Negotiation support in highly-constrained trading scenarios

Juan A. Rodríguez-Aguilar[1,2], Antonio Reyes-Moro[1], Andrea Giovanucci[1], Jesús Cerquides[1], and Francesc X. Noria[1]

[1] iSOCOLab
Intelligent Software Components, S. A.
Edificio Testa, C/ Alcalde Barnils, 64-68 A
08190 Sant Cugat del Vallès, Barcelona, Spain
{jar,toni,andrea,cerquide,fxn}@isoco.com
http://www.isoco.com

[2] Artificial Intelligence Research Institute, IIIA
Spanish Council for Scientific Research, CSIC
08193 Bellaterra, Barcelona, Spain.
jar@iiia.csic.es
http://www.iiia.csic.es

Abstract. Negotiation events in industrial procurement involving multiple, highly customisable goods pose serious challenges to buyers when trying to determine the best set of providers' offers. Typically, a buyer's decision involves a large variety of constraints that may involve attributes of a very same item as well as attributes of different, multiple items. In this paper we present the winner determination capabilities of *iBundler*[10], an agent-aware decision support service offered to buyers to help them determine the optimal bundle of received offers based on their constraints and preferences. In this way, buyers are relieved with the burden of solving too hard a problem, and thus concentrate on strategic issues. *iBundler* is intended as a negotiation service for buying agents and as a winner determination service for reverse combinatorial auctions with side constraints.

1 Introduction

One particular, key procurement activity carried out daily by all companies concerns the negotiation of both direct and indirect goods and services. Throughout negotiations the decision-making of buyers and providers appears as highly challenging and intricate because of an inherently high degree of uncertainty and the large number of parameters, variables, and constraints to take into account.

Consider the problem faced by a buyer when negotiating with providing agents. In a negotiation event involving multiple, highly customisable goods, buyers need to express relations and constraints between attributes of different items. On the other hand, it is common practice to buy different quantities of the very same product from different providers, either for safety reasons or because offer aggregation is needed to cope with high-volume demands. This introduces the need to express business constraints on the number of suppliers and the amount of business assigned to each of them. Not forgetting the provider side, suppliers may also impose constraints or conditions over their offers. Offers may be only valid if certain configurable attributes (f.i. quantity bought, delivery days) fall within some minimum/maximum values, and assembly or packing constraints need to be considered. Once the buyer collects offers, he is faced with the burden of determining the winning offers. The problem is essentially an extension of the combinatorial auction (CA) problem, which can be proven to be NP[11]. It would be desirable to relieve buyers from solving such a problem.

We have tried to make headway in this direction by deploying *iBundler*[10], a decision support service acting as a combinatorial negotiation solver (solving the winner determination problem) for both multi-attribute, multi-item, multi-unit negotiations and auctions. Thus, the service can be employed by both buyers and auctioneers in combinatorial negotiations and combinatorial reverse auctions [13] respectively. In this paper we introduce the formal model and the computational realisation of the decision problem solved by *iBundler*.

The paper is organised as follows. Section 2 introduces the market scenario where buyers and traders are to negotiate, along with the requirements and constraints they may need. Next, a formal model capturing such requirements is put forward to set the foundations for the decision problem. The actual computational realisation of our decision support for negotiations is thoroughly detailed in section 4. Finally, section 5 provides some hints on the expected performance of the service.

2 Market scenario

Although the application of combinatorial auctions (CA) to e-procurement scenarios (particularly reverse auctions) may be thought as straightforward, the fact is that there are multiple new elements that need to be taken into consideration. These are new requirements explained by the nature of the process itself. While in direct auctions, the items that are going to be sold are physically concrete (they do not allow configuration), in a negotiation event involving highly customisable goods, buyers need to express relations and constraints between attributes of different items. On the other hand, it is common practice to buy different quantities of the very same product from different providers, either for safety reasons or because offer aggregation is needed to cope with high-volume demands. This introduces the need to express constraints on the number of providers and the amount of business assigned to each of them. Not forgetting the provider side, providers may also impose constraints or conditions over their bids/offers. Offers may be only valid if certain configurable attributes (f.i. quantity bought, delivery days) fall within some minimum/maximum values, and assembly or packing constraints need to be considered.

