# Winner Determination Algorithms for Electronic Auctions: A Framework Design

Martin Bichler, Jayant Kalagnanam, Ho Soo Lee, Juhnyoung Lee

IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA
{bichler, jyl, leehs, jayant}@us.ibm.com
http://www.research.ibm.com

**Abstract.** During the past few years, auctions have become popular in conducting trade negotiations on the Internet. The design of new auctions and other negotiation protocols has become an important topic for both, industry and academia. Traditional auction mechanisms allow price-only negotiations for which the winner determination is a computationally simple task. However, the need for new auction mechanisms that allow complex bids such as bundle bids and multi-attribute bids has been raised in many situations. The winner determination in these auctions is a computationally hard problem. The computational complexity has been a significant hurdle for the widespread use of these advanced auction models. In this paper, we will outline the auction design space and classify resource allocation algorithms along multiple dimensions. Then, we will explain the design of an object framework providing an API to different types of winner determination algorithms. This framework enables application programmers to specify buyer preferences, allocation rules and supplier offerings in a declarative manner, and solve the allocation problems without having to re-implement the computationally complex algorithms.

## 1 Introduction

Auctions have become a fairly successful way of supporting or even automating negotiations on trading markets on the Internet. Examples of traditional auction mechanisms include the English, Dutch, First-Price Sealed-Bid, or Vickrey auction[1]. The competitive process of auctions serves to aggregate the scattered information about bidder's valuation and to dynamically set the prices and conditions of a deal. The typical auction process implemented by most auction platforms consists of the steps of bid submission, bid evaluation (in literature, winner determination and resource allocation are often used for bid evaluation; the three terms will be used interchangeably in this paper), and the calculation of settlement prices, followed by some feedback to the bidders in an iterative, or open-cry auction (see Figure 1). Auctions close either at a fixed point in time or after a certain closing criterion is met (e.g., a certain time elapsed).

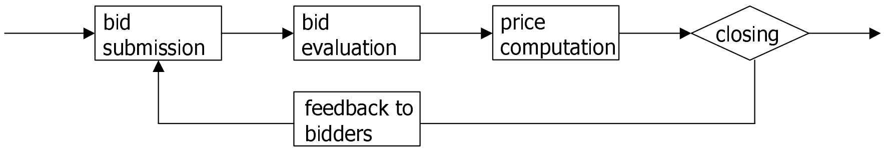


**Fig. 1.** Simplified auction process

A primary emphasis of auction research has been the various rules governing the auction process. The traditional literature mainly focuses on situations where participants negotiate only on price, with the quantity and quality of goods or services being fixed a priori (by the auctioneer). In these cases, the bid evaluation, i.e., the actual task of allocating (demand-side) bids to (sell-side) asks, is computationally simple and consists mainly of sorting bids by price.

During the past few years, however, a number of new auction formats that allow matching buyer's preferences with seller's offerings in a more general sense have been developed. These auctions allow complex bids such as bundle bids and multi-attribute bids and, therefore, allow the negotiation not only on price but also on quantity and qualitative attributes. A popular example of such advanced auction types is the combinatorial auction. It provides a useful negotiation mechanism when there are complementarities or substitutability among several products. In a procurement situation, for example, suppliers can often provide better overall prices if they are allowed to deliver not just one product type for the buyer (e.g., an office owner) but a bundle of product types that complement each other (e.g., computers, monitors and printers). Explicit consideration of these complementarities in combinatorial reverse auctions can lead to substantial cost saving for buying organizations. However, a major hurdle for the use of combinatorial auctions in practice has been that their winner determination is computationally hard. Many of the new auction mechanisms (e.g., multi-unit auctions, volume-discount auctions, and multi-attribute reverse auctions) require solving NP-hard optimization problems (see section 2.3).

The design and implementation of the appropriate decision analysis and optimization models requires skills in operations research as well as software engineering. In this paper we introduce an object framework with a generic API (Application Programming Interface) to winner determination algorithms. The software is intended as a standard solver component for electronic marketplaces and is currently in use with a large-scale procurement platform for the retail industry. This framework enables application programmers to specify buyer preferences, allocation rules and supplier offerings in a declarative manner, and solve the allocation problems without having to re-implement the computationally complex algorithms.

