INTERNATIONAL JOURNAL OF APPLIED MATHEMATICS AND INFORMATICS
DOI: 10.46300/91014.2020.14.7 Volume 14, 2020

An exact method for the Multi-objective Winner Determination Problem of combinatorial auctions

Chahrazad Adiche
LaROMaD, Department of Mathematics,
Faculty of Sciences
U.M.B.Boumerdes, 35000 Algeria
Email: adichechahra@yahoo.fr

Méziane Aïder
LaROMaD, Faculty of Mathematics,
U.S.T.H.B., B.P. 32 El Alia,
Bab Ezzouar 16111 Algiers, Algeria
Email: m-aider@usthb.dz

***Abstract*—We are interested by the problem of combinatorial auctions in which multiple items are sold and bidders submit bids on packages. First, we present a multi-objective formulation for a combinatorial auctions problem extending the existing single-objective models. Indeed, the bids may concern several specifications of the item, involving not only its price, but also its quality, delivery conditions, delivery deadlines, the risk of not being paid after a bid has been accepted and so on. The seller expresses his preferences upon the suggested items and the buyers are in competition with all the specified attributes done by the seller. Second, we develop and implement an exact algorithm based on a multi-objective branch-and-bound method.**

Keywords: multi-objective combinatorial optimization, combinatorial auctions, multi-objective branch-and-bound method.

I. Introduction

The general multi-objective combinatorial optimization problem can be expressed as:

$$\text{(MOCO)} \begin{cases} \text{“max”} F(x) = (f^1(x), f^2(x), \ldots, f^p(x)) \\ x \in S \end{cases}$$

where $p \geq 2$ is the number of objective functions, $x = (x_1, x_2, \ldots, x_d)$ is the vector representing the decision variables, S is the (finite) set of feasible solutions in the solution space $\mathbb{R}^d$. The set $Z = F(S)$ represents the feasible points (outcome set) in the objective space $\mathbb{R}^p$ and $z = (z^1, z^2, \ldots, z^p)$, with $z^i = f^i(x)$, is a point of the objective space.

Observe that, in (MOCO), the term “max” appears in quotation marks because, in general, there does not exist a single solution that is maximal on all objectives. As a consequence, several concepts must be established to define what an optimal solution is. The more used one is the dominance relation also known as Pareto dominance (see Fig. 1.).

Definition 1: We say that a point $z = (z^1, z^2, \ldots, z^p)$ dominates a point $w = (w^1, w^2, \ldots, w^p)$ and we write $z \succeq w$ if and only if for all $i \in \{1, \ldots, p\}$, $z^i \geq w^i$ with for at least one $i_0 \in \{1, \ldots, p\}$, $z^{i_0} > w^{i_0}$.

Definition 2: A solution $x^* \in S$ is called (Pareto) efficient for (MOCO) if and only if there does not exist any other feasible solution $x \in S$, such that x dominates x^*. The point $F(x^*)$ is then called a non-dominated point.

The set of efficient solutions, also called the Pareto optimal set, is often denoted by E and the image of E in Z is called the non-dominated frontier or the Pareto optimal front, and is denoted by Z_E.

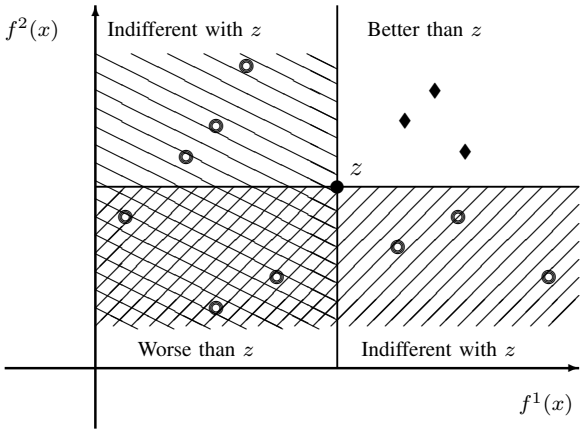

Fig. 1. Dominations in the Pareto sense in a bi-objective space.