Current CA reviewed do not model these features with the exception of[3, 12], where coordination and procurement constraints can be modelled. The rest of work focuses more on computational issues (CA is an NP-complete problem[11]) than in practical applications to e-procurement. Suppose that we are willing to buy 200 chairs (any colour/model is fine) for the opening of a new restaurant, and at that aim we employ an e-procurement solution that launches a reverse auction. If we employ a state-of-the-art CA solver, a possible resolution might be to buy 199 chairs from provider A and 1 chair from provider B, simply because it is 0.1% cheaper and it was not possible to specify that in case of buying from more than one provider a minimum of 20 chairs purchase is required. On the other hand the optimum solution might tell us to buy 50 blue chairs from provider A and 50 pink chairs from provider B. Why? Because although we had no preference over the chairs' colour, we could not specify that regarding the colour chosen all chairs must be of the same colour. Although simple, this example shows that without means of modeling these natural constraints, solutions obtained are seen as mathematically optimal, but unrealistic and with a lack of common sense, thus obscuring the power of decision support tools, and preventing the adoption of these technologies in actual-world settings.

Next we detail the capabilities required by buyers in the kind of negotiation scenario outlined above. The requirements below are intended to capture buyers' constraints and preferences and outline a powerful providers' bidding language:

Negotiate over multiple items. A negotiation event is usually started with the preparation of a request for proposal (RFQ) form. The RFQ form describes in detail the requirements (including attribute-values such as volume, quality specifications, dates as well as drawings and technical documentation) for the list of items (goods or services) defined by the negotiation event.

Offer aggregation. A specific item of the RFQ can be acquired from several providers simultaneously, either because not a single provider can provide with the requested quantity at requested conditions or because buyers explicit constraints (see below).

Business sharing constraints. Buyers might be interested to restrict the number of providers that will finally trade for a specific item of the RFQ, either for security or strategical reasons. It is also of usual practice to define the minimum amount of business that a provider may gain per item.

Constraints over single items. Every single item within an RFQ is described by a list of negotiable attributes. Since: a) there exists a degree of flexibility in specifying each of these attributes (i.e. several values are acceptable) and b) multiple offers referring the very same item can be finally accepted; buyers need to impose constraints over attribute values. An example of this can be the following: suppose that the deadline for the reception of certain item A is two weeks time. However, although items may arrive any day within two weeks, once the first units arrive, the rest of units might be required to arrive in no more than 2 days after.

Constraints over multiple items. In daily industrial procurement, it is common that accepting certain configuration for one item affects the configuration of a different item, for example, when dealing with product compatibilities. Also, buyers need to express constraints and relationship between attributes of different items of the RFQ.

Specification of providers' capacities. Buyers cannot risk to award contracts to providers whose production/servicing capabilities prevent them to deliver over-committed offers. At this aim, they must require to have providers' capacities per item declared. Analogously, next we detail the expressiveness of the bidding language required by providers. The features of the language below are intended to capture providing agents' constraints and preferences.

Multiple bids over each item. Providers might be interested in offering alternate conditions/configurations for a same good, i.e., offering alternatives for a same request. A common situation is to offer volume-based discounts. This means that a provider submits several offers and each offer only applies for a minimum (maximum) number of units.

Combinatorial offers. Economy efficiency is enhanced if providers are allowed to offer (bid on) combination of goods. They might lower the price, or improve service assets if they achieve to get more business.

Multi-unit offering. Each provider specifies the minimum (maximum) amount of units to be accepted in a contract.

Homogeneous combinatorial offers. Combinatorial offering may produce inefficiencies when combined with multi-unit offering. Thus a provider may wind up with an award of a small number of units for a certain item, and a large number of units for a different item, being both part of the very same offer (e.g. 10 chairs and 200 tables). It is desirable for providers to be able to specify homogeneity with respect to the number of units for complementary items.

