iAuctionMaker: A decision support tool for mixed bundling

Antonio Reyes-Moro[1], Juan A. Rodríguez-Aguilar[2]

[1]iSOCO Lab. Intelligent Software Componets S.A
c\ Alcalde Barnils 64/68 A 08190 Sant Cugat del Valles, Barcelona, Spain
antonio.reyes@isoco.es
[2] Institut d'Investigació en Intel.ligència Artificial, IIIA
Consejo Superior de Investigaciones Científicas, CSIC
Campus de la Universitat Autònoma de Barcelona
08193 Bellaterra, Barcelona, Spain
jar@iiia.csic.es

Abstract. This paper presents iAuctionMaker as a novel tool that serves as a decision support for e-sourcing professionals on their pursuing of auction optimisation. Given a set of items to auction, iAuctionMaker helps an auctioneer determine how to separate items into promising bundles that are likely to produce better outcomes than the bundle of items as a whole. Promising bundles are those that satisfy certain properties believed to be present in competitive sourcing scenarios. These properties are defined by e-sourcing professionals and capture their experience and knowledge in the domain. iAuctionMaker models this knowledge as constraints to be satisfied by any bundle, and implements an optimisation algorithm to find the bundles that maximize satisfaction. Experimental results are shown to demonstrate the applicability of the approach. Case studies are presented to demonstrate that iAuctionMaker improves current e-sourcing practices and provides an alternative to combinatorial scenarios whose complexity hinders their application in actual-world sourcing scenarios.

1 Introduction

The negotiation scenario considered in this paper starts out with a *buyer* requiring to acquiring a set of items (be them either products or services). The buyer will negotiate the price and conditions[1] of each item by means of one or more on-line reverse auctions [16]. A set of providers will be invited to bid under certain auction rules that include bidding and winning rules. The auction is expected to allocate to some (all) of these providers some (all) of the items at auction.

---

[1] Notice that we consider multi-unit, multi-attribute items.

A common industrial scenario involves multiple goods or services to be purchased as a whole with the intention of benefiting from volume-based discounts. One may think, for example, of demand aggregation from different companies. The winner gets huge business and the losers get nothing, a simple and well-known strategy for lowering the price.

Unfortunately, things are not so simple. Maybe there is just one single provider that can provide everything so, if auctioning a single bundle, he will not face difficulties in getting the business at the price he quotes. Maybe it is not acceptable that (as providers bid for the whole thing), they lower the price for company-A product at the expense of increasing the price for company-B product (and consequently deal with B complaining about why are they buying more expensive than last year).

Therefore, the question is: should a seller who wants to maximise his revenue conduct separate auctions, one for each of several objects, or should he conduct a single auction for the entire bundle, or should he group items into bundles and conduct several auctions?

Efforts and tools have been developed to answer this problem. The general procedure is to allow flexibility in bidding by allowing providers to bid over combinations of items according to their preferences. I.e., give providers a way to state that they will offer a better price if there is a guarantee that they will get a certain amount of the business. This mechanism is known as combinatorial bidding [10] and has been widely studied in literature as an optimum artifact to maximize results.

To achieve coherent and practical results from a reverse combinatorial auction it is a must to introduce constraints that sacrifices mathematical optimality of the winning set in favour of obtaining realizable and practical outcomes [8][10][17]; (it is unnatural to have 40 different winners, for example, so it will be convenient to limit the amount of winners and state a lower bound of the amount of business they can get).

Unfortunately, combinatorial bidding capabilities are rarely found on commercial systems[2] [4] [15], and yet there is a major problem that prevents the practical application of combinatorial auctions: complexity. Bidding in a combinatorial auction requires accurate knowledge and understanding of the auction's dynamics in order to decide what is the next bid to place [14], [18]. Moreover, the constraints imposed to determine winners make winning rules complex.

The conclusion is that the practical application of the above methods has been limited to very controlled and specific environment (e.g. [5]).