Note that if $x, y \in S$ are such that $F(x)$ dominates $F(y)$ we usually say that x dominates y and we also write $x \succeq y$.

In (MOCO), we can optimize each of the objectives by solving the following problems:

$$\text{(COP}(k)\text{)} \begin{cases} \max\ f^k(x) \\ x \in S \end{cases} \qquad k = 1, \ldots, p$$

Suppose that $x^{k*},\ k = 1..p$ are optimal solutions of the above problems respectively. Then, the optimal value of objective k is given by $f^{k*} = f^k(x^{k*})$.

Definition 3: The point $F^* = (f^{1*}, f^{2*}, \ldots, f^{p*})$ is called the ideal point in the objective space.

In general, an ideal point is not a feasible solution. Otherwise, the objective would not be in conflict with one another.

II. Single-objective combinatorial auction models

The auctions research started essentially in 1961 with the Nobel prized economist William Vickrey, but the early work on auctions first appeared in operations research journals with Friedman [5] and Rothkopf [12]. Since then, the field of auctions studies has grown to more wide multidisciplinary fields like economics, games theory, operations research, computer science, decision analysis, multicriteria decision making, ...

Numerous applications have been reported in the literature for combinatorial auctions. They have been employed in a variety of industries (truckload transportation, bus routes, industrial procurement, ...), in airport arrival and departure slots, in telecommunication (allocating radio spectrum), in electronic business (eBAY, ...), in public sector for procuring meals for schools, ...

In combinatorial auctions, the auctioneer has a set M of m items ($M = \{a_1, a_2, \ldots, a_i, \ldots, a_m\}$) to sell, and the buyers submit a set B of n bids, ($B = \{B_1, B_2, \ldots, B_j, \ldots, B_n\}$). The compelling motivation for the interest on such problem is the presence of complementarities (the value of the whole bundle is larger than the sum of the values of its components taken separately) and substitutions (the bidder only wants one of the items) among the items. These characteristics differ across bidders and allow them to fully express their preferences. A bid is a tuple $B_j = \langle S_j, c_j \rangle$, where $S_j \subseteq M$ is a set of items and c_j is a price for the whole package S_j. The selection of the winning bids becomes in this case more complicate (NP-hard problem [14]). This problem is known as the Winner Determination Problem (WDP) of combinatorial auctions and the most research on this area focuses on the computational issues [3].

A. Single-unit case

The Winner Determination Problem in the single-unit case is to label the bids $j = 1..n$ as winning ($x_j = 1$) or losing ($x_j = 0$), so as to maximize the auctioneer's revenue under the constraint that each item can be allocated to at most one bidder:

$$\text{(WDP)} \begin{cases} \max Z(x) = \sum_{j=1}^{n} c_j x_j & \\ \sum_{j/i \in S_j}^{n} x_j \leq 1 & i = 1, \ldots, m \\ x_j \in \{0,1\} & j = 1, \ldots, n \end{cases}$$

(WDP) is intractable. The branch and bound approaches ([15], [16], [17]) are the most common usual methods in the single unit case. Exact methods guarantee that an optimal solution is found but do not guarantee the running time! Recently, heuristics have been introduced to solve (WDP) in combinatorial auctions ([10], [8], [13]).

B. Multi-unit case

In this case, the auctioneer has some number μ_i of available units of each item a_i ($i = 1..m$). The buyers submit a set of bids $\{B_1, B_2, \ldots, B_j, \ldots, B_n\}$. A bid is a tuple $B_j = \langle \{\lambda_j^i, \lambda_j^1, \ldots, \lambda_j^i, \ldots, \lambda_j^m\}; c_j \rangle$, where λ_j^i is the (non negative integer) number of units of the item a_i ($i = 1..m$), required by the j-th buyer ($j = 1..n$). The corresponding model is given by (WDP').

$$\text{(WDP')} \begin{cases} \max Z(x) = \sum_{j=1}^{n} c_j x_j. & \\ \sum_{j=1}^{n} \lambda_j^i x_j \leq \mu_i & i = 1, \ldots, m \\ x_j \in \{0,1\} & j = 1, \ldots, n \end{cases}$$

Several exact approaches have been used for solving (WDP'): dynamic programming [14], linear programming [11], integer programming [1] and constraint programming [9].