Packing constraints. Packing units are also a constraint, in the sense that it is not possible to serve an arbitrary number of units (e.g. a provider cannot sell 27 units to a buyer because.his items come in 25-unit packages). Thus providers require to be capable of specifying the size of packing units.

Complementary and exclusive offers. Providers usually submit XOR bids, i.e., exclusive offers that cannot be simultaneously accepted. Also, there may exist the need that an offer is selected only if another offer is also selected. We refer to this situation as an AND bid. This type of bids allows to express volume-based discounts. For example, when pricing is expressed as a combination of base price and volume-based price (e.g. first 1000 units at $2.5 p.u. and then $2 each).

Obviously, many more constraints regarding pricing and quantity can be considered here. But we believe these faithfully address the nature of the problem. Actually, *iBundler* has been applied to scenarios where some of these constraints do not apply while additional constraints needed to be considered. This was the case of a virtual shopping assistant, an agent that was able to aggregate several on-line supermarkets and optimize the shopping basket. To do so, it was necessary to model the fact that delivery cost depends on the amount of money spent at each supermarket.

3 Formal model

In this section we provide a formal model of the problem faced by the buyer (auctioneer) based on the description in section 2. Prior to the formal definition, some definitions are in place.

Definition 1 (Items). *The buyer (auctioneer) has a vector of items $\Lambda = \langle \lambda_1, \ldots, \lambda_m \rangle$ that he wishes to obtain. He specifies how many units of each item he wants $U = \langle u_1, \ldots, u_m \rangle, u_i \in \mathbb{R}^+$. He also specifies the minimum percentage of units of each item $M = \langle m_1, \ldots, m_m \rangle, m_i \in [0,1]$, and the maximum percentage of units of each item $\bar{M} = \langle \bar{m}_1, \ldots, \bar{m}_m \rangle, \bar{m}_i \in [0,1], \bar{m}_i \geq m_i$, that can be allocated to a single seller. Furthermore, he specifies the minimum number of sellers $S = \langle s_1, \ldots, s_m \rangle, s_i \in \mathbb{N}$, and the maximum number of sellers $\bar{S} = \langle \bar{s}_1, \ldots, \bar{s}_m \rangle, \bar{s}_i \in \mathbb{N}, \bar{s}_i \geq s_i$, that can have simultaneously allocated each item. Finally, a tuple of weights $W = \langle w_1, \ldots, w_m \rangle, 0 \leq w_i \leq 1$, contains the degree of importance assigned by the buyer to each item.*

Definition 2 (Item attributes). *Given an item $\lambda_i \in \Lambda$, let $\langle a_{i_1}, \ldots, a_{i_k} \rangle$ denote its attributes.*

Definition 3 (Sellers' capacities). *Let $\Pi = \langle \pi_1, \ldots, \pi_r \rangle$ be a tuple of providers. Given a provider $\pi_i \in \Pi$ the tuple $C^i = \langle c^i_1, \ldots, c^i_m \rangle$ stands for the minimum capacity of the seller, namely the minimum number of units of each item that the seller is capable of serving. Analogously, the tuple $\bar{C}^i = \langle \bar{c}^i_1, \ldots, \bar{c}^i_m \rangle$ stands for the maximum capacity of the seller, i.e. the maximum number of units of each item that the seller is capable of providing.*