To overcome this situation, e-sourcing professionals usually follow an alternate approach: based on market real data and knowledge, the whole bundle is divided into separate auctions where the appropriate providers are invited and where certain properties are satisfied. These properties model the expertise of e-sourcing specialists in the form of *rules of thumb*, and their applicability they believe can turn into interesting benefits [6].

---

[2] To develop a commercial system that allows fully flexible wining rule configuration to cope with real situations is costly.

Each auction is a simple reverse auction supported by the majority of existing commercial on-line auction platforms, and their execution present none of the complexities previously discussed. And the outcome of each auction is highly promising, as they have been designed to verify some criteria known to maximise savings.

It is interesting to state that theoretically, this methodology is expected to produce not as good results as a full combinatorial auction as it is likely that hidden synergies or interesting market situations are left out and unexploited. In spite of that, we believe that the former assert is true if and only we assume the ideal situation where everybody knows and controls the combinatorial auction mechanisms. Such a supposition is, for the best of our experience, never satisfied in practice and, consequently, not only it cannot lead to the desired result, but even produce catastrophic outcomes.

Nevertheless, in this methodology, the core process that is responsible for the success of the e-sourcing event is obviously the process of determining the grouping of items into bundles to be auctioned. Expertise and market knowledge are key factors, but in many situations the number of lines (100s or more) and the number of providers (20s or more) makes the problem intractable. These difficulties make desirable to count on a tool to aid e-sourcing professionals at this stage. We have called this tool iAuctionMaker.

iAuctionMaker is a decision support tool that assists an auctioneer in defining the *ideal* bundles by declaring a list of pre-existing constraints that can be tuned and prioritised according to his preferences. iAuctionMaker solves the problem of finding the bundles that maximizes the satisfaction of these constraints.

Although the bundling problem has been previously addressed in the literature we observed that it has mostly focused on the issue of whether a seller ought to sell items separately or as a bundle and to determine the price of the bundle(s) to be sold. In general, the bundling literature has evolved from early works considering a single seller bundling two goods [1] to works considering the analysis of a monopolist bundling multiple goods [3], to more recent works spurred by the advent of the Internet that contemplate competition of multiple sellers [2] [12]. As to multi-seller markets, in [12] the authors propose and analyse a bundling model to set both price and bundle composition in which a seller is not considered in isolation but in a market scenario wherein additional sellers compete to offer their bundles.

iAuctionMaker takes a different stance. We depart from a market scenario in which a single buyer aims at acquiring a bundle of multi-attribute goods. Unlike traditional approaches, it is not our aim to decide whether the buyer ought to purchase the goods separately or as a bundle, along with an appropriate pricing strategy. Our goal is to produce a bundle composition for a buyer that leads to clusters of providers (bidders) exhibiting high degrees of competitiveness, while at the same time satisfying the buyer's preferences (modelled as a collection of constraints). Furthermore, we expect that the partitioning of the whole bundle of items also benefits bidders since they are expected to address the bid construction problem [14] [18] for smaller bundles (less goods and competitors).

2. Problem definition

The formal formulation of the problem is the following:

- $I = \{I_1, I_2, ... I_n\}$ is a finite set of n items representing the goods or services to be purchased.
- $P = \{P_1, P_2, ... P_m\}$ is a finite set of m providers.
- $A = \{A_1, A_2, \ldots A_o\}$ where each element of A is a function $A{:}2^I \rightarrow R$ that models a property or observation of a subset of I representing a bundle. These properties might be various (number of providers, number of lines, lot volume, etc.) and come from different sources (previous provider behaviour, a preliminary RFQ[3], provider and item characterization, etc, etc.)
- $C = \{c_1, c_2, \ldots, c_r\}$. Each $c_i \in C$ is a constraint defined as a tuple $<A_i, S_i, w_i>$ with the following meaning: A_i is the bundle property to be evaluated, S_c: $R \rightarrow [0\ldots1]$ is a scoring function (formally defined in section 3) that expresses the satisfaction degree for A_i (0 indicates no satisfaction at all; and 1 indicates maximum satisfaction); and $w \in R^+$ expresses the relative weight of the constraint.