III. Multi-objective combinatorial auctions

Most studies in the literature are focus either on single-unit combinatorial auctions with price only (single-objective) or on single-item (but non-combinatorial auctions) with multi-objective auctions. However, both auctions types alone are already very complicated. Thus far, there has not been much work on multi-objective (multi-attribute) combinatorial auctions and the most of works in this area use the weighting function to translate the multi-objective into utility function or use a single objective branch-and-bound algorithm based on the $\varepsilon - constraint$ method [2] and a very recent work on the e-constraint method for finding the exact pareto set in multi-objective integer programs [7].

A. Problem formulation

The multi-objective formulation of (WDP') is:

$$\text{(MOWDP')} \begin{cases} \text{“opt”} Z^k(x) = \sum_{j=1}^{n} c_j^k x_j & k = 1, \ldots, p \\ \sum_{j=1}^{n} \lambda_j^i x_j \leq \mu_i & i = 1, \ldots, m \\ x_j \in \{0,1\} & j = 1, \ldots, n \end{cases}$$

where c_j^k is the value of the bid j ($j = 1, \ldots, n$) for the criterion k ($k = 1, \ldots, p$) and the decision variables are defined as follows:

$$x_j = \begin{cases} 1 & \text{if the bid } B_j \text{ is accepted (a winner offer);} \\ 0 & \text{otherwise.} \end{cases}$$

The seller expresses his preferences upon the suggested items and the buyers are in competition with all the specified attributes done by the seller. So, the Multi-Objective Winner Determination Problem (MOWDP') consists of finding the accepted bids which simultaneously, for example, maximize the revenue of the seller and minimize the payment time, under the constraints that at most the available number of units of each item is allocated.

An acceptable bid (non risk of overlapping with other bids) for which the vector of specifications (revenue vector) is not dominated by any other vector of specifications of bids, is an *efficient* solution for (MOWDP').

B. Multi-Objective Branch and Bound method for (MOWDP')

In this subsection, we propose an adaptation of the branch-and-bound method dedicated to the multi-objective knapsack problem type in $0 - 1$ [4], to the Multi-Objective Winner Determination Problem (MOWDP'). The addition of multiple units of each item to (WDP) involves too many possible combinations to evaluate and so, causes new levels of complications in the auctions process. Furthermore, the mathematical formulation of the studied problem (MOWDP') is closely related to the multi-objective multiconstraint knapsack problem. Many other practical problems can also be formulated like the multiconstraint knapsack problem [18].

In the branch-and-bound scheme, the solution space is explored by dynamically building a tree and by using the following three basic procedures:

- Procedure of separation
  - To define a partition of the fundamental set S.
  - To choose a (node) subset to be treated.
- Procedure of evaluation
  - To precise how to evaluate each node.
- Procedure of sterilization
  - To determine when a node can be pruned.

a) Procedure of separation: Partial solutions (nodes of the search tree) are created by assigning zeros and ones to subsets of bids denoted β_0 and β_1, respectively. Bids that are not assigned either zero or one define the set $\mathcal{F} \subseteq \{1, 2, \ldots, n\}$ of the *free bids* , thus, we have $\{1, 2, \ldots, n\} = \beta_0 \cup \beta_1 \cup \mathcal{F}$.

The branching sequence is crucial for the performance of the method. Let θ be the order according to which variables (bids) of a partial solution will be assigned a value. The order θ can be defined as by Florios et al. [6] according to the increasing values of either (1) or (2) heuristics rules:

$$Ave\_sort_j = (p.m)^{-1} \sum_{k=1}^{p} \sum_{i=1}^{m} \frac{c_j^k}{\lambda_j^i} \quad j = 1, \ldots, n. \tag{1}$$

$$max_j = \max_{k=1,\ldots,p;\ i=1,\ldots,m} \frac{c_j^k}{\lambda_j^i} \quad j = 1 \ldots, n. \tag{2}$$