Definition 4 (Bid). *The providers in Π submit a tuple of bids $B = \langle B^1, \ldots, B^n \rangle$. A bid is a tuple $B^j = \langle \Delta^j, P^j, M^j, \bar{M}^j, D^j \rangle$ where $\Delta^j = \langle \Delta^j_1, \ldots, \Delta^j_m \rangle$ are tuples of bid values per item, where $\Delta^j_i = \langle \delta^j_{i_1}, \ldots, \delta^j_{i_k} \rangle \in \mathbb{R}^k, 1 \leq i \leq m$, assigns values to the attributes of item λ_i; $P^j = \langle p^j_1, \ldots, p^j_m \rangle, p^j_i \in \mathbb{R}^+$, are the unitary prices per item; $M^j = \langle m^j_1, \ldots, m^j_m \rangle, m^j_i \in \mathbb{R}^+$, is the minimum number of units per item offered by the bid; $\bar{M}^j = \langle \bar{m}^j_1, \ldots, \bar{m}^j_m \rangle, \bar{m}^j_i \mathbb{R}^+, \bar{m}^j_i \geq m^j_i$, is the maximum number of units of each item offered by the bid; and $D^j = \langle d^j_1, \ldots, d^j_m \rangle$ are the* bucket *or* batch *increments in units for each item ranging from the minimum number of units offered up to the maximum number of units.*

Given a bid $B^j \in B$, we say that B^j does not offer item $\lambda_i \in \Lambda$ iff $m^j_i = \bar{m}^j_i = 0$.

In order to model homogeneity constraints, we define a function $h : B \to 2^\Lambda$. Given a bid $B^j \in B$, $h(B^j) = \{\lambda_{j_1}, \ldots, \lambda_{j_k}\}$ indicates that the bid is homogeneous with respect to the items in $h(B^j)$. In other words, if the buyer (auctioneer) picks up bid B^j the number of units allocated for the items in $h(B^j)$ must be equal.

Furthermore, in order to relate sellers to their bids we define function $\rho : \Pi \times B \to \{0,1\}$ such that $\rho(\pi_i, B^j) = 1$ indicates that seller π_i is the owner of bid B^j. This function satisfies the following properties:

– $\forall B^j \in B \;\; \exists \pi_i \in \Pi$ such that $\rho(\pi_i, B^j) = 1$; and
– given a bid $B^j \in B$ if $\exists \pi_i, \pi_k \in \Pi$ such that $\rho(\pi_i, B^j) = 1$ and $\rho(\pi_k, B^j) = 1$ then $\pi_i = \pi_k$.

The conditions above impose that each bid belongs to a single seller.

Definition 5 (XOR bids). *Let* $xor : 2^B \rightarrow \{0, 1\}$ *be a function that defines whether a subset of bids must be considered as an XOR bid. Only bids owned by the very same seller can be part of an XOR bid. More formally* $xor(\mathcal{B}) = 1 \Rightarrow \exists!\ \pi \in \Pi$ *such that* $\rho(\pi, B^i) = \rho(\pi, B^j) = 1 \ \forall B^i, B^j \in \mathcal{B}, B^i \neq B^j$. *Thus, f.i. if* $\exists B^j, B^k \in B \ xor(\{B^j, B^k\}) = 1$ *both bids are mutually exclusive, and thus cannot be simultaneously selected by the buyer.*

Definition 6 (AND bids). *Let* $and : \cup_{i=1}^{n} B^i \rightarrow \{0, 1\}$ *be a function that defines whether an ordered tuple of bids must be considered as an AND bid. Thus, given an ordered tuple of bids* $\langle B^{j_1}, \ldots, B^{j_k} \rangle$ *such that* $and(\langle B^{j_1} \ldots B^{j_k} \rangle) = 1$ *then the buyer can only select a bid* $B^{j_i}, 1 < i \geq k$*, whenever* $B^{j_1}, \ldots, B^{j_{i-1}}$ *are also selected. Furthermore, all bids in an AND bid belong to the very same seller. Put formally,* $and(\mathcal{B}) = 1 \Rightarrow \exists!\ \pi \in \Pi$ *such that* $\rho(\pi, B^i) = \rho(\pi, B^j) = 1 \ \forall B^i, B^j \in \mathcal{B}, B^i \neq B^j$.

AND bids are intended to provide the means for the buyer to express volume-based discounts. However, they should be regarded as a generalisation of bidding via price-quantity graphs.

