensure that $x_j = 1$, *iff* column *j* of the matrix is in the solution[9].

IV. Genetic Algorithm

The usual form of genetic algorithm was described by Goldberg [10]. Genetic algorithms are stochastic search techniques based on the mechanism of natural selection and natural genetics. Genetic algorithms, differing from conventional search techniques, start with an initial set of random solutions called *population*. Each individual in the population is called a *chromosome*, representing a solution to the problem at hand.

V. Genetic Algorithm Framework

In the various of references represented different genetic algorithms, but all have same body and structure, only different is in the population size, crossover type and mutation type [11],[12]. The total structure is:

step1 *: initialization (initial population)*
step2 *: evaluation*
step3 *: selection*
step4 *: crossover*
step5 *: mutation*
step6 *: jump to step2 if condition not satisfied.*

VI. Design and Implementation Proper Genetic Algorithm

Encoding: the problem is discrete, so the best encoding algorithm is recognized *Random Key Encoding* [13]. For example if existence four bids (B1,B2,B3,B4) (L= 4: key sequence) first generates four random numbers between 0 , 1 that generated random numbers sequenced by r = (0.07, 0.75 , 0.56 , 0.67), sophisticate this order is permutation of keys value, this means auctioneer firstly accepted bid B2 and secondly B4 accepted if acceptance of bid B4 with accepted bid B2 not conflict and so on B3 then B1 will be accepted, so r_s = B2,B4,B3,B1 could be the one of the solutions.

Fitness function: the fitness function value in this problem is total benefit of accepted bids.

Total fitness value $= \sum_{1}^{n} fitness(Bi)$ if B_i Accepted by Auctioneer.

Population size: according to number of bids and the type of this problem, it is better that population size selected equal 2 or 2.5 time bids.

Selection: the approach has both random and deterministic features, is more than other various approaches, tournament selection method with 2 size used in this implementation [14]. At first the population area randomly divided to sets with two members and secondly from each set the one of that have more benefit is selected and two copied moved to next generation population. This approach increases the survival of robust chromosomes.

Crossover operator: the operator with two random points is used in this problem. This means for each pair chromosomes from population that candidate to mating, two random points is selected. Because the length of each chromosome is variable such as *Knapsack Problem*, firstly the select two points that begin in shorter chromosome range by randomly, secondly for reach to more speed in convergence, *Injection Crossover* is applied.

1. select two points from shorter parent, and candidate exchange segments in pair parents.
2. exchange and insert segments to first random point position of parents.
3. remove repeated genom in new chromosomes from outside of segment chromosomes for generates offsprings.
4. validiate of consistence new chromosomes, if existence conflitions whoud use *first fit heuristic* method.

Mutation operator :

1. remove one genom from candidate chromosome for mutation by randomly.
2. insert the genom that is not in exist genoms in chromosome.
3. validiate consistence genoms inside new chromosome, if conflict , select *first fit heuristic* method .

Note : In genetic algorithm, crossover operator is for convergency of chromosomes to reach optimum solution rapidly, mutation operator is for decreasing speed of convergency chromosomes, and prevent of go into local optima solution and loss probability robust solutions. So according to problem type we must make a tradeoff between crossover rate and mutation rate. In this problem, crossover rate between 0.4, 0.7 and mutation rate between 0.01, 0.06 are modifying dynamically. The followings are pseudo code, with Microsoft Visual Studio C++ implemented.

```
-------------------------------------------------------------------------------
Combinatorial Auction Winner Determination with Genetic Algorithm
Begin
  Primary parameter determination // popsiz , pCrossover ,
                                     pMutation , GenerationSize
  Initiate p(t)              // generate of intial population
  Evaluate p(t) // evaluation of choromosome's fitness in population
   While condition do
    Begin
     t = t+1                 // sequence of generations
     Selet p(t) from p(t-1) // Binary Tournament Selection
     Alter p(t)  // recombination:two point injection Crossover & Mutation
     Evaluation
    End
End
-------------------------------------------------------------------------------
```

VII. Experimental Results

The following table compares performance of tree algorithms: *Iterative Greedy Search (I.G.), Simulated Annealing Greedy (S.A.G.)* and designed and implemented *Genetic Algorithm (G.A.)* in this paper.