In the particular case, when we have two objectives such that the first is to maximize (e.g. the revenue) and the second is to minimize (e.g. the payment time) we propose to order the bids according to increasing values of the following formula:

$$\text{quot}_j = \frac{c_j^1}{c_j^2} \quad j = 1, \ldots, n. \tag{3}$$

The problem corresponding to the partial solution (β_1, β_0) is again a Multi-Objective Winner Determination Problem:

$$\text{(MOWDP')} \begin{cases} \text{“opt”} Z^k(x) = \sum_{j \in \mathcal{F}} c_j^k x_j + \sum_{j \in \beta_1} c_j^k & k = 1, \ldots, p \\ \sum_{j \in \mathcal{F}} \lambda_j^i x_j \leq \overline{\mu}_i & i = 1, \ldots, m \\ x_j \in \{0, 1\} & j \in \mathcal{F} \end{cases}$$

where:

$$\overline{\mu}_i = \mu_i - \sum_{j \in \beta_1} \lambda_j^i \qquad i = 1, \ldots, m. \tag{4}$$

A solution formed by assigning a value to all free variables is called a *completion* of a partial solution.

b) Procedure of evaluation: We can evaluate the subset S' of S if we know how to determine a vector $g(S') = (g^1(S'), \ldots, g^p(S'))$ in such a way that there exists no solution $\overline{s} \in S'$ such that $Z(\overline{s})$ *dominates* $g(S')$. If $S' = \emptyset$, then the only possible evaluation is:

- $g(S') = +\infty$ for minimization problem;
- $g(S') = -\infty$ for maximization problem.

For each node S' of the tree, we associate a vector valued bounds. To compute the bounds, we define:

- $\sum_{j \in \beta_1} c_j^k$, $k = 1, \ldots, p$: the value of the bids that have already been assigned value 1.
- $Z^{k*}(S')$: the value of the optimal solution according to the k-th objective, $k = 1, \ldots, p$.

The values of $\sum_{j \in \beta_1} c_j^k$ and $(Z^{k*}(S') + \sum_{j \in \beta_1} c_j^k)$ $(k = 1, \ldots, p)$ represent an estimation and an evaluation of the subset S', respectively.

TABLE I
LOWER AND UPPER BOUNDS AT A PARTIAL SOLUTION

Bounds	maximization problem	minimization problem
Lower bound	estimation	evaluation
Upper bound	evaluation	estimation

The components of the lower bound $\underline{Z}(S')$ and the upper bound $\overline{Z}(S')$ at the partial solution S', are given according to TABLE I.

If we consider two objectives, the first is to maximize and the second is to minimize, the lower bound $\underline{Z}(S')$ and the upper bound $\overline{Z}(S')$ at the partial solution S', are given by:

$$\underline{Z}(S') = \underline{z} = (\underline{z}_1, \underline{z}_2) = \left(\sum_{j \in \beta_1} c_j^1, \left(Z^{2*}(S') + \sum_{j \in \beta_1} c_j^2\right) \right) \tag{5}$$

$$\overline{Z}(S') = \overline{z} = (\overline{z}_1, \overline{z}_2) = \left(\left(Z^{1*}(S') + \sum_{j \in \beta_1} c_j^1\right), \sum_{j \in \beta_1} c_j^2 \right) \tag{6}$$

c) Procedure of sterilization: A subset S' of the set S of solutions of a multi-objective combinatorial auctions problem is said to be pruned if $S' = \emptyset$ or we know a solution $s^* \in S$ such that s^* dominates any solution of S'.

A vector valued bounds, defined above, allows us to prune a partial solution S', when no *completion* of S' can possibly contain an efficient solution. So, we do not need to develop further the exploration of such a node.

Assume that the bids B_1, B_2,..., B_n are ordered according to one of rules $(1) - (3)$:

Let $l_* = \min\{l : \lambda^l \leq \mu\}$, with

- $(\lambda^l)^t = \left(\sum_{j=1}^{l} \lambda_1^j, \sum_{j=1}^{l} \lambda_2^j, \ldots, \sum_{j=1}^{l} \lambda_m^j\right)$ and
- $\mu^t = (\mu_1, \mu_2, \ldots, \mu_m)$.