The objective is to find $L=\{L_1, L_2, \ldots L_q\}$ a set partition of I that maximises the following expression:

$$S(L) = \sum_{L_k \in L} \frac{\dfrac{\sum_{c \in C} S_c(A_c(L_k)) \cdot w_c}{\sum_{c \in C} w_c}}{|\, L \,|} \qquad \textbf{(1)}$$

subject to: (1) $L_i \cap L_j = \varnothing \quad \forall L_i, L_j \in L; L_i \neq L_j$; and (2) $L_1 \cup L_2 \cup \ldots L_l = L$.

This problem is a particular instance of the set partition problem [13], which is known to be NP complete [7].

3. Solution

To solve the problem formulated in the previous section, we first have to define a lot's utility theory (i.e, define S_c and design an optimisation algorithm.

3.1 MAUT-based bundle evaluation

The method used in *iAuctionMaker* for scoring a lot is based on Multi attribute utility theory [11], since lot's utility or *goodness* can be evaluated by the degree of satisfaction of a list of lot's attributes for a given user's preference and importance.

---

[3] Request For Quotation

To model preferences and importance we have previously defined a set of constraints. Each constraint $c_i = <A_i, S_i, w_i>$ evaluates some property A_i of a lot by means of a scoring function S_i. To define S_i, we have followed the guidelines proposed in [15] where membership functions are studied to *intuitively* model human preferences. With these considerations in mind, $Sc: R \rightarrow [0\ldots1]$ is defined as follows:

$$
\begin{array}{ll}
S_c(p) = \alpha+(b-p)\cdot\beta & p\in [a, b] \\
S_c(p) = \max \{S_c(a)-(a-p)\cdot\delta , 0) & p<a \\
S_c(p) = \max \{S_c(b)-(p-b)\cdot\phi, 0) & p>b
\end{array}
\qquad (2)
$$

where: (i) *a>b* define the preferred range *[a…b]* of values; (ii) $sl \in\{ANY,LIB,MIB,LIBC,MIBC\}$ defines the preference slope of S_c; (iii) $mh \in\{YES, NO\}$. if *mh= YES* then values are not accepted out of the preference interval (they will score *0)*; and (iv) α, β, δ, ϕ, depend on the value of *sl* and *mh* and are calculated as shown in table 1.

Table 1. Values of α, β, δ, φ

	ANY	LIB	MIB	LIBC	MIBC
No	$\alpha = 1$ $\beta = 0$ $\delta = \gamma\cdot\frac{(1-\eta)}{(b-a)}$ $\phi = \delta$	$\alpha = \eta$ $\beta = \frac{(1-\eta)}{(b-a)}$ $\delta = 0$ $\phi = \gamma\cdot\beta$	$\alpha = 1$ $\beta = -\frac{(1-\eta)}{(b-a)}$ $\delta = \gamma\cdot\beta$ $\phi = 0$	$\alpha = \eta$ $\beta = \frac{(1-\eta)}{(b-a)}$ $\delta = -\gamma\cdot\beta$ $\phi = -\gamma\cdot\beta$	$\alpha = 1$ $\beta = -\frac{(1-\eta)}{(b-a)}$ $\delta = \gamma\cdot\beta$ $\phi = \gamma\cdot\beta$
Yes	$\alpha = 1$ $\beta = 0$ $\delta = +\infty$ $\phi = +\infty$	$\alpha = \eta$ $\beta = \frac{(1-\eta)}{(b-a)}$ $\delta = 0$ $\phi = +\infty$	$\alpha = 1$ $\beta = -\frac{(1-\eta)}{(b-a)}$ $\delta = +\infty$ $\phi = 0$	$\alpha = \eta$ $\beta = \frac{(1-\eta)}{(b-a)}$ $\delta = +\infty$ $\phi = +\infty$	$\alpha = 1$ $\beta = -\frac{(1-\eta)}{(b-a)}$ $\delta = +\infty$ $\phi = +\infty$