Based on the definitions above we can formally introduce the decision problem to be solved to provide support to the buyer (auctioneer):

Definition 7 (Multi-attribute, multi-unit combinatorial reverse auction). *The multi-attribute, multi-unit combinatorial reverse auction winner determination problem (MMCRAWDP) accounts to the maximisation of the following expression:*

$$\sum_{j=1}^{n} y_j \cdot \sum_{i=1}^{m} w_i \cdot V_i(q_i^j, p_i^j, \Delta_i^j)$$

subject to the following constraints:

1. $q_i^j \in 0 \cup [m_i^j, \bar{m}_i^j]$
2. $q_i^j \ mod \ d_i^j = 0$
3. $\sum_{j=1}^{n} q_i^j = u_i$[3]
4. $\forall \pi_k \in \Pi \quad q_i^j \cdot \rho(\pi_k, B^j) \in \{0\} \cup [c_i^k, \bar{c}_i^k]$
5. $\forall \pi_k \in \Pi \quad q_i^j \cdot \rho(\pi_k, B^j) \in \{0\} \cup [m_i \cdot u_i, \bar{m}_i \cdot u_i]$
6. $\forall \lambda_{j_t}, t \in h(B^j) \quad q_i^j = q_t^j$
7. $\forall \lambda_i \in \Lambda \quad \sum_{k=1}^{r} x_i^k \in [s_i, \bar{s}_i]$
8. $and(\langle B^{j_1}, \ldots, B^{j_k} \rangle) = 1 \Rightarrow y^{j_1} \geq \ldots \geq y^{j_k}$
9. $\forall B' \subseteq B$ *such that* $xor(B') = 1 \quad \sum_{B^j \in B'} y^j \leq 1$

[3] We assume here that there is no free disposal, i.e. sellers are not willing to keep any units of their winning bids, and the buyer is not willing to take any extra units.

10. $a \cdot v_{i,l} + b \geq \delta^j_{i,l} \geq a' \cdot v_{i,l} + b'$ *where* $a, b, a', b' \in \mathbb{R}$
11. $c \cdot v_{i,l} + d \geq v_{j,k} \geq c' \cdot v_{i,l} + d'$ *where* $c, d, c', d' \in \mathbb{R}$

where

- $y_j \in \{0,1\}, 1 \leq j \leq n$, *are decision variables for the bids in* B*;*
- $x^k_i \in \{0,1\}, 1 \leq i \leq m, 1 \leq k \leq r$, *are decision variables to decide whether seller* π_k *is selected for item* λ_i*;*
- $q^j_i \in \mathbb{N} \cup \{0\}, 1 \leq j \leq n, 1 \leq i \leq m$, *are decision variables on the number of units to select from* B^j *for item* λ_i*;*
- $V_i : \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^{i_k} \rightarrow \mathbb{R}, 1 \leq i \leq m$, *are the bid valuation functions for each item; and*
- $v_{i,l}$ *stands for a decision variable for the value of attribute* a_l *of item* λ_i*.*
- $a, b, a', b', c, c', d, d'$ *are buyer-defined constant values.*

Next, we detail the semantics of the restrictions above:

1. This constraint forces that when bid B^j is selected as a winning bid, the allocated number of units of each item q^j_i has to fit between the minimum and maximum number of units offered by the seller.
2. The number of allocated units q^j_i to a bid B^j for item λ_i must be a multiple of the batch d^j_i specified by the bid.
3. The total number of units allocated for each item must equal the number of units requested by the buyer.
4. For each item, the number of units allocated to a seller cannot exceed his capacities.
5. The total number of units allocated per seller cannot exceed or be below the maximum and minimum percentages that can be allocated per seller specified by the buyer.
6. For homogeneous bids, the number of units allocated to the items declared homogeneous must be the same.
7. The number of sellers to be awarded each item cannot exceed or be below the maximum and minimum number of total sellers specified by the buyer.
8. Bids being part of an AND bid can only be selected if the bids preceding them in the AND bid are selected too.
9. XOR bids cannot be jointly selected.
10. Intra-item constraints are modelled through this expression. It indicates that only those bids whose value for the attribute item related to the decision variable that satisfy the expression can be selected.
11. Inter-item constraints are modelled through this expression. It puts into relation decision variables of attributes belonging to different items.