Thus, l_* is the smallest index l such that $\exists i_0$ with

$$\sum_{j=1}^{l} \lambda_{i_0}^j > \mu_{i_0}.$$

Observe that the critical bid is b_{l_*}. So, the subset $S' = \{B_1, B_2, \ldots, B_{l_*-1}\}$ of bids is a feasible solution because all bids in S' are not in conflict i.e. they have not items in common. The branch-and-bound algorithm (Algorithm 1) starts by fixing many bids according to the θ order to fastly find a good feasible solution. Thus, many branches of the tree can be pruned early. The list $\mathcal{N}$ of nodes is maintained as a LIFO stack (Last In First Out). When a node is pruned, the algorithm backtracks and creates a new node by moving the last bid in β_1 to β_0. In addition, all bids in β_0 after this new bid become free. If, however, n was the last bid in β_1, the algorithm removes all bids $\{v, \ldots, n\}$ in β_1 (v is the smallest one) and defines β_0 to be all previous elements of β_0 up to $v - 1$ and to include v. When a node is not pruned, the algorithm progresses deeper down the tree and creates a new successor node. Indeed, as many bids as possible are included in β_1, according to order θ, i.e. as they appear in $\mathcal{F}$. But if the remaining vector $\overline{\mu}$ does not allow bid l to be added to β_1, the first possible bid r of $\mathcal{F}$, which can be added to β_1 is sought and bid r is added to β_1. Of course, all bids $\{i, \ldots, r - 1\}$ must be added to β_0.
The method is summarized in Algorithm 1.

Algorithm 1 MOBB algorithm.

Require: Data of (MOWDP').
Ensure: A set of efficient solutions.

Initialization: Create the root node N_0 as follows:

$$\beta_1 := \emptyset,\ \beta_0 := \emptyset,\ \mathcal{F} := \{1, \ldots, n\},\ \mathcal{L} := \phi,\ \mathcal{N} := \{N_0\}.$$

while $\mathcal{N} \neq \emptyset$ **do**
  Choose the last node $N \in \mathcal{N}$.
  Compute $\underline{z}$ (lower bound at the partial solution).
  Add $\underline{z}$ to $\mathcal{L}$ if it is not dominated.
  Compute $\overline{z}$ (an upper bound at the partial solution).
  if $\{j \in \mathcal{F}$: B_j is not in conflict with $\beta_1\} = \emptyset$ **or** $\overline{z}$ is dominated by some $w \in \mathcal{L}$ **then**
    (Case 1.)
    Prune the node N. $\mathcal{N} := \mathcal{N} \setminus \{N\}$.
    Go backwards of the node N (node N is pruned).
    Create a new node N'.
    Update β_1, β_0 and $\mathcal{F}$.
    $\mathcal{N} := \mathcal{N} \cup \{N'\}$.
    if the set β_1 of N' is smaller than β_1 of the predecessor nodes of N, which are not predecessors of N', **then**
      Fathom these nodes.
    end if
    if $\beta_1 = \emptyset$ **then**
      Compute the upper bound, $\overline{z}$, at this partial solution.
      if there is a solution in $\mathcal{L}$ which dominates $\overline{z}$ **then**
        STOP (no new node can be created).
      end if
    end if
  else
    (Case 2.)
    Go deeper down the tree (node N is not pruned).
    Create a new node N'.
    Update β_1, β_0 and $\mathcal{F}$.
    $\mathcal{N} := \mathcal{N} \cup \{N'\}$.
  end if
end while

C. Didactic example

The method presented in Algorithm 1 is illustrated by the following example.

