# Optimal clearing algorithms for multi-unit single-item and multi-unit combinatorial auctions with demand/supply function bidding

[Relevant track: eCommerce Technologies]

Viet Dung Dang* and Nicholas R. Jennings
Department of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ, UK.
{vdd00r,nrj}@ecs.soton.ac.uk

## ABSTRACT

This paper presents new clearing algorithms for multi-unit single-item and multi-unit combinatorial auctions with piecewise linear demand/supply functions. We analyse the complexity of our algorithms and prove that they are guaranteed to find the optimal allocation.

## Keywords

Auctions, Winner Determination, Intelligent Agents

## 1. INTRODUCTION

Traditionally, the most common forms of online auction are the simple, single-sided auctions in which a single item is traded (e.g. English, Dutch, first price sealed-bid and Vickrey). However, such auctions are inefficient when there is a *correlation* between different items that bidders want to purchase. When there are such synergies (i.e. benefits from combining complementary items), the inability to bid on groups of items means the bidder faces the risk of winning only a part of the desired set. To overcome the problems associated with correlated items, more sophisticated marketplaces are needed in which multiple units of multiple (potentially inter-related) items can be traded simultaneously. Such auctions are called *combinatorial auctions*.[1] In this type of auction, bidders may bid for arbitrary combinations of items. For example, a single bid may be for $q$ units of item 1 and $2 * q$ units of item 2 at price $40 * q$ if $q < 20$, at price $34 * q$ if $20 \leq q < 40$, and at price $30 * q$ if $q \geq 40$.

While combinatorial auctions have many potential benefits from an economic perspective [5], their main disadvantages stem from the lack of efficient *clearing algorithms*[2] for determining the prices, quantities and trading partners as a function of the bids made. To overcome this problem, there has been considerable recent work in this area (see section 4 for more details). However, almost all of this work (e.g. [3], [4], [6], [7], [10]) has considered bids to be *atomic propositions* that are either accepted in their entirety or rejected. This view, while appropriate in some cases, has the disadvantage of limiting the choice, and hence the potential profit, available to the auctioneer. For example, consider the case where there are only two bids for the same good: $x_1$ units at price $p_1$ and $x_2$ units at price $p_2$, and the auctioneer wants to trade fewer than $x_1 + x_2$ units of the good. In this case, the auctioneer has no choice other than selecting one or other of the two bids. This may prevent the auctioneer from maximising its payoff. For example, the auctioneer may find it more beneficial to accept both bids partially; that is, trade $y_1$ ($y_1 < x_1$) units with bidder 1 at price $\frac{y_1}{x_1} \cdot p_1$ and trade $y_2$ ($y_2 < x_2$) units with bidder 2 at price $\frac{y_2}{x_2} \cdot p_2$.

Moreover, if the bids are expressed in terms of the correlation between the quantity of items and the price (rather than the simple linear extrapolation above[3]), there will be even more choice for the auctioneer, and, consequently, even more chance of maximising its payoff. When viewed from the bidder's perspective, the atomic nature of bids and the inability to explicitly relate price and quantity means that opportunities for trade are lost because the auctioneer may not want the entire package being offered, even though elements of it may be acceptable. Although nearly all the aforementioned work permits XOR (exclusive-or) bids[4], and, in theory, the

*The first author is a PhD student and is funded by BTexact Technologies.

[1]If there is only a single unit of each type of item, the auctions are called *single-unit combinatorial auctions*. If there are multiple units of each type of item, the auctions are called *multi-unit combinatorial auctions* [10].

[2]Also called the *winner determination problem* [10] or the *bid evaluation problem* [4].

[3]In many cases, linear extrapolation simply doesn't work because bidders value bundles of items non-linearly. This may occur, for example, because there is a high set-up cost and then producing multiple versions of the same bundle is comparatively cheap.