Intuitively, when a value falls within the preference limits, is given a value that is at least η, which models the limit of satisfaction. Depending on the slope *sl* the scoring progress from η to *1* as we move through *[a...b]*. When a value falls outside the preference limits, it receives a score that will be progress from η or *1* to *0* depending on how ‘close enough’ is the value to the preferred side. By ‘close enough’ we consider values within a neighbourhood of the interval (computed as a percentage $1/\gamma$ of the interval length). Figure 1 shows the scoring for *a=10, b= 20, mh=NO,* $\eta = 0.3$, $\gamma = 2$.

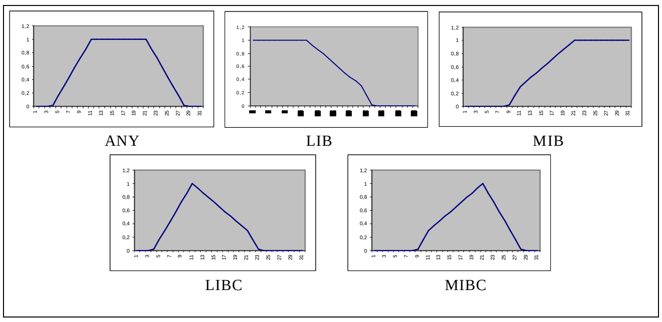

Fig. 1. Scoring functions

- ANY is used when we just want that a property *p* of the lot to fall between [a…b].
- LIB states that we will consider the constraint satisfied if $p\leq b$, and that we will consider full satisfaction when $p\leq a$.
- MIB states that we will consider the constraint satisfied if $p \geq a$, and that we will consider full satisfaction when $p\geq b$.
- LIBC means that a constraint is satisfied if $p\in[a\ldots b]$; full satisfaction will be considered when $p=a$; and minimum satisfaction when $p=b$.

So far we have explained how to obtain the degree of constraint satisfaction for a lot by means of a scoring function. For a set of constraints, scoring functions are then weighted to obtain the overall lot scoring as shown:

$$\frac{\sum_{c\in C} S_c(A_c(L_k)) \cdot w_c}{\sum_{c\in C} w_c} \qquad (3)$$

By setting each w_c accordingly, the user is allowed to state satisfaction preference among constraints.

3.2 Optimisation algorithm

The optimisation algorithm implemented for *iAuctionMaker* falls in the category of random probabilistic methods. Well-known exponents of these are genetics algorithms and neighbourhood search [9]. The main idea within these methods is to start by a number of initial solutions and implement a try and test procedure to discover better solutions. The try and test procedure, or search, is basically an iterative random change of these solutions until the best exponent converge to a local optima or a maximum number of try-and-test cycles are done. The search is directed by favouring changes that follow certain heuristics.

We have devised our own probabilistic search procedure which is mainly a neighbourhood search directed by heuristics. Whether it is a new procedure, extension, optimisation, or rediscovering of existing search algorithms is, at the current stage of the work, of no importance to us and is beyond the scope and interest of this paper. Literature is full of claimed-as-new algorithms that are actually rediscovers of existing algorithms previously applied to different problem domains.

The reason why we have applied such an algorithm is because random search methods are usually fast, perform relatively well and are easy to implement. Also, random search methods are usually independent from the objective function, and their performance do not heavily rely on exploiting problem characteristics and lower bounds identification. This is of special interest to us, as we expect the number and type of the constraints to be highly determined by the final user and his application domain (e.g. food, transportation, indirect materials, etc.). Even if we are able to model the problem as an integer program, the introduction of new constraints will force us to study the feasibility of the current model and to change it accordingly. Employing a branch-and-bound procedure would require a considerable amount of expert knowledge and effort to tune the heuristics function each time new constraints are changed or refined, in order to maintain algorithm performance.