The design of new market mechanisms is a new and emerging field. Many new allocation algorithms have been developed during the past few years, and many new approaches can be expected in the near future. It is important to understand the design space for allocation algorithms, in order to design an easily extensible object framework. We introduce a multidimensional perspective of the auction design space that helps classify advanced resource allocation problems and improve the understanding of their requirements. The framework design can be used as a reference model for similar solver components of this type.

The rest of this paper is structured as follows: Section 2 explains the dimensions of the auction design space and classifies advanced auction formats by three dimensions. Also, typical allocation problems for the auction types are explained. Section 3 describes a number of auction design considerations including the representation of offers. In Section 4, the multidimensional winner determination package with its design objectives, object model, and implementation is described. Finally, in Section 5, the work is summarized and conclusions are drawn.

## 2 Allocation Algorithms for Multidimensional Auctions

In this section, we introduce and classify a number of advanced auction mechanisms and associated allocation algorithms for bid evaluation that have been developed over the past few years. We begin by outlining the overall design space for these allocation algorithms that later will be used to lay the foundation for the design of an object framework.

### 2.1 Design Space of Resource Allocation Algorithms

The problem of bid evaluation arises in many different applications, because it occurs in a variety of negotiation forms including bilateral bargaining, auctions and RFQ (Request For Quotes). Furthermore, auction protocols can be sealed-bid or open-cry, and the auctioneer can reveal various degrees of information about the current status. Also, the protocols can involve rules about who is allowed to bid, how the auction is closed, and what the minimum bid increment is. In addition, an auction can be chained with other negotiation protocols or conducted in multiple rounds, each involving a separate phase of bid evaluation and allocation. Problems similar to bid evaluation arise also in other applications. For example, in a catalog aggregation application where product and pricing data is automatically aggregated from multiple suppliers, buyers need to select one or more among the offerings. There is no negotiation involved in this application, but automated evaluation of various offerings is still a key functionality.

The design space for resource allocation algorithms is huge. The primary criteria for characterizing the allocation problems on electronic markets are:

- the number of participants, and
- the types of traded goods.

The *number of participants* heavily influences the design of an allocation algorithm. Micro-economists distinguish different market structures to categorize different negotiation situations (e.g., monopoly). Different types of negotiation protocols are analyzed in different settings ranging from bilateral bargaining to single-sided auctions and to double auctions. Likewise, the allocation problems change in different settings in terms of the number of participants:

- bilateral allocation problems,
- n-bilateral allocation problems (e.g., single-sided and double auctions), and
- multilateral allocation problems.

Bilateral allocation problems involve bargaining situations and bid evaluation with only two parties. N-bilateral allocation problems typically involve auction settings where there are two types of participants, i.e., buyers and sellers, but there can be more than one of them on each side of the bargaining table. In a reverse auction case, there is one buyer and multiple suppliers, whereas in a double auction, offers of multiple buyers and sellers have to be matched. Finally, multilateral negotiation situations involve more than two parties (e.g., negotiations for mergers and acquisitions including a selling organization, a buying organization and a bank).

An equally important criterion is the *type of traded goods* or services, and more specifically the way, how offers are described. Traditional micro-economic theory distinguishes between homogeneous and heterogeneous goods. For the actual design of a bid evaluation algorithm, one needs to consider more than homogenous goods. Earlier, we introduced combinatorial auctions and briefly mentioned other advanced auction mechanisms such as volume-discount auctions and multi-attribute reverse auctions. These advanced auctions formats can be categorized by several different dimensions of the goods in negotiation (as illustrated in Figure 2):

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

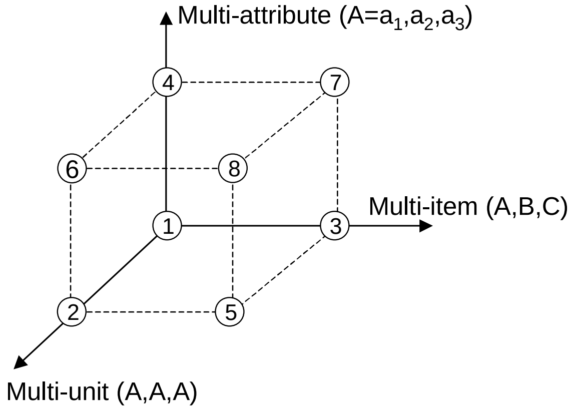


**Fig. 2.** Dimensions of a trade negotiation [2]

### 2.2 Multidimensional Auctions

These three dimensions of goods or services in a trade negotiation provide a way to characterize the relevant aspects of offers (i.e., bids and/or asks). Note that the offer type impacts the bid evaluation algorithm used. Figure 2 shows eight distinct types of offers:

- Offer type 1 describes single-item, price-only, single-unit offers. These offers are used in traditional auctions where all qualitative attributes and quantity are fixed *a priori*, and there is only one particular item in negotiation, e.g., the price for a rare piece of art. The economic behavior of these types of auctions has been extensively studied in game theory and experimental economics (see [1] for an overview).

- Offer type 2 describes single-item, price-only, multi-unit offers. Only one type of item is negotiated, and the quality is predefined. Examples include price-quantity offers in *multi-unit auctions* or offers describing supply-curves in a *volume-discount auction* (see next section).
- Offer type 3 is multi-item, price-only, fixed-quantity offers. An example is a bundle offer that comprises a number of items (a single unit for each) with a single price. Bundle offers are allocated by *combinatorial auctions*.
- Offer type 4 describes pure multi-attribute offers having a single item. These offers have only one type of one item up for negotiation and the quantity is fixed, i.e., suppliers can only provide the entire demanded quantity. These offers are often used in *multi-attribute reverse auctions* and in RFQs.
- Offer types 5 to 8 describe combinations of the dimensions. Offer type 5 contains multiple items where quantity is also negotiable. Offer type 6 describes fixed-quantity offers where qualitative attributes and quantity of goods are negotiable. Offer type 7 is multi-attribute, multi-item offers where the quantity is fixed. Finally, Offer type 8 describes pure multi-dimensional offers where all dimensions are negotiable.

In the next section, we will explain how these multidimensional offers can be evaluated in advanced auction mechanisms that allow complex offer types, such as multi-unit auctions, volume-discount auctions, combinatorial auctions or multi-attribute reverse auctions.

### 2.3 Allocation Problems in Multidimensional Auctions

Recently, there has been active research on *multi-unit auctions* in which bidders submit both the number of units of an item they wish to buy and how much they are willing to bid per unit. The first-price auction has two multi-unit generalizations which are used in practice – *discriminatory auctions* in which the bidders who bid the highest prices pay their bid, and *uniform-price auctions* in which all $k$ successful bidders pay the ($k$+1)st bid price. The uniform price is set by the bid on the ($k$+1)st unit which is not a winning bid, because there are only $k$ units on sale. Most on-line auctions are discriminatory auctions, where each bidder pays the bid price. If offers are divisible, i.e., the bidders are willing to accept a partial quantity for the same price, the winner determination becomes trivial and can be solved by sorting the offers by unit price. If, however, offers are indivisible, which is the usual case in multi-unit auctions, the auctioneer needs to solve a 0/1-Knapsack problem known to be NP-complete [3]:

$$\max \sum_i (n_i p_i) x_i \quad (1)$$

$$s.t. \sum_i n_i x_i \leq K, \quad x_i \in \{0,1\} \quad (2)$$

In the above formulation a binary variable $x_i$ is assigned to each offer, which indicates acceptance or rejection. $p_i$ defines price and $n_i$ the number of units in each bid. In practice, these problems are often solved using dynamic programming.

*Volume-discount auctions* are an interesting extension of multi-unit auctions. These auctions can be used in a procurement context where bidders are allowed to specify the price they charge for an item as a function of order quantity. Bids take the form of supply curves that specify the price to be charged per unit of item when the quantity of items being purchased lies within a particular quantity interval. The allocation algorithm described in [4] uses a mixed integer programming formulation to select the cost-minimizing set of bids.