[4]An XOR bid is one in which a bidder submits an arbitrary number of atomic proposition bids with the condition that

correlation function between the quantity and the price may be expressed using XOR atomic proposition bids to specify points; in practice, it is nearly impossible as the number of points on the graph of the function could be exponential. For example, suppose a bidder wants to trade 1000 units with unit price 10 if the quantity is less than 100, and with unit price 9 if the quantity is in the range between 100 and 1000. With XOR bidding, the bid has to be expressed as XOR of 1000 atomic proposition bids, in which each atomic bid is a pair of quantity and price for every quantity from 1 to 1000. This is clearly inefficient.

To overcome the aforementioned shortcomings associated with atomic propositions, Sandholm and Suri consider the case in which agents can submit bids that correspond to a demand or supply curve depending on whether it is an auction or a reverse auction respectively [9]. Thus, bids are expressed in terms of a curve that correlates the quantity with the price of an item. For example, an agent may express the bid as $q = 2 * p + 1$, which means that the agent is willing to trade up to $q = 2 * p + 1$ units if the unit price equals $p$.[5] Unfortunately, their work is limited to multi-unit single-item auctions and does not deal with the combinatorial case. Other researchers have further considered multi-unit combinatorial reverse auctions with supply curves [3] [4]. However, in this work, bidders submit separate supply curves for different items, and it is assumed that the price of a package of items is equal to the sum of all the prices of the separate items.[6] This means that these auctions are not truly *combinatorial* in nature as the correlation between items is ignored. In previous work [2], we have developed algorithms for multi-unit combinatorial reverse auctions with supply function bidding, however, these algorithms, while running in polynomial time and producing solutions that are within a finite bound of the optimal, are not guaranteed to find the optimal allocation.

Against this background, we advance the state of the art in this paper by making two important contributions. First, we develop a compact bid representation for multi-unit combinatorial auctions with demand/supply curves that allows bidders to express the correlation between separate items. Secondly, we develop *optimal* clearing algorithms for this class of auctions. Thus, our work removes the shortcomings associated with the atomic proposition and the non-combinatorial nature of the aforementioned work.[7] Specifically, we consider multi-unit single-item and multi-unit combinatorial auctions in which bids contain an agent's demand/supply function. We develop provably optimal clearing algorithms and analyse their computational complexity for the case where the demand/supply curves for each individual commodity are *piece-wise linear functions.* This case is important to consider because such curves are commonly used in industrial trading applications [3] and any general curve can be approximated arbitrarily closely by a family of such functions [9].

The remainder of the paper is organised as follows. Section 2 formalises the problem of auction clearing. Section 3 presents the algorithms for the case of piece-wise linear function bids. Section 4 discusses related work, and section 5 concludes and presents future work.

it is willing to obtain at most one of these bids [8].

[5]Their price function calculates quantity from unit price. However, in our work, the price function will calculate unit price from quantity, because we find the later more natural.

[6]This property is called *additive separability* in [4].

[7]This paper just reports on the reverse case (for reasons of space) although the same algorithms and results can also be applied to the forward case (see [1] for more details).

## 2. AUCTION CLEARING WITH SUPPLY FUNCTION BIDS

This section formalises the problem of clearing in multi-unit combinatorial auctions with supply function bids. Assume there are $m$ items (goods or services): $1, 2, ..., m$ and $n$ bidders $a_1, a_2, ..., a_n$. The auctioneer has a demand $(q_1, q_2, ..., q_m)$, in which $q_j$ is the quantity of item $j$ that the auctioneer is willing to buy. Let $u_i^j$ be the maximum quantity of item $j$ that $a_i$ is able or willing to sell (if $a_i$ is not willing to sell an item $j$, then $u_i^j = 0$). Let $\mathbb{N}$ be the set of natural numbers and $\mathbb{R}^*$ be the set of non-negative real numbers.