The main criticism to approximate random search is sub-optimality. In our case, this is not critical as the solution is just an intermediate phase in our process that will terminate with the execution of an auction, the real outcome of which is unpredictable; consequently, it is inherently wrong that the best possible lot set will result in the best possible of the results. The algorithm can be outlined as follows:

```
L = ∅
for each Ii ∈ I
    create Li = {Ii}
solution = copy_of(L)
while (convergence is not reached)
  randomly pick Li ∈ L; with probability
  inversely proportional to S({Lj})
  randomly pick Lj ∈ L ∪ {∅} ; Lj ≠ Li with uniform
  probability
  randomly pick Ik ∈ Li with probability
  inversely proportional to S({Li}) – S({Li – Ik})
  Li = Li – Ik
  Lj = Lj ∪ Ik
  if Li = ∅ then L = L - Li
  if Lj ∉ L then L = L ∪ Lj
  if S(solution)<S(L) then
        solution = copy_of(L)
return solution
```

The previous procedure is explained as follows: An initial solution is built by considering each item to be auctioned in isolation. Then we enter an iterative phase where we randomly select a bundle. The bundle is selected implementing a roulette wheel [9] where the chances of each lot are inversely proportional to its constraints satisfaction value (i.e. *bad* bundles will be selected more often, in an attempt to transform them into good ones) From this bundle, we select the item that is most probably causing the low satisfaction value of the bundle. To identify *bad* items we calculate the difference between the constraint satisfaction degree of the bundle with and without the item. Once an item has been selected, we remove it from the bundle, and next we randomly select a bundle (or a new bundle) where to add the item. The heuristic beneath this procedure is to penalise bad combinations of items. The random nature of the neighbourhood search prevents excessive falling into local optima.

The basic procedure has been enhanced with typical techniques to allow the algorithm to converge faster:

- Implement a backtracking procedure that allow to backtrack to a previous state if the current solution differs more than certain percentage from the best solution known so far.
- Avoid useless operations (i.e., changes that are known not to lead to better solutions).

Implement an alternate *change* operator that joins two bundles. A probabilistic factor is used to randomly determine which operator to apply at each step of the algorithm.

3.3 Implementation

The core of iAuctionMaker is implemented in *java* and verifies the XML and J2EE standards, which simplify the implementation of multiple user interfaces (web-based, for example) as well as its integration with existing applications. The object model of iAuctionMaker works with *interface* declarations of *constraints* and *properties* objects. This allows the easy extension of the system by *implementations* of client-custom constraints. As mentioned in the previous section, the random search procedure implemented will not need of reformulation if new constraints are added. This will optimise product customisation, without incurring in costly AI expertise and algorithm-refactoring.

4. Results

This section will present two experimental outcomes of *iAuctionMaker*. The first results aims to demonstrate that the search procedure developed for the tool performs satisfactorily. The second results present the commercial application of *iAuctionMaker* to two real sourcing scenarios.

In order to validate the optimisation algorithm proposed we performed two experimental tests. The first one aims at measuring the optimality of the algorithm when compared to a complete search procedure. The second experiment aims at measuring the correctness and applicability of the algorithm for large instances of the problem.

For the first experiment we generated 1000 random instances of a problem consisting of 11 lines and 11 providers[4]. The problem instance considers that all providers are capable of providing all lines. The price provider *j* offers for line *j* is randomly determined with uniform probability between [0..10]. We considered four constraints: *volume aggregation, best bidder presence, bids' variability* and *number of providers.*

Each problem is solved optimally by a brute force search procedure[5], i.e., all possible solutions are generated. Then, each problem is solved with our algorithm and results are compared. 7 rounds are executed, each varying the algorithm termination condition, which controls the number of iterations (different solutions) explored by our algorithm, for each round. Table 2 shows the results obtained.

Results suggest that *iAuctionMaker* seems to perform accordingly with the objectives, i.e., optimal solutions are found with considerable less search effort and, maybe more important, the relative difference between the optimal and the sub-optimal solution found is more than acceptable.

---

[4] An affordable size to be solved by a brute force search algorithm.