*Combinatorial auctions* (also referred to as multi-item or bundle auctions) are an approach to achieving efficient allocations in situations where bidders are allowed to place bids on combinations of possibly heterogeneous goods or services [5, 6]. An example is a bid on a group of adjacent real estate properties. A combinatorial bid can win only if it is greater than the sum of the high bids in the auctions for the individual licenses. A winning bid on a shipping lane between two production plants is valuable to a transporter only if it also wins the lane in the opposite direction. The transporter must bid higher combinatorial for both than the sum of the high bids for the individual lanes.

Combinatorial auctions are also used in a procurement context where several goods or services that have complementarities among them are purchased together. In these situations, the auctioneer tries to find the combination of bids with the lowest cost. In other words, the winner determination problem for a combinatorial reverse auction is to select a winning set of bids such that the demand for each item is satisfied and, at the same time, the total cost of procurement is minimized. This problem can be formulated as a weighted set covering problem which is also known to be NP-hard. Set covering can also be solved by using integer programming techniques. Combinatorial forward auctions can be treated in a similar manner as a set packing problem [7].

Another interesting type of multidimensional auction is the *multi-attribute auction* [8] where bidders are allowed to submit multi-attribute bids and therefore negotiate not just on price but quantity and qualitative attributes. Typical multi-attribute auctions describe bids as a set of attribute-value pairs. An approach to handling multiple attributes is to convert qualitative attributes into price-equivalents by using a certain set of rules. Another approach is to define scoring functions for individual attributes and calculate the bid scores by aggregating attribute scores using a linear or non-linear utility function. These techniques have their origin in multi-attribute decision analysis [9]. More advanced settings considering multiple sourcing and indivisible bids also need to solve optimization problems.

All these advanced auction mechanisms provide bidders with increased flexibility and expressiveness in describing their offers. They promise higher market efficiency, because they allow for a more effective exchange of relevant market information. Most of these auctions already found successful applications in a variety of industries, and one can expect more developments in this field.

## 3 Design Considerations

A number of evaluation criteria are used to analyze auctions and other economic mechanisms [8, p. 73]. The auction process should converge to equilibrium, i.e., a solution, in which no agent wishes to change its bid. Of course, the speed of this convergence is important, too. A solution of an auction should be stable, so that no subset of agents could have done better by coming to an agreement outside the auction. Mechanism design theory also suggests that optimal auctions should be incentive-compatible; i.e., honest reporting of valuations is a Nash-equilibrium. These criteria are important for the design of the auction process. The key criterion for the winner determination is, however, whether the solution to an allocation is shows *allocative efficiency*. In other words, no rearrangement of resources could make someone better off without making someone else worse off.

In section 2.3 we have illustrated the complexities of determining efficient resource allocations. The major factor influencing this complexity is the bid representation. Bid representation has not been an issue in auction research until now. For example, offers in multi-attribute auctions are presented as sets of attribute-value pairs. A recent paper by Bichler, Kalagnanam and Lee [10] extends the notion of multi-attribute offers to *configurable offers*. The ECCO system presented in this paper allows the specification of multiple options for each attribute in an offer. The system defines for each attribute a set of possible values and their associated markup prices. Also, it allows the offers to include a set of logical rules on how to combine various attribute values (so called *configuration rules*), and how certain configurations impact the price (i.e. *discount rules*). The ECCO winner determination algorithm takes into account these rules along with buyer preferences and selects the best out of all possible configurations. In a similar way, logical bid languages have been defined for bundle offers, which allow for a *compact description* of a large number of explicit bundle bids [11]. Reeves et al. [12] use even the expressiveness of first-oder logic in representing complex offers. They presented an approach where offers in a negotiation are expressed as logic programs. More elaborate bid representation schemes enable higher expressiveness and flexibility in a negotiation at the expense of increased complexity of bid evaluation.

## 4 A Framework for Multidimensional Auction Markets

The Multidimensional Auction Platform (MAP) is a set of software modules for building multidimensional auction markets which has been developed at IBM T. J. Watson Research Center. MAP features several Java packages:

- Package *com.ibm.allocation* provides an object framework for resource allocation algorithms in electronic markets.
- Package *com.ibm.allocation.solvers* provides façade classes to various optimization packages such as IBM's OSL, or Ilog's CPLEX. These façade classes make it possible to seamlessly switch from one optimization package to another.