The *supply function* is the price function of the items that each bidder is willing to sell. The supply function of bidder $i$ is:

$$P_i : \mathbb{N}^m \rightarrow \mathbb{R}^*,$$

where $P_i(r_1, r_2, ..., r_m)$ is the price offered by bidder $i$ for the package of items $(r_1, r_2, ..., r_m)$ and $r_j$ is the quantity of item $j$, $r_j \in \mathbb{N}$, $0 \leq r_j \leq u_i^j$, $\forall\ 1 \leq j \leq m$.

Having determined the demand function, we now consider the *supply allocation* which is the amount that the auctioneer trades with each bidder.

*Definition 1.* A supply allocation is a tuple $\langle r_i^j \rangle, 1 \leq i \leq n, 1 \leq j \leq m$ such that the auctioneer buys $r_i^j$ units of item $j$ from each agent $a_i$.[8]

Given the definitions of the supply function and the supply allocation, the problem of reverse auction clearing is then to find a supply allocation $\langle r_i^j \rangle, 1 \leq i \leq n, 1 \leq j \leq m$ that:

- Satisfies the demand constraint:

$$\sum_{i=1}^{n} r_i^j \geq q_j, \forall 1 \leq j \leq m \tag{1}$$

  That is, the quantity of each item that the auctioneer buys from all bidders is not less than the auctioneer's demand for that item.

- Optimises the auctioneer's total revenue:

$$P(\langle r_i^j \rangle) = \sum_{i=1}^{n} P_i(r_i^1, r_i^2, .., r_i^m) \text{ is minimal.} \tag{2}$$

  That is, the total price of all the units of all the items supplied by the bidders should be as small as possible.

However, the auction clearing problem has been shown to be NP-complete, even for the simplified case of single-items with piecewise linear demand/supply curves [9].[9] Thus, it is impossible to find a polynomial algorithm that is guaranteed

[8]Because the auctioneer buys items at the price the bidders offer, it may well be the case that the auctioneer will buy the same package from different bidders at different prices. Thus, the auctions have *discriminatory pricing* (which is a widely used assumption in the literature).

[9]Although [9] does not explicitly consider our cases, their proof also holds for them.

to find the optimal allocation, unless P = NP. To this end, the next section presents our algorithms for the case where the function curves for each individual commodity are piecewise linear. In this paper, we concentrate on optimality and so, necessarily, our algorithms are not polynomial.

# 3. PIECE-WISE LINEAR SUPPLY CURVE BIDS

In this section, we consider the case where:

$$P_i(r_1, r_2, ..., r_m) = \omega_i(t_1, t_2, ..., t_m) \cdot (\sum_{j=1}^{m} P_i^j(r_j))$$

where $P_i^j$ is the price function of agent $i$ for item $j$, in the form of a piecewise linear curve (i.e. the function's graph is composed of many segments, each of which is linear), $t_j$ is the segment number of $P_i^j$ that $r_j$ belongs to and

$$\omega_i : \{(t_1, t_2, ..., t_m)|t_j \text{ is a segment number of } P_i^j\} \rightarrow \mathbb{R}$$

is the function that expresses correlations between items in the set $S$.

More precisely, each piece-wise linear function $P_i^j$ is composed of $N_i^j$ linear segments, numbered from 1 to $N_i^j$. Each individual segment with segment number $l$, $1 \leq l \leq N_i^j$, is described by a starting quantity $s_{i,l}^j$ and an ending quantity $e_{i,l}^j$, a unit price $\pi_{i,l}^j$ and a fixed price $c_{i,l}^j$, with the meaning that: bidder $i$ wants to trade any $r$ units of item $j$, $s_{i,l}^j \leq r \leq e_{i,l}^j$ with the price:

$$P = \pi_{i,l}^j \cdot r + c_{i,l}^j$$

Note that the function $P_i^j$ is not required to be continuous; that is, $(s_{i,l+1}^j - e_{i,l}^j)$ may not equal 1. Also, for convenience, we call segment number 0 the segment in which the starting quantity, the ending quantity, the unit price and the fixed price are all equal to 0. Thus, the number of segments of $P_i^j$, including this special segment, will equal $N_i^j + 1$.