[5] The total number of different solutions for this problem size is 678570 (see Bell Numbers for further information).

Table 2. Results obtained for experiment 1

% of search effort	% of problems solved optimally	% variation between optimal and best solution found
0.2	23	4.11
1.05	56	2.87
1.93	69	2.3
8.36	85	1.63
16.22	89	1.4
77	93	1.53
150	96	1.36

The second experiment consisted in solving problems for which we know the best solution in advance. Problems are generated by randomly separating items into 2, 4, 5, 10, or 20 bundles. Each bundle is only provided by a certain group of providers. Two constraints were considered, *number of providers*, and *bundle volume.* Hence, the best possible solution is to find such bundles. The problem size was fixed to 100 lines and 20 providers, which is a problem size larger than the ones to be solved in our case studies. Table 3 shows the optimality evolution as we increase the search effort.

Table 3. Results obtained for experiment 1

% of problems solved optimally	Average mean solution time (seconds)
17%	0.16
54%	0.49
70%	0.9
87%	3.7
89%	6.8
98%	26.8
99%	65.9

Results suggest that the performance indicators observed for experiment 1 can also be obtained for large instances of problems and at affordable cost.

In conclusion, these experiments suggest that we can be fairy confident on the *goodness* of *iAuctionMaker* random optimisation procedure to be applied in real scenarios.

4.1 Case study: Electricity purchase

The first scenario studied the initials offers received from 5 South-Europe electricity companies to power a total of 20 manufacturing facilities in Spain that belong to the same company. The plants are all of similar power consumption and are geographically distributed across the country. For each location, bidders decide whether to bid or not. The bid presented stands for the average price of the Kilowatt, according to last year consumption records. The scenario is therefore translated into a problem consisting of 20 lines and 5 providers. Table 4 presents the initial offer given by each provider for each location[6].

Table 4. Electricity market data

Locations/Providers	P1	P2	P3	P4	P5
BAD	5,545	5,836	5,415	5,493	5,329
BEZ	5,313	5,528	5,384	5,269	5,028
CAR	5,599	5,896	5,339	5,604	5,311
COD	5,495	5,91	5,247	5,489	5,195
COR	4,417	4,831	4,484	4,444	
DULC	5,883	6,296	5,761	5,978	
ESP	5,496	5,881	5,431	5,483	5,361
GEN	5,129	5,402	4,886	5,211	4,903
GRA	5,317	5,739	5,24	5,264	5,13
GUA	5,366	5,119		5,347	5,141
PER	5,219	5,583	5,112	5,265	5,151
PLANT	5,494	5,988	5,606	5,381	5,261
RAI	5,724	6,353	5,727	5,743	5,561
RIB	5,795	6,021	5,575	6,033	5,423
RON	5,803	6,204	5,774	5,869	5,498
SES	5,31	5,831	5,422	5,289	5,134
SEV	5,182		5,083	5,238	5,101
VOÑÑ	5,345	5,745	5,452	5,212	5,146
ZAM	5,312	5,634	5,067	5,439	5,093
FRI	5,258	5,428	5,005	5,209	5,035

The company's sourcing professionals know that in order to achieve savings, it will be of interest to group facilities rather than auction each facility in isolation. However, some of the bidders are new, small companies (the Spanish electricity market was liberalized short ago) which are geographically specialized and are likely to bid aggressively for facilities in their area, whereas unable to compete for others.

To model this knowledge into *iAuctionMaker* three constraints were given:

1. The bigger the bundle (in price terms[7]), the better.
2. The best possible offer for the whole lot by a single provider must be at most 1% worse than the offer obtained by selecting the best offer per location.
3. Ideally there should be 3 providers whose offers for the whole lot differ less than 3%.

Constraint 1 tried to make bigger lots, where there exists place for competitiveness (constraint 3). To prevent missing very competitive offers for certain locations, constraint 2 is given.

[6] Bidders are first invited to an RFQ where to place their first offer. (They do not know yet that an auction may later take place).