- Package *com.ibm.allocation.datasource* provides façade classes to different kinds of data sources. These façade classes make it possible to use algorithms with different data sources, e.g., databases or XML files, and different data schemas.
- Package *com.ibm.preference* provides Java classes for various preference elicitation methods useful in a multi-attribute negotiation [13, 14].

### 4.1 Design Objectives

In the following discussion, we will concentrate on the *allocation package* (com.ibm.allocation), which is the centrepiece of MAP. The allocation package is implemented as an object framework in the sense that it is a collection of cooperating objects that provide an integrated solution customizable by the developer. It is important to note that unlike a class library approach, the framework approach provides a bi-directional flow of control between an application and the framework [15]. This feature enables easy extension with customer specific code.

The primary design objective for the allocation framework relates to its reuse of various classes in solving different instances of allocation problems. The fundamental strategy used in designing the object framework involves identifying high-level abstraction of common problem classes. The main objectives in designing the framework are summarized as follows:

- The framework should provide an easy-to-use and declarative interface which hides the complexity of the underlying allocation algorithms.
- It should allow for rapid prototyping with maximal reuse of available problem representation and algorithms.
- It should be possible to switch between existing types of allocation algorithms with minimal effort, and adopt new allocation algorithms into the framework with little customisation.
- The basic data structures and algorithms should be reusable in different environments and applications, e.g., in electronic auction implementation, in stand-alone decision support tools for procurement, or in agent-based computer simulations.

### 4.2 Object Model

The object model of the allocation framework is illustrated in Figure 3. There are three core interfaces in the framework, *Offer*, *AllocationAlgorithm*, and *Solution*. These interfaces provide common denominators for various types of allocation algorithms possible. The *Offer* interface is implemented by many classes including *PriceOnlyOffer*, *PriceQuantityOffer*, *VolumeDiscountOffer*, *BundleOffer*, *MultiAttributeRequest*, and *MultiAttributeOffer*. Bundle offers assign a single price to a bundle of heterogeneous items. Volume-discount offers are used in volume-discount auctions as described in Section 2. Each *Offer* is submitted by an *Agent* and assigned to an *AllocationAlgorithm*. For each *Agent* or *AgentGroup*, an application programmer can define *AgentConstraints*, which defines minmum and maximum allocations per agent, or per agent and item. An implementation of *AllocationAlgorithm* can then allocate certain implementations of *Offer*, and return the result of this allocation in a *Solution*. For example, a *VolumeDiscountAllocation* assigns one or more responses of class *VolumeDiscoutOffer* to a *BundleOffer* which specifies the request.

Based on the number of participants, we distinguish three broad types of allocation algorithms, namely, *TwoPartyAllocation*, *SingleSidedAllocation* and *ClearingHouseAllocation*. Examples of two-party situations are bilateral negotiations on price, or on multiple attributes, where only one buy-side offer and one sell-side offer are matched at a point in time. In addition, a set of classes implements the interface *ClearingHouseAllocation*. These classes allocate multiple buy-side and sell-side bids at a point in time. A typical application of such an algorithm can be found in financial call markets where bids and asks on price and quantity are collected over a specified interval of time, and then the market clears, i.e., the offers are allocated at a price that maximizes the turnover. Finally, an implementation of *SingleSidedAllocation* can be used when one or more responses need to be selected to satisfy a single request. Typical applications of these algorithms are single-sided forward or reverse (i.e., procurement) auctions. In Section 2, we have discussed a number of such allocation problems.

