

A Tabu Search 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 a metaheuristic algorithm based on a tabu search method.

Keywords: multi-objective combinatorial optimization, combinatorial auctions, metaheuristics, tabu search.

I. INTRODUCTION

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

$$(MOCO) \left\{ \begin{array}{l} \text{“max”} F(x) = (f^1(x), f^2(x), \dots, f^p(x)) \\ x \in S \end{array} \right.$$

where $p \geq 2$ is the number of objective functions, $x = (x_1, x_2, \dots, x_d)$ 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 $z = (z^1, z^2, \dots, z^p)$, with $z^i = f^i(x)$, is a point of the objective space.

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

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 .

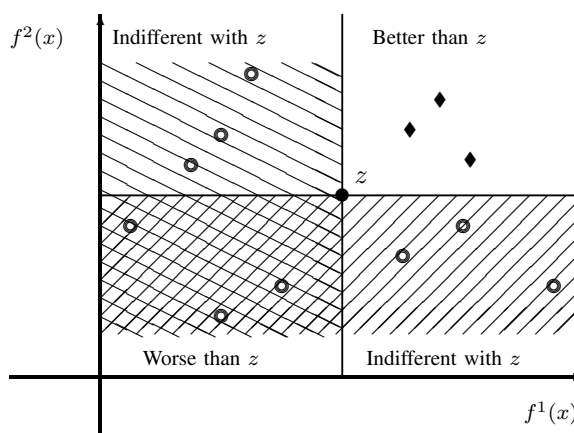


Fig. 1. Dominations in the Pareto sense in a bi-objective space.

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:

$$(COP(k)) \left\{ \begin{array}{l} \max_{x \in S} f^k(x) \\ k = 1..p \end{array} \right.$$

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 $F^* = (f^{1*}, f^{2*}, \dots, f^{p*})$ 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 [6] and Rothkopf [14]. 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 ($M = \{a_1, a_2, \dots, a_i, \dots, a_m\}$) to sell, and the buyers submit a set B of n bids, ($B = \{B_1, B_2, \dots, B_j, \dots, B_n\}$). 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 [15]). 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 [4].

A. Single-unit case

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

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

(WDP) is intractable. The branch and bound algorithms ([16], [17], [18]) 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 and approximate methods have been introduced to solve (WDP) in combinatorial auctions ([11],[9]).

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 $\{B_1, B_2, \dots, B_j, \dots, B_n\}$. A bid is a tuple $B_j = \langle \{\lambda_j^1, \lambda_j^2, \dots, \lambda_j^i, \dots, \lambda_j^m\}; c_j \rangle$, 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').

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

Several exact approaches have been used for solving (WDP'): dynamic programming [15], linear programming [13] integer programming [1] and constraint programming [10].

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 ε -constraint method [3].

A. The problem formulation

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

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

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

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

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. Tabu Search (TS) method for (MOWDP')

In this subsection, we propose an adaptation of the Tabu Search (TS) method to (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 (MOWDP') is closely related to the multi-objective multiconstraint knapsack problem. Tabu Search is a local search strategy [8] used for intensifying the research and designed for escaping from local minima. The main components of tabu search are:

- Research space.
- Neighborhood.
- Tabu list.
- Random walk.

a) *Research space*: A solution is given by a binary vector $x = (x_1, x_2, \dots, x_n)$ which verify all constraints i.e.,

$$\sum_{j=1}^n \alpha_j^i x_j \leq \mu_i, \forall i \in \{1, \dots, m\}.$$

Research space S is composed of all these binary vectors, i.e.,

$$S = \left\{ x \in \{0, 1\}^n / \sum_{j=1}^n \alpha_j^i x_j \leq \mu_i, i \in \{1, \dots, m\} \right\}.$$

For generating an initial solution we use the Random Key encoding introduced by Bean [2]. We generate a sequence r of n random real numbers between 0 and 1 where n is the number of submitted bids. To each bid B_j , a key value r_j , which constitutes its order of selection, is associated. The first bid to choose and to include in the selection is the one with the largest key value. Then, the bid having the second largest key in the collection is accepted if its acceptance does not create any conflict with bids already selected, otherwise it is rejected. And so on, until all bids are reviewed. We get a subset

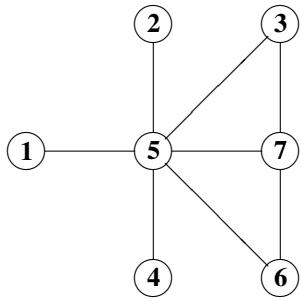


Fig. 2. The conflict graph

of bids that may be a solution to (MOWDP’).