[7] All prices are in EURO.

Table 5. Modelling constraints

Constraint	A (Observable variable)	S_c[8]	w_c
c1	Total lot price, calculated as the sum of the mean offer for each line.	a = 20 b = 100	1
c2	Number of providers that are %1 from the optimal.	A = 1 b = 1	1
C3	Number of providers capable of offer the whole lot.	a = 2 b = 3	1

With these constraints, *iAuctionMaker* finds the 3-bundle distribution shown in *Figure 2*[9].

All three lots are quite similar and satisfy the desired properties:

- There are 3 providers close enough to compete.
- The amount of business is interesting.

In case of no auction activity, the risk is to purchase %0.47 more expensive than the current situation.

4.2 Case study: Transportation purchase

The second scenario studied the initials offers received from 16 transportation companies to deliver a company range of products to 81 destinations across Europe. A first round of RFQs where conducted to obtain initial price-matrix (destination by kg). Based on historical data, the price matrix was reduced to a single column representing the total cost to each destination for each bidder.

The application of *iAuctionMaker* to this scenario produced an interested outcome: it was not possible to find any *promising* lot to undergo an auction.

To obtain an explanation for this, *iAuctionMaker* was given the following two constraints:

- The bigger the lot, the better.
- The best bidder for the lot must be also the best bidder for each single destination in the lot.

In other words, with these two constrains *iAuctionMaker* was configure to identify the current winning set of providers and group destinations accordingly.

---

8 The rest of parameters are set as follows sl=MIB, mh=NO, η = 0.3, γ = 2. Refer to table 1.

9 This solution depicted is just one among others that scored identically. The company purchasing department evaluated them all and the final auction configuration (not shown here) was selected considering geographical distribution.

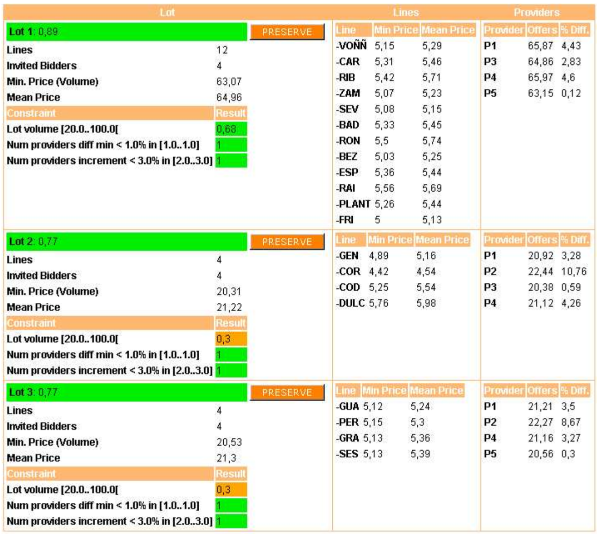

Fig. 2. iAuctionMaker results for the electricity problem

As seen in figure3, the solution obtained has 3 lots, each corresponding to 3 winners: *P9, P13 and P4.* Notice the difference in price between the winner and the immediate competitor (a minimum of 43% for Lot 3). This explains why there is no room for an auction. Obviously, the lots obtained correspond to a particular geographical distribution for which each winner is clearly specialised (*Lot 1* only contains locations in *Italy*, for example).

After obtaining these results, the company purchasing department verified that there was no mistake in the offers received and the negotiation ended after a second round of offers.

Conclusively, *iAuctionMaker* proved to be useful in assisting the user to identify scenarios where the application of an on-line auction will not produce clear benefits.

Fig. 3. iAuctionMaker results for the transportation problem

5. Conclusions

This paper has presented *iAuctionMaker as* a novel decision support tool for e-sourcing professionals. The motivation was to improve current e-sourcing procedures and provide an alternative to combinatorial or constraint bidding scenarios whose complexity prevent their application in real sourcing scenarios. The methods and algorithms developed were highly directed by software industry needs of efficiency and easy of extension and customisation. Experimental results showed promising results, which were later verified by successful application to real industry problems. Customers highly evaluated the tool and were satisfied with the results obtained.