The correlation function $\omega_i$ has many potential uses in real-life scenarios. For example, suppose bidder $i$, selling 3 items (1, 2 and 3), wants to express things like "I am willing to sell $r_1$ units of item 1 and $r_2$ units of item 2 together with a price $p$, but not separately". Using our correlation function, this can be expressed by adding segments $t_1$ and $t_2$, which contain only $r_1$ and $r_2$, to the functions $P_i^1$ and $P_i^2$, respectively, then giving $\omega_i(t_1, t_2, t_3)$ a very small value, for every $t_3$, and giving $P_i^1(r_1)$ and $P_i^2(r_2)$ very big values. This way, the auctioneer will never choose to buy $r_1$ or $r_2$ separately.

Although this is not our main focus, this means of representing bids is novel and superior to those previously discussed. Compared with [9], [3] and [4], for instance, it is more expressive as it allows bidders to detail the correlation between separate items. Compared to XOR atomic proposition presentations, it is as expressive but much more compact (as per our example in section 1).

For convenience, from this section on, we will use the following terms.

*Definition 2.* A *valid allocation* is a supply allocation that completely satisfies the demand constraint.

*Definition 3.* A supply allocation $\langle \bar{r}_i^j \rangle$ is *not less profitable than* a supply allocation $\langle r_i^j \rangle$ if the former brings the auctioneer an equal or bigger revenue than the latter. That is:

$$P(\langle \bar{r}_i^j \rangle) \leq P(\langle r_i^j \rangle)$$

According to this definition of profitability, the most profitable valid allocation optimises the auctioneer's total revenue. Thus, this is what our algorithms aim to find. We first consider the multi-unit single-item case (section 3.1), before moving onto the combinatorial case (section 3.2).

## 3.1 MULTI-UNIT SINGLE-ITEMS

Using the notation from the previous section, the single-item case can be re-formulated as follows. Let $n$ be the number of bidders. The auctioneer has a demand $q$. Each bidder $i$ submits bids in the form of a piece-wise linear supply curve: $P_i : N \rightarrow R$, which is composed of $N_i$ linear segments. Each segment $l$, $0 \leq l \leq N_i$ is described by a starting quantity $s_{i,l}$ and an ending quantity $e_{i,l}$, a unit price $\pi_{i,l}$ and a fixed price $c_{i,l}$.

*Definition 4.* The dominant set $D$ is the set of all allocations $(r_1, r_2, .., r_n)$ such that there exists a $k$, $1 \leq k \leq n$, such that all $r_{\lambda_1}, ..., r_{\lambda_{k-1}}$ equal the ending quantity of the segments that they belong to, and all $r_{\lambda_{k+1}}, ..., r_{\lambda_n}$ equal the starting quantity of the segments that they belong to:[10]

$$\begin{cases} r_{\lambda_i} &= e_{\lambda_i, t_{\lambda_i}}, \forall 1 \leq i \leq k-1 \\ r_{\lambda_i} &= s_{\lambda_i, t_{\lambda_i}}, \forall k+1 \leq i \leq n \\ r_{\lambda_k} &= q - \sum_{i=1, i \neq k}^{n} r_{\lambda_i} \end{cases}$$

where:

- $t_i$ is the segment on $P_i$ that $r_i$ belongs to. That is, $s_{i,t_i} \leq r_i \leq e_{i,t_i}$.
- $(\lambda_i)_{i=1}^n$ is any permutation of $(1, 2, ..., n)$ such that $\{\pi_{\lambda_1, t_{\lambda_1}}\}_{i=1}^n$ is sorted increasingly:[11]

$$\pi_{\lambda_1, t_{\lambda_1}} \leq \pi_{\lambda_2, t_{\lambda_2}} \leq ... \leq \pi_{\lambda_n, t_{\lambda_n}}$$

From this, a number of lemmas follow:

LEMMA 1. *For every allocation $(r_1, r_2, ..., r_n)$ there exists an allocation in the dominant set $D$ that is not less profitable than it.*

PROOF. Let $(r_1, r_2, ..., r_n)$ be an allocation. Let $t_i$ be the segment that $r_i$ belongs to. Suppose $(\lambda_i)_{i=1}^n$ is a permutation of $(1, 2, ..., n)$ such that:

$$\pi_{\lambda_1, t_{\lambda_1}} \leq \pi_{\lambda_2, t_{\lambda_2}} \leq ... \leq \pi_{\lambda_n, t_{\lambda_n}} \quad (3)$$

**Step 1**: we prove that there exists an allocation $\langle r_i^{(1)} \rangle$, that is not less profitable than $\langle r_i \rangle$, where $r_i^{(1)}$ belongs to segment $t_i$ of $P_i$, $\forall 1 \leq i \leq n$ and, either $r_{\lambda_1}^{(1)} = e_{\lambda_1, t_{\lambda_1}}$ or:

$$\begin{cases} r_{\lambda_i}^{(1)} &= s_{\lambda_i, t_{\lambda_i}}, \forall 2 \leq i \leq n \\ r_{\lambda_1}^{(1)} &= q - \sum_{i=2}^{n} r_{\lambda_i}^{(1)} \end{cases}$$

Let us consider the case where $r_{\lambda_1} < e_{\lambda_1, t_{\lambda_1}}$ and there exists a $k$, $2 \leq k \leq n$, such that $r_{\lambda_k} > s_{\lambda_k, t_{\lambda_k}}$.

[10] There may be many dominant sets $D$, as there may exist many permutations $(\lambda_i)_{i=1}^n$.

[11] There may exist many such permutations $(\lambda_i)_{i=1}^n$, as there may be many ways to sort the set $\{\pi_{i,t_i}\}_{i=1}^n$.

Consider the allocation $(r'_1, r'_2, ..., r'_n)$ where:

$$\begin{cases} r'_{\lambda_1} & = & r_{\lambda_1} + 1 \\ r'_{\lambda_k} & = & r_{\lambda_k} - 1 \\ r'_{\lambda_i} & = & r_{\lambda_i}, \forall 1 \le i \le n, i \neq 1, i \neq k \end{cases}$$

Because $r_{\lambda_1} < e_{\lambda_1, t_{\lambda_1}}$ and $r_{\lambda_k} > s_{\lambda_k, t_{\lambda_k}}$, $r'_i$ belongs to segment $t_i$ of $P_i$, $\forall 1 \le i \le n$.

Now let us compare the revenues of two allocations $\langle r_i \rangle_{i=1}^n$ and $\langle r'_i \rangle_{i=1}^n$. We have:

$$\begin{aligned} & P(\langle r_i \rangle) - P(\langle r'_i \rangle) \\ & = \sum_{i=1}^n (P_{\lambda_i}(r_{\lambda_i})) - \sum_{i=1}^n (P_{\lambda_i}(r'_{\lambda_i})) \\ & = P_{\lambda_1}(r_{\lambda_1}) + P_{\lambda_k}(r_{\lambda_k}) \\ & \quad -(P_{\lambda_1}(r_{\lambda_1}+1) + P_{\lambda_k}(r_{\lambda_k}-1)) \\ & = (\pi_{\lambda_1, t_{\lambda_1}} \cdot r_{\lambda_1} + c_{\lambda_1, t_{\lambda_1}}) + (\pi_{\lambda_k, t_{\lambda_k}} \cdot r_{\lambda_k} + c_{\lambda_k, t_{\lambda_k}}) \\ & \quad -(\pi_{\lambda_1, t_{\lambda_1}} \cdot (r_{\lambda_1}+1) + c_{\lambda_1, t_{\lambda_1}}) \\ & \quad -(\pi_{\lambda_k, t_{\lambda_k}} \cdot (r_{\lambda_k}-1) + c_{\lambda_k, t_{\lambda_k}}) \\ & = \pi_{\lambda_k, t_{\lambda_k}} - \pi_{\lambda_1, t_{\lambda_1}} \end{aligned}$$