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 example) 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.

- First, we generate a key sequence r of seven random numbers between 0 and 1: $r = (0.07, 0.75, 0.4, 0.09, 0.29, 0.56, 0.67)$, then the bids $B_j, j = 1, \dots, n$, are ordered according to the increasing value of their keys r_j : $B_2, B_7, B_6, B_3, B_5, B_4, B_1$.
- Second, a bid is accepted if it is in conflict with no of the previous accepted bids. Thus, the bids B_6, B_3, B_5 are not accepted because they are in conflict with the bid B_7 (see Figure 2). So, $x^0 = (1, 1, 0, 1, 0, 0, 1)$ could be an initial solution.

b) Neighborhood: The neighborhood \mathcal{N} of our problem is defined in the following way: let x and x' in S , x and x' are neighbors if and only if they differ exactly in one component. It results that from a current solution x , it is possible to obtain a neighbor solution x' ($x' \in \mathcal{N}(x)$) by adding or removing a bid such that x' remains feasible (it does not conflict with all previous accepted bids). The movement from x to x' is then characterized by the integer j which is considered to be the attribute of movement and represents the index of the component x_j that was changed i.e., $(x_j : 0 \rightarrow 1 \text{ or } 1 \rightarrow 0)$.

c) Evaluation of the Neighborhood: The evaluation of the neighborhood is based on weighted Tchebychev metrics [19] for measuring the distance, $\|z(x') - z(x)\|_\infty^\lambda$, between two neighbors $x = (x_1, x_2, \dots, x_n) \in S$ and $x' = (x'_1, x'_2, \dots, x'_n) \in \mathcal{N}(x)$. Thus, the assessment of x' is defined by the following function:

$$\text{eval}(x, x') = \|Z(x') - Z(x)\|_\infty^\lambda \quad (1)$$

$$= \max_{k=1..p} \{\lambda_k |z^k(x') - z^k(x)|\}. \quad (2)$$

with $\lambda \in \left\{ \lambda \in \mathbb{R}^p : \lambda_k > 0 \text{ and } \sum_{k=1}^p \lambda_k = 1 \right\}$.

The λ vector corresponds to a search direction in the objective space, and enables to diversify the research. The best nearby configuration x^* , to replace the current configuration x , is determined using the following equation:

$$\text{eval}(x, x^*) = \max_{x' \in \mathcal{N}(x)} \text{eval}(x, x'). \quad (3)$$

If multiple configurations check the equation, the algorithm chooses one randomly.

d) Management of the tabu list: Every time a movement is applied to go from the current solution x , to the neighbor solution x' . To avoid cycling cases in which we would come back to x and oscillate between x' and x , the indication of the attribute of the movement is registered in a tabu list. So, the inverse movement (which corresponds on the way back to the departure configuration) is forbidden for a certain number of subsequent moves.

e) Random Walk Tabu Search (RWTS) algorithm: Contrary to the basic tabu search algorithm, where the diversification is ensured only by the tabu list, random walk tabu search algorithm consists of realizing from time to time a move which is no more guided by the evaluation function and then constitutes a diversification diagram.

At every iteration of RWTS algorithm, a real value $rw \in [0, 1]$ is randomly generated. Let us put $q \in [0, 1]$ the value threshold, then, if $rw > q$ the algorithm will select the best movement, otherwise, the algorithm will make a feasible random movement. RWTS algorithm can be described as follows:

Algorithm 1 RWTS algorithm.

Require: Number of iterations $nb - iter$, random threshold q .

Ensure: A set ND of potentially non-dominated solutions.

- 1: Randomly select an initial solution x , according to the "RK" encoding. $ND \leftarrow \{x\}$.
 - 2: **for** $i = 0$ **to** $nb - iter$ **do**
 - 3: Generate a random value $rw \in [0, 1]$.
 - 4: **if** $rw \leq q$ **then**
 - 5: choose a random allowed move j^* .
 - 6: **else**
 - 7: Choose the best allowed move j^* .
 - 8: Update the tabu list with j^* .
 - 9: Perform the chosen move j^* in x .
 - 10: Update the set of non-dominated solutions with x .
 - 11: **end if**
 - 12: **end for**
-

C. 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 approach which is compared to the extended multi-objective branch-and-bound (MOBB) method (a submitted paper). The instance $pWDPn - m$ 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. A tabu list length is fixed to 15 for all the instances and the threshold value for random walk q is set to 0,15. The algorithms have been implemented in matrix laboratory (Matlab R2009a) and each instance is re-run ten times with each algorithm using a Pentium PC