Let be:

- $M = \{a_1, a_2, a_3\}$ the set of three items to be auctioned.
- $\mu_1 = 5$, $\mu_2 = 10$ and $\mu_3 = 7$ (number of available units of the each item).
- The offers B_j $j = 1..7$ upon the set M and their revenue vectors c_j (where each of their components is to maximize) are done as follows:
  - $B_1 = \langle\{1, 2, 3\}; c_1 = (10, 12, 5)\rangle$
  - $B_2 = \langle\{1, 3, 2\}; c_2 = (6, 8, 10)\rangle$
  - $B_3 = \langle\{4, 6, 4\}; c_3 = (7, 5, 14)\rangle$
  - $B_4 = \langle\{1, 3, 0\}; c_4 = (9, 4, 17)\rangle$
  - $B_5 = \langle\{5, 2, 0\}; c_5 = (6, 3, 9)\rangle$
  - $B_6 = \langle\{1, 4, 0\}; c_6 = (13, 11, 6)\rangle$
  - $B_7 = \langle\{2, 7, 1\}; c_4 = (5, 4, 16)\rangle$

The conflict graph is given in Figure 2.

$B_1 = \langle\{1, 2, 3\}; c_1 = (10, 12, 5)\rangle$ (for instance) means that the bid B_1 contains one unit of item a_1, two units of a_2 and three units of a_3 and c_1 is its revenue vector.

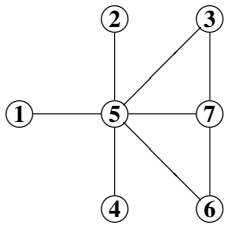

Fig. 2. The conflict graph

The order θ is computed according to the rule (2).

$$\theta = \{4, 7, 6, 1, 2, 5, 3\}.$$

There are four efficient solutions, presented in TABLE II.

TABLE II
SOLUTIONS OF THE DIDACTIC EXAMPLE

N	**Efficient bids**	**Revenue vector**
1	$\{B_2, B_4, B_6\}$	(28,23,33)
2	$\{B_1, B_4, B_6\}$	(32,27,28)
3	$\{B_1, B_2, B_4\}$	(25,24,32)
4	$\{B_1, B_2, B_6\}$	(29,31,21)

D. Experimental results

Numerical experiments are realized upon some randomly generated instances (no benchmarks have been found in the literature for the multi-objective case) of different sizes to test and prove the efficiency of our extended multi-objective branch-and-bound (MOBB) method. The instance pWDP$n - m$ provides the number of objectives (p), the type of problem (Winner Determination Problem), the number of bidders (n) and the number of items (m). We focus our experiments on a bi-objective case. However, the results remain valid for a larger number of objectives. The algorithms have been implemented in matrix laboratory (Matlab R2009a) using a Pentium PC with dual core processor, FSB 800 Mb, DDR1 2 Go in Windows operating system. Experimental results are provided in TABLE III. They show that the CPU time (measured in second) of the MOBB method depends on the size of (MOWDP'). The complete set of efficient solutions can be generated in reasonable computational time only for small problems. For large problems, one can use metaheuristics or approximate methods.

TABLE III
RESULTS OF THE EXTENDED MOBB METHOD

Instances	$\lvert E \rvert$	**CPU(t)**
2WDP5-3	1	0.23
2WDP7-3	3	0.49
2WDP8-5	3	1.57
2WDP10-3	4	1.20
2WDP10-5	4	2.01
2WDP15-3	7	4.33
2WDP20-3	5	9.14
2WDP20-7	6	25.73
2WDP25-3	7	32.85
2WDP30-3	8	73.76
2WDP30-9	7	100.99
2WDP35-3	14	127.37
2WDP40-3	19	2837.81
2WDP45-3	21	3372.71
2WDP50-3	20	7242.21

IV. Conclusion

In this paper we have described a Multi-Objective Branch-and-Bound (MOBB) method developed for the Winner Determination Problem of combinatorial auctions which can be modeled as a multi-objective multiconstraint knapsack problem. The multi-objective branch-and-bound used here was initially developed for the bi-objective knapsack problem presented in [4]. We have adapted it to the multi-objective combinatorial auctions problem, in order to make it possible of handling more than one constraint. A didactic example was presented to illustrate our method.
In future works, we propose to develop better bounds for the multi-objective branch-and-bound to improve the speed of the method. More experiments are needed and relations with other combinatorial problems (knapsack, bin packing, etc) will be examined.