There are several aspects that make our model differ from related work. Firstly, traditionally all combinatorial auction models assume that the buyer (auctioneer) equally prefers all items. Such constraint is not considered in our model, allowing the buyer to express his preferences over items. Secondly, multi-attribute auctions and combinatorial auctions with side constraints have been separately dealt with. On the one hand, Bichler [2] extensively deals with multi-attribute auctions, including a rich bidding language. On the other hand, Sandholm et al. [13] focus on multi-item, multi-unit combinatorial auctions with side constraints where items are not multi-attribute. We have attempted at formulating a model which unites both. Lastly, to the best of our knowledge neither inter-item nor intra-item constraints have been dealt with in the literature at the attribute level [7] though they help us better cope with multiple sourcing.

4 Implementation

The problem of choosing the optimal set of offers sent over by providing agents taking into account the features of the negotiation scenario described in section 2 is essentially an extension of the combinatorial auction (CA) problem in the sense that it implements a larger number of constraints and supports richer bidding models. The CA problem is known to be NP-complete, and consequently solving methods are of crucial importance. In general, we identify three main approaches that have been followed in the literture to fight the complexity of this problem:

- As reported in [8], attempts to make the combinatorial auction design problem tractable through specific restrictions on the bidding mechanism have taken the approach of considering specialised structures that are amenable to analysis. But such restrictions violate the principle of allowing arbitrary bidding, and thus may lead to reductions in the economic outcome.
- A second approach sacrifices optimality by employing approximate algorithms [6]. However, and due of the intended actual-world usage of our service, it is difficult to accept the notion of sub-optimality.
- A third approach consists in employing an exact or complete algorithm that guarantees the global optimal solution if this exists. Although theoretically impractical, the fact is that effective complete algorithms for the CA problem have been developed .

Many of the works reviewed in the literature adopt global optimal algorithms as a solution to the CA because of the drawbacks pointed out for incomplete methods. Basically two approaches have been followed: traditional Operations Research (OR) algorithms and new problem specific algorithms[4]. It is always an interesting exercise to study the nature of the problem in order to develop problem specific algorithms that exploit problem features to achieve effective search reduction. However, the fact is that the CA problem is an instance of the multi-dimensional knapsack problem MDKP (as indicated in [5]), a mixed integer program well studied by the operation research literature. It is not surprising, as reported in [1], that many of the main features of these problem specific new algorithms are rediscoveries of traditional methods in the operations research community. In fact, our formulation of the problem can be regarded as similar to the binary multi-unit combinatorial reverse auction winner determination problem in [13] with side constraints[12]. Besides, expressing the problem as a mixed integer programming problem with side constraints enables its resolution by standard algorithms and commercially available, thoroughly debugged and optimised software which have shown to perform satisfactorily for large instances of the CA problem.

With these considerations in mind, the core of our service has been modelled and implemented as a mixed integer programming problem. We have implemented two versions: a version using ILOG CPLEX 7.1 in combination with SOLVER 5.2; and another version using using iSOCO's Java MIP modeller that integrates the GLPK library [9]. In both cases it takes the shape of a software component. Hereafter we shall refer to this component as the *iBundler* solver.

5 Validation and performance

After the above-described implementation two major issues arose. On the one hand, unitary tests were needed in order to guarantee that *iBundler*'s behaviour is sound or, in other words, that *iBundler* produces the optimal solution taking into account the variety of constraints and bidding expressivenes described in section 2. On the other hand, since combinatorial auction solvers are computationally intensive, a major issue is whether our service was to behave satisfactorily in highly-demanding trading scenarios.