TABLE I
RESULTS OF THE RWTS METHOD

| Instances | hypervolume (%) | CPU(t) |
|-----------|-----------------|--------|
| 2WDP5-3 | 100 | 0.01 |
| 2WDP7-3 | 100 | 0.02 |
| 2WDP8-5 | 100 | 0.03 |
| 2WDP10-3 | 100 | 0.05 |
| 2WDP10-5 | 97.41 | 0.59 |
| 2WDP15-3 | 96.93 | 0.83 |
| 2WDP20-3 | 80.06 | 0.94 |
| 2WDP20-7 | 73.37 | 1.09 |
| 2WDP25-3 | 74.72 | 1.20 |
| 2WDP30-3 | 65.46 | 2.71 |
| 2WDP30-9 | 74.58 | 4.89 |
| 2WDP35-3 | 74.70 | 8.77 |
| 2WDP40-3 | 88.10 | 28.98 |
| 2WDP45-3 | 61.52 | 43.55 |
| 2WDP50-3 | 63.77 | 127.85 |

with dual core processor, FSB 800 Mb, DDR1 2 Go in Windows operating system. The approximate algorithms have been evaluated according to two measures of quality of \hat{E} (approximate set of efficient solutions):

- 1) Hypervolume metric [12]: it is based on calculating the volume (in the objective space) covered by members of the obtained non-dominated set of solutions.
- 2) Computational time represented by CPU(t) and measured in seconds.

Experimental results are provided in Table I. They show that the RWTS method is one of the most successful variants of the TS method. In particular, it exploits partially the search space. Thus for the MOWDP case, no factor guarantees that the search space is completely explored. This influences on the set of solutions that will not be always efficient (potentially efficient solutions). Indeed evaluations of approximate solutions obtained by RWTS, show that they are often far from the Pareto optimal front. Furthermore, the number of iterations and the value of the random walk influence the results a lot.

IV. CONCLUSION

In this paper we have adapted a Multi-Objective Random Walk Tabu Search (MORWTS) method for the Winner Determination Problem of combinatorial auctions in order to make it possible of handling more than one constraint. Evaluations of the metaheuristic of Random Walk Tabu Search show that they are often far from the Pareto optimal front.

In future works, we propose to develop other mechanisms of diversification and intensification more successful than the RWTS and to hybridize this metaheuristic with an exact method to improve the quality of solutions.

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, pp. 39–46, 2000.
- [2] Bean, J.C., "Genetics and random keys for sequencing and optimization," ORSA Journal of Computing, Vol 6, n2, pp. 154–160, 1994.
- [3] 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.
- [4] 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.

- [5] M. Ehrgott, Multicriteria Optimization (2. ed.). Springer 2005, isbn 978-3-540-21398-72.
- [6] Friedman, Lawrence, "A Competitive Bidding Strategy," Operations Research, 4, 104–112, 1956.
- [7] K. Florios, G. Mavrotas and D. Diakoulaki, "Solving multiobjective, multiconstraint knapsack problems using mathematical programming and evolutionary algorithms," European Journal of Operational Research, 14–21 (2010).
- [8] Glover, F., "Tabu Search Part II," ORSA Journal on Computing 2: 1, 4–32 (1990).
- [9] 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).
- [10] A. Holland, B. O'sullivan, "Towards Fast Vickrey Pricing using Constraint Programming," Artificial Intelligence Review, Vol 21, n 3-4/ June, pp: 335–352, 2004.
- [11] H. Hoos, C. Boutilier, "Solving combinatorial auctions using stochastic local search," in: Proceedings of the 17th national conference on artificial intelligence, pp. 22–29, 2000.
- [12] Knowles J., Thiele L. and E. Zitzler E., "A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers," TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, February 2006.
- [13] 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, pp.1–12, 2000.
- [14] Rothkopf, H. Michael, "A Model of Rational Competitive Bidding," Management Science, 15, 362–372 (1969).
- [15] M.H. Rothkopf, A. Pekee and M. Ronald, "Computationally manageable combinatorial auctions," Management Science, Vol. 44, No. 8, pp. 1131-1147, 1998.
- [16] T. Sandholm, S. Suri, "Improved Optimal Algorithm for Combinatorial Auctions and Generalizations," in Proceedings of the 17th national conference on artificial intelligence, pp. 9066–97 (2000).
- [17] 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, pp. 1102–1108, 2001.
- [18] T. Sandholm, "Optimal Winner Determination Algorithms," in P. Cramton et al. (ed.), Combinatorial Auctions, MIT Press (2006).
- [19] Steuer R.E., Multiple criteria optimization: Theory, Computation, and Application, John Wiley & Sons 1986.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US