Moreover, one can adapt the newly developed method based on a differential evolution algorithm (DEA) [19] or other metaheuristics to solve the multi-objective winner determination problem.

References

[1] A. Anderson, M. Tenhunen and F. Ygge, "Integer programming for combinatorial auction winner determination," in Proceedings of 4th International Conference on Multi-Agent Systems, IEEE Computer Society Press, July, 39–46, 2000.
[2] T. Buer and G. Pankratz, "A Bi-Objective Winner-Determination Problem in a Transportation-Procurement Auction," in Working paper No. 488, Faculty of Business Administration and Economics, University of Hagen (Germany), 2010.
[3] S. De Vries and R. Vohray, "Combinatorial auction: A survey," Tech. rep., Department of Managerial Economics and Decision Sciences, Kellogg School of Management, Northwestern University, 2000.
[4] M. Ehrgott, Multicriteria Optimization (2. ed.), isbn 978-3-540-21398-72, Springer, 2005.
[5] L. Friedman, "A Competitive Bidding Strategy," Operations Research, 4, 104–112, 1956.
[6] K. Florios, G. Mavrotas and D. Diakoulaki, "Solving multiobjective, multiconstraint knapsack problems using mathematical programming and evolutionary algorithms," European Journal of Operational Research, Vol. 203, No. 1, 14–21, 2010.
[7] K. Florios and G. Mavrotas, "An improved version of the augmented e-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems," Applied Mathematics and Computation, Volume 219, Issue 18, 9652-9669, 2013.
[8] Y. Guo, A. Lim, B. Rodrigues, Y. Zhu, "Heuristics for a bidding problem," Journal of Computers and Operations Research -Volume 33 no 8: 2179–2188, 2006.
[9] A. Holland, B. O'sullivan, "Towards Fast Vickrey Pricing using Constraint Programming," Artificial Intelligence Review, Vol 21, n 3-4/ June, 335–352, 2004.
[10] H. Hoos, C. Boutilier, "Solving combinatorial auctions using stochastic local search," in: Proceedings of the 17th national conference on artificial intelligence, 22–29, 2000.
[11] N. Nisan, "Bidding and allocation in combinatorial auctions," in Proceedings of the ACM Conference on Electronic Commerce (EC-00), Minneapolis: ACM SIGecom, ACM Press, October, 1–12, 2000.
[12] Rothkopf, H. Michael, "A Model of Rational Competitive Bidding," Management Science, 15, 362–372, 1969.
[13] A. Portilla-Figueras, S. Salcedo-Sanz, P. Garcia-Diaz and K. Hackbarth, "A Genetic Algorithm for Solving the First Price Sealed Bid Auction in Communication Networks," Proceedings of the 5th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications, Madrid, Spain, February 15-17, 2006, 1–6.
[14] M.H. Rothkopf, A. Pekee and M. Ronald, "Computationally manageable combinatorial auctions," Management Science, Vol. 44, No. 8, 1131–1147, 1998.
[15] T. Sandholm, S. Suri, "Improved Optimal Algorithm for Combinatorial Auctions and Generalizations," in Proceedings of the 17th national conference on artificial intelligence, 9066–97, 2000.
[16] T. Sandholm, S. Suri, A. Gilpin, D. Levine, "CABoB: a fast optimal algorithm for combinatorial auctions," in Proceedings of the International joint conferences on artificial intelligence, 1102–1108, 2001.
[17] T. Sandholm, "Optimal Winner Determination Algorithms," in P. Cramton et al. (ed.), Combinatorial Auctions, MIT Press, 2006.
[18] B. K. Seljak, "Computer-Based Dietary Menu Planning," in Proceedings of the 7th WSEAS International Conference on Evolutionary Computing, Cavtat, Croatia, June 12-14, 2006, 39–44.
[19] R. Thangaraj and M. Pant, "Differential Evolution Algorithm for Solving Multi-objective Optimization Problems," in Proceedings: Recent Advances in Mathematics, Cambridge, MA, USA, January 30 - February 1, 2013, 40–45.