But by inequation (3): $\pi_{\lambda_k, t_{\lambda_k}} \ge \pi_{\lambda_1, t_{\lambda_1}}$. Thus:

$$P(\langle r_i \rangle) \ge P(\langle r'_i \rangle)$$

This means by taking 1 more unit from bidder $\lambda_1$ and taking 1 less unit from bidder $\lambda_k$, we will have a new allocation that is not less profitable than the original one.

Repeating the above process, we will always get a new allocation that is not less profitable than the original one. Eventually we get an allocation $\langle r_i^{(1)} \rangle$, that is not less profitable than the original one, where $r_i^{(1)}$ belongs to segment $t_i$ of $P_i$, $\forall 1 \le i \le n$, and either $r_{\lambda_1}^{(1)} = e_{\lambda_1, t_{\lambda_1}}$ or:

$$\begin{cases} r_{\lambda_i}^{(1)} & = & s_{\lambda_i, t_{\lambda_i}}, \forall 2 \le i \le n \\ r_{\lambda_1}^{(1)} & = & q - \sum_{i=2}^n r_{\lambda_i}^{(1)} \end{cases}$$

**Step 2**: In the case if $r_{\lambda_1}^{(1)} = e_{\lambda_1, t_{\lambda_1}}$ and $r_{\lambda_1}^{(1)} < q - \sum_{i=2}^n s_{\lambda_i, t_{\lambda_i}}$, by repeating the above step, there exists an allocation $\langle r_i^{(2)} \rangle$, that is not less profitable than $\langle r_i \rangle$, where:

- $r_i^{(2)}$ belongs to segment $t_i$ of $P_i$, $\forall 1 \le i \le n$.
- $r_{\lambda_1}^{(2)} = r_{\lambda_1}^{(1)} = e_{\lambda_1, t_{\lambda_1}}$
- Either $r_{\lambda_2}^{(2)} = e_{\lambda_2, t_{\lambda_2}}$ or:

$$\begin{cases} r_{\lambda_i}^{(2)} & = & s_{\lambda_i, t_{\lambda_i}}, \forall 3 \le i \le n \\ r_{\lambda_2}^{(2)} & = & q - \sum_{i=1, i \neq 2}^n r_{\lambda_i}^{(2)} \end{cases}$$

By repeating the above steps again and again, we will finally stop at some step $k$, $1 \le k \le n$ and get an allocation $\langle r_i^{(k)} \rangle$, that is not less profitable than $\langle r_i \rangle$, where $r_i^{(k)}$ belongs to segment $t_i$ of $P_i$, $\forall 1 \le i \le n$, and:

$$\begin{cases} r_{\lambda_i}^{(k)} & = & e_{\lambda_i, t_{\lambda_i}}, \forall 1 \le i \le k-1 \\ r_{\lambda_i}^{(k)} & = & s_{\lambda_i, t_{\lambda_i}}, \forall k+1 \le i \le n \\ r_{\lambda_k}^{(k)} & = & q - \sum_{i=1, i \neq k}^n r_{\lambda_i}^{(k)} \end{cases}$$

□

The above lemma leads directly to the following corollary:

ALGORITHM 1. *For every tuple $\langle t_i \rangle_{i=1}^n$ such that $t_i$ is a segment on $P_i$:*

- *If $\sum_{i=1}^n e_{i,t_i} < q$ or $\sum_{i=1}^n s_{i,t_i} > q$:*
  *Continue; // Jump to the next $\langle t_i \rangle$ tuple.*
- *Sort $\{\pi_{i,t_i}\}$ increasingly.*
- *For $k = 1$ to $n$ do:*
  - *If $\sum_{i=1}^k e_{i,t_i} + \sum_{i=k+1}^n s_{i,t_i} > q$:*
    - *Set:*
      $$\begin{cases} r_i & = & e_{i,t_i}, \forall 1 \le i \le k-1 \\ r_i & = & s_{i,t_i}, \forall k+1 \le i \le n \\ r_k & = & q - \sum_{i=1, i \neq k}^n r_i \end{cases}$$
    - *End $k$ for loop.*
- *Compare $P(\langle r_i \rangle)$ with the price of the best allocation found so far.*