Future work basically lays in the application of the tool to more real sourcing scenarios from various industries. This will provide us with useful feedback from e-sourcing professionals as well as to test new constraints obtained from their domains. Our goal is to provide an extensive library of *rules of thumb* that contains the expert knowledge of sourcing professionals.

References

[1] Adams, W. and Yellen, J. (1976). Commodity bundling and the burden of monopoly. Quartely Journal of Economics, 90:475-498, 1976.

[2] Bakos, Y. and Brynjolfsson. (1999). Bundling and competition on the internet: Aggregation strategies for information goods. Working paper series, MIT Sloan School of Management.

[3] Bakos, Y. and Brynjolfsson, E. (1998). Bundling information goods: Pricing, profits and efficiency. In D. Hurley, B. Kahin, and H. Varian, editors, The Economics of Digital Information Goods. MIT Press, Cambridge, Massachussetts.

[4] Combinenet. http://www.combinenet.com, 2004.

[5] Elmaghraby, W. and Keskinocak, P. (2002). Technology for Transportation Bidding at the Home Depot. To appear in C. Billington, T. Harrison, H. Lee, J. Neale, editors, The Practice of Supply Chain Management., Kluwer.

[6] Eppen, G. D., Hanson, W. A. (1991). Bundling- New Products, New Markets, Low Risk. Sloan Management Review 32(4): pp. 7-14.

[7] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guideto the Theory of NP-Completeness*. W. H. Freeman and Company, New York.

[8] Giovannucci, A. Rodríguez, J.A. Reyes, A. Noria, F.X. and Cerquides, J. (2004). *Towards automated procurement via agent-aware negotiation support*. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-agent Systems (to appear), New York City, July 19-24, 2004.

[9] GoldBerg, D.E. *Genetics Algorithms in Search, Optimisation & Machine Learning.* (1989) Addison-Wesley.

[10] Kalagnanam, J., Parkes D. (2003) Auctions, Bidding and Exchange Design. Chapter 10, in Supply Chain Analysis in the eBusiness Era, Edited by David Simchi-Levi, S. David Wu and Z. Max Shen,. (forthcoming)

[11] Keeny, R. L. and H. Raiffa. (1993). *Decision Making with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge, UK: Cambridge University Press.

[12] Kephart, J. O. and Fay, S. A. (2000). Competitive Bundling of Categorized Information Goods. EC'00, October 17-20, Minneapolis, Minnesota.

[13] Kreher, D.L. Stinson, D. R. (1998) *Combinatorial Algorithms: Generation, Enumeration, and Search*, CRC Press.

[14] Parkes, D. (2000) Optimal Auction Design for Agents with Hard Valuation Problems, In Agent Mediated Electronic Commerce II: Towards Next-Generation Agent-Based Electronic Commerce Systems (LNAI 1788), Moukas et al. (eds), Springer-Verlag.

[15] Reyes, A. Rodríguez J.A., López M., Cerquides J. and Gutierrez. D. (2003) *Embedding Decision Support in e-Sourcing Tools: Quotes, A Case Study.* Group Decision and Negotiation Journal, Vol. 12, pp 347-355.

[16] Sandholm, T., Suri, S., Gilpin, A., and Levine, D. (2002). Winner Determination in Combinatorial Auction Generalizations. In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), Bologna, Italy, July, pp. 69-76. .

[17] Sandholm, T. and Suri, S. (2001). Side Constraints and Non-Price Attributes in Markets. International Joint Conference on Artificial Intelligence (IJCAI), In Proceedings of the Workshop on Distributed Constraint Reasoning, Seattle, WA, August 4th.

[18] Song, J. and Regan, A. (2004). Approximation algorithms for the Bid Construction Problem in Combinatorial Auctions for the Procurement of Freight Transportation Contracts. In Proceedings of the 2004 TRB meeting.