At this aim, we devised a customisable generator of data sets targeted at serving for the two purposes above. Our generator artificially created negotiation problems for *iBundler* by wrapping an optimal solution with noisy bids. Thus *iBundler* is fed by the generator with an RFQ, plus a buyer's constraints, plus a set of bids of varying features (single, combinatorial, AND, and XOR). The generator constructs the artificial negotiation problem based on several parameters, namely: number of providers, number of items, number of bids per provider (mean and variance), number of items per bid (mean and variance), offer price per item (mean and variance). In this way, not only were we able to measure the performance of *iBundler* but also to automatically verify its sound behaviour.

Figure 1 shows how *iBundler* behaves when solving negotiation problems as the number of bids, the number of items, and the items per bid increase. The results show that *iBundler* can be employed to conduct real-time decision-making in actual-world negotiation scenarios. It is hard to imagine a negotiation event in which several hundreds of bids are concurrently submitted by a given set of providers to compete for RFQs containing dozens of items. Nonetheless *iBundler* performs well even in such extreme scenarios. Therefore, its scalability is also ensured.

References

1. A. Andersson, M. Tenhunen, and F. Ygge. Integer programming for combinatorial auction winner determination. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS)*, pages 39–46, Boston, MA, 2000.

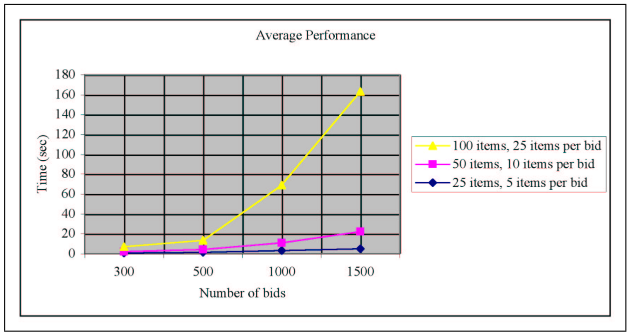

Fig. 1. Average performance of iBundler

2. Martin Bichler and Jayant Kalagnanam. Bidding languages and winner determination in multi-attribute auctions. *European Journal of Operation Research*. (to appear).
3. J. Collins and M. Gini. An integer programming formulation of the bid evaluation problem for coordinated tasks. In R. V. Dietrich, B.and Vohra, editor, *Mathematics of the Internet: E-Auction and Markets*, volume 127 of *IMA Volumes in Mathematics and its Applications*, pages 59–74. Springer-Verlag, 2001.
4. Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceeding of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 548–553, August 1999.
5. R. C. Holte. Combinatorial auctions, knapsack problems, and hill-climbing search. In *Lecture Notes in Computer Science*, volume 2056. Springer-Verlag, Heidelberg, 2001.
6. H. H. Hoos and C. Boutilier. Solving combinatorial auctions using stochastic local search. In *Proceedings of AAAI-2000*, 2000.
7. Jayant Kalagnanam and David C. Parkes. *Supply Chain Analysis in the eBusiness Era*, chapter Auctions, Bidding and Exchange Design. 2003. forthcoming.
8. Frank Kelly and Richard Steinberg. A combinatorial auction with multiple winners for universal service. *Management Science*, 46:586–596, 2000.
9. A. Makhorin. Glpk – gnu linear programming toolkit. http://www.gnu.org/ directory/GNU/glpk.html, 2001.
10. Juan A. Rodríguez-Aguilar, Andrea Giovanucci, Antonio Reyes-Moro, Francesc X. Noria, and Jesús Cerquides. Agent-based decision support for actual-world procurement scenarios. In *2003 IEEE/WIC International Conference on Intelligent Agent Technology*, Halifax, Canada, October 2003.
11. M. H. Rothkopt, A. Pekec, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 8(44):1131–11147, 1995.
12. T. Sandholm and S. Suri. Side constraints and non-price attributes in markets. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, WA, 2001. Workshop on Distributed Constraint Reasoning.
13. T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Winner determination in combinatorial auction generalizations. In *First Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, pages 69–76, Bologna, Italy, July 2002.