TABLE I
COMPARISON BETWEEN G.A. , S.A.G. AND I.G. HEURISTICS ON LARGE INSTANCES

M	N	# instance	*µI.G.*	*µS.A.G.*	*µG.A.*	S.A.G. performance compare to I.G.	G.A. performance compare to S.A.G.
1000	500	100	70295.46	74750.66	77748.16	%6.34	%4.01
500	1000	100	62732.89	65874.39	68720.16	%5.01	%4.32
1000	1000	100	80256.87	85205.58	88520.08	%6.17	%3.89
1000	1500	100	80404.98	84422.91	86972.48	%5.00	%3.02
1500	1500	100	98255.77	103426.68	108473.9	%5.26	%4.88

In order to compare the performance of the implemented Genetic Algorithm , S.A.G. and I.G. heuristic in real world situations, we used pre-generated several sets of large test cases of the 500 instances are available at:
http://logistics.ust.hk/~zhuyi/instance.zip,
to serve as our benchmark. The result of 500 test cases files are grouped into 5 sets up to m = 1500 jobs and n=1500 bids as in Table I, where *µG.A.* , *µS.A.G.* and *µI.G.* are the arithmetic average of the 100 instances in each group for G.A., S.A.G. and I.G. method respectively. From Table I, we see that the S.A.G. method always gives a 5 to 6 percent improvement in results in comparison to I.G., and G.A. method always gives a 4 percent improvement in results in comparison to S.A.G.. According to above notes, the algorithms based on creation initial random search area of solutions and improvement and reinforcement those results better solutions (chromosomes) on the next of generations in comparison with I.G. and S.A.G..

VIII. Conclusion

In this paper, we modeled the combinatorial auction brokering problem as a NP-complete *Set Packing Problem* and designed a genetic algorithm method to solve the problem. For obtaining global optimum in large scale search space we can use from different heuristic and meta heuristic methods. By attention to the results of Table I, we can understand that designing and implementation this problem by using of genetic algorithm from the view of performance, is better than S.A.G.. In a way that the S.A.G. was the best method in comparison with other heuristic methods till 2003, further more by using of techniques in the proposition part would get better results. So from one hand using of optimization technique in genetic algorithm is a new phenomenon in the e-Commerce scope, and from the other views like flexibility, scalability and generalizability the genetic algorithm give better result in comparison with other methods.

Flexibility: on the base of your need you can make a tradeoff between performance and consuming time by tuning effective parameters in convergency speed.

Scalability: we can find out optimum allocation for larger scale problems by using of genetic algorithm.

Generalizability: by attending to simplified structure and encoding type in this implementation we can generalize it to other applications such as *Multi Unit Multi Item Auctions* by easily. The other capabilities the implemented G.A. in this paper are increasing of generation number and mutation rate dynamically by attending to convergency amount during continuous generations.

IX. Propositions

So according to behavior of genetic algorithm, especially the behavior of above design and implemented algorithm of this paper, may could be improved performance this approach in this paper by applying random number generator function of MATLAB software environment for generate random number with more quality and uniform distribution, and dynamically generation size with fuzzy linear functions. Our future researches are above issue and generalization of *Multi Item Single Unit Auction* in this implementation to *Multi Item Multi Unit Auction* for applying in Share Exchange Market.

References

[1] Martin Bichler, Jayant Kalagnanam, Ho Soo Lee, Juhnyoung Lee, "*Winner Determination Algorithms for Electronic Auctions: A Framework Design",* IBM T. J. Watson Research Center, 2002.

[2] M. Tenhunen A. Anderson and F. Ygge, "Integer programming for combinatorial auction winner determination", in *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS00)*, pages 39–46, 2000.

[3] Y. Fujishima, K. Leyton-Brown, and Y. Shoham, "Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches", in *the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 548–553, 1999.

[4] H. H. Hoos and C. Boutilier, "Solving combinatorial auctions using stochastic local search", in *the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, 2000.

[5] K. Leyton-Brown, Y. Shoham, and M. Tennenholtz, "An algorithm for multi-unit combinatorial auctions", in *the Seventeenth National Conference on Artificial Intelligence (AAAI- 2000)*, 2000.

[6] T. Sandholm and S. Suri, "Improved algorithms for optimal winner determination in combinatorial auction and generalization", in *the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, 2000.

[7] E. Balas and A. Ho, "Set covering algorithm using cutting planes, heuristics, and sub gradient optimization: A computational study"; in *Mathematical Programming*, volume 12, pages 37–60, 1980.

[8] Hoong Chuin Lau and Yam Guan Goh, "An intelligent brokering system to support multi agent web-based 4th-party logistics", in *Proceedings of the Fourteenth International Conference on Tools with Artificial Intelligence*, pages 10–11, 2002.

[9] Y.Guo, A.Lim, B.Rodrigues, "Heuristics for Brokering Set Packing Problem", Department of Computer Science, National University of Singapore, 2003.

[10] Genetic Algorithms in search, optimization and machine learning, Goldberg, 1989.

[11] A. Arjenahi ,M. S. Thesis "Design and Implementation Genetic Algorithm Processor" , Sharif Uni. Tech. , November 2004.

[12] Genetic Algorithms + Data Structure = Evolation Programs, Michalewicz, 1996.

[13] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization", *ORSA Journal on Computing*, vol. 6, no. 2, pp. 154-160, 1994.

[14] Genetic Algorithms and Engineering Design, Gen, 1998.