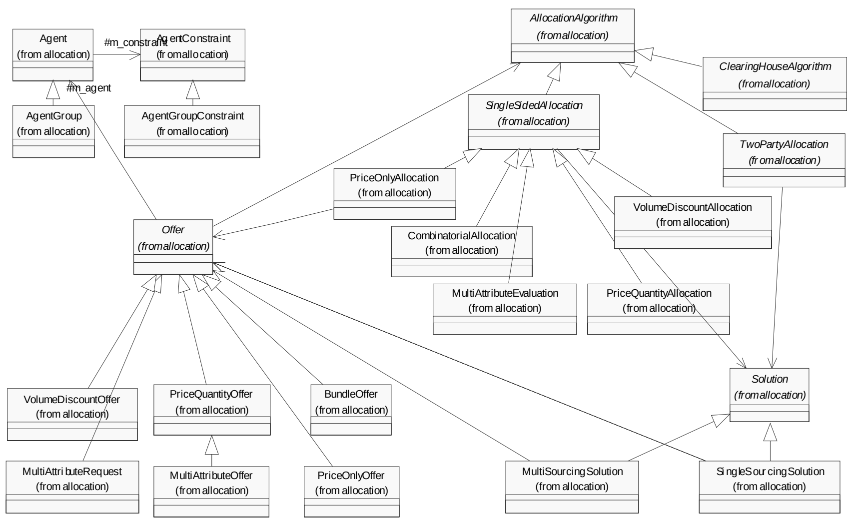


**Fig. 3.** Selected Class Hierarchies of the Allocation Package

## 5 Conclusions

Recently, there has been a significant shift in e-commerce from static forms of pricing to dynamic pricing [2]. Many new and versatile auction formats have been developed, which allow more flexibility in specifying demand and supply, and ultimately lead to more efficient outcomes. This trend has led to an increasing need for fast and efficient winner determination algorithms. Reusable implementations of these algorithms enable businesses to easily deploy new negotiation protocols in their procurement processes. In this paper, we have discussed the design space and the design objectives for such resource allocation algorithms. We have also described an extensible object framework which enables the reuse of the advanced allocation algorithms as a standard solver component in electronic markets. It provides a declarative interface and sheds developers from the complexities of a particular allocation algorithm. The framework has been developed at IBM T. J. Watson Research Center and is currently being used in a number of customer engagements.

## References


1. E. Wolfstetter, "Auctions: An Introduction," *Journal of Economic Surveys*, vol. 10, pp. 367-420, 1996.
2. M. Bichler, J. Kalagnanam, K. K. Katircioglu, A. J. King, R. D. Lawrence, H. S. Lee, G. Y. Lin, and Y. Lu, "Applications of Flexible Pricing in Business-to-Business Electronic Commerce," *IBM Systems Journal*, 2002.
3. M. Eso, "Evaluation algorithms in auctions," IBM T.J. Watson Research Center, Yorktown Heights, Working Paper 2001.
4. A. Davenport and J. Kalagnanam, "Price Negotiations for Procurement of Direct Inputs," IMA "Hot Topics" Workshop: Mathematics of the Internet: E-Auction and Markets, Minneapolis, USA, Workshop Report December 3-5, 2000 2000.
5. M. H. Rothkopf and A. Pekec, "Computationally Manageable Combinatorial Auctions," presented at Maryland Auction Conference, Maryland, USA, 1998.
6. T. Sandholm, "Approaches to winner determination in combinatorial auctions," *Decision Support Systems*, vol. 28, pp. 165-176, 1999.
7. G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, *Optimization*. Amsterdam: Elsevier, 1989.
8. M. Bichler, *The Future of eMarkets: Multi-Dimensional Market Mechanisms*. Cambridge, UK: Cambridge University Press, 2001.
9. D. L. Olson, *Decision Aids for Selection Problems*. New York, et al.: Springer, 1995.
10. M. Bichler, J. Kalagnanam, and H. S. Lee, "ECCO - Automated Evaluation of Configurable Offerings," IBM Research, Yorktown Heights, Research Report 2002.
11. C. Boutilier and H. H. Hoos, "Bidding Languages for Combinatorial Auctions," presented at 17th International Joint Conference on Artificial Intelligence (IJCAI) 2001, Washington, USA, 2001.
12. D. Reeves, B. Grosof, and M. Wellman, "Automated Negotiations from Declarative Contract Descriptions," presented at AAAI-2000 Workshop on Knowledge-Based Electronic Markets, Austin, TX, 2000.
13. M. Bichler, J. Lee, C. H. Kim, and H. S. Lee, "Design and Implementation of an Intelligent Decision Analysis System for E-Sourcing," presented at International Conference on Artificial Intelligence 2001, Las Vegas, NV, 2001.
14. V. S. Iyengar, J. Lee, and M. Campbell, "Q-Eval: Evaluating Multiple Attribute Items Using Queries," IBM Research, New York, Research Report 2001.
15. J. Sametinger, *Software Engineering with Reusable Components*. New York: Springer, 1997.