**Figure 1: Clearing algorithm for multi-unit single-item case with piece-wise linear supply function bids.**

COROLLARY 1. *The dominant set $D$ must contain an optimal allocation.*

LEMMA 2. *The number of elements in the set $D$ is not more than $\prod_{i=1}^n (N_i + 1)$.*

PROOF. For each tuple $\langle t_i \rangle_{i=1}^n$, in which $t_i$ is a segment on $P_i$, there exists at most one $k$,[12] so the number of elements in the set $D$ is not more than the number of such tuples. But the number of tuples $\langle t_i \rangle_{i=1}^n$ is $\prod_{i=1}^n (N_i + 1)$. Thus:

$$|D| \le \prod_{i=1}^n (N_i + 1)$$

□

With these lemmas in place, we can now present our algorithm for the single-item case (see figure 1). Basically, the algorithm searches through all the allocations of the set $D$ and chooses the most profitable valid one. We can now analyse the algorithm to assess its properties.

THEOREM 1. *The algorithm is guaranteed to find an optimal allocation.*

PROOF. The algorithm searches all the allocations of the dominant set $D$. Also, by corollary 1, the dominant set $D$ contains an optimal allocation. Thus the algorithm is guaranteed to find an optimal allocation. □

THEOREM 2. *The complexity of the algorithm is $O(n \cdot (K+1)^n)$, where $K$ is the upper bound on the number of segments of $P_i$.*

PROOF. The number of allocations searched by the algorithm is equal to the number of elements of the dominant set. By lemma 2, the number of elements of the dominant set is not more than $\prod_{i=1}^n (N_i+1) \le (K+1)^n$. Also, it takes $O(\log n)$ to sort $\{\pi_{i,t_i}\}$ and $O(n)$ to find $k$, so the complexity of the algorithm is $O(n \cdot (K+1)^n)$. □

[12] There may be more than one $k$, for example, in the case where $s_{i,t_i} = e_{i,t_i}$ for every $i$, but in such cases, it does not matter which $k$ is chosen.

Having dealt with the multi-unit single-item case, the next section generalises the algorithm to the multi-unit combinatorial case.

## 3.2 MULTI-UNIT COMBINATORIAL ITEMS

As before, we define a dominant set that is proved to contain an optimal allocation.

*Definition 5.* The dominant set $D$ is the set of all allocations $\langle r_i^j\rangle$ such that for every $1 \le j \le m$, there exists a $k_j$, $1 \le k_j \le n$, such that all $r^j_{\lambda^j_1}, ..., r^j_{\lambda^j_{k-1}}$ equal the ending quantities of the segments that they belong to, and all $r^j_{\lambda^j_{k+1}}, ..., r^j_{\lambda^j_n}$ equal the starting quantities of the segments that they belong to :[13]

$$\begin{cases} r^j_{\lambda^j_i} &= e^j_{\lambda^j_i, t^j_{\lambda^j_i}}, \forall 1 \le i \le k-1 \\ r^j_{\lambda^j_i} &= s^j_{\lambda^j_i, t^j_{\lambda^j_i}}, \forall k+1 \le i \le n \\ r^j_{\lambda^j_k} &= q_j - \sum_{i=1, i\neq k}^n r^j_{\lambda^j_i} \end{cases}$$

where:

- $t_i^j$ is the segment on $P_i^j$ that $r_i^j$ belongs to.
- $(\lambda_i^j)_{i=1}^n$ is any permutation of $(1,2,...,n)$ such that $\{\omega_{\lambda^j_i}(\langle t^j_{\lambda^j_i}\rangle)\cdot \pi^j_{\lambda^j_i, t^j_{\lambda^j_i}}\}_{i=1}^n$ is sorted increasingly:

$$\omega_{\lambda^j_1}(\langle t^j_{\lambda^j_1}\rangle)\cdot \pi^j_{\lambda^j_1, t^j_{\lambda^j_1}} \le \omega_{\lambda^j_2}(\langle t^j_{\lambda^j_2}\rangle)\cdot \pi^j_{\lambda^j_2, t^j_{\lambda^j_2}} \le ... \le \omega_{\lambda^j_n}(\langle t^j_{\lambda^j_n}\rangle)\cdot \pi^j_{\lambda^j_n, t^j_{\lambda^j_n}}$$

From this, a number of lemmas follow:

LEMMA 3. *For every allocation $\langle r_i^j\rangle$ there exists an allocation in the dominant set $D$ that is not less profitable than it.*

PROOF. Let $\langle r_i^j\rangle$ be an allocation. Let $t_i^j$ be the segment that $r_i^j$ belongs to. Suppose $(\lambda_i^j)_{i=1}^n$ is any permutation of $(1,2,...,n)$ such that $\{\omega_{\lambda^j_i}(\langle t^j_{\lambda^j_i}\rangle)\cdot \pi^j_{\lambda^j_i, t^j_{\lambda^j_i}}\}_{i=1}^n$ is sorted increasingly:

$$\omega_{\lambda^j_1}(\langle t^j_{\lambda^j_1}\rangle)\cdot \pi^j_{\lambda^j_1, t^j_{\lambda^j_1}} \le \omega_{\lambda^j_2}(\langle t^j_{\lambda^j_2}\rangle)\cdot \pi^j_{\lambda^j_2, t^j_{\lambda^j_2}} \le ... \le \omega_{\lambda^j_n}(\langle t^j_{\lambda^j_n}\rangle)\cdot \pi^j_{\lambda^j_n, t^j_{\lambda^j_n}} \quad (4)$$

For any $\bar{j}$, $1 \le \bar{j} \le m$, by proving in similar manner to lemma 1, there exists an allocation $\langle \bar{r}_i^j\rangle$, that is not less profitable than $\langle r_i^j\rangle$, where $\bar{r}_i^j$ belongs to segment $t_i^j$ of $P_i^j$, $\forall 1 \le i \le n$, $\forall 1 \le j \le m$ and for some $k$, $1 \le k \le n$:

$$\bar{r}^{\bar{j}}_{\lambda^{\bar{j}}_i} = e^{\bar{j}}_{\lambda^{\bar{j}}_i, t^{\bar{j}}_{\lambda^{\bar{j}}_i}}, \forall 1 \le i \le k-1$$

$$\bar{r}^{\bar{j}}_{\lambda^{\bar{j}}_i} = s^{\bar{j}}_{\lambda^{\bar{j}}_i, t^{\bar{j}}_{\lambda^{\bar{j}}_i}}, \forall k+1 \le i \le n$$

$$\bar{r}^{\bar{j}}_{\lambda^{\bar{j}}_k} = q_j - \sum_{i=1, i\neq k}^{n} \bar{r}^{\bar{j}}_{\lambda^{\bar{j}}_i}$$

$$\bar{r}_i^j = r_i^j, \forall 1 \le i \le n, \forall 1 \le j \le m, j \neq \bar{j}$$

Repeating the above step for every $\bar{j}$ from 1 to $m$, we complete the proof. □

[13]Similar to section 3.1, there may be many dominant sets $D$.

> ALGORITHM 2. *For every tuple $\langle t_i^j\rangle$, $1 \le i \le n$, $1 \le j \le m$ such that $t_i^j$ is a segment on $P_i^j$:*
>
> - *For every $j = 1$ to $m$ do:*
>   - *If $\sum_{i=1}^n e^j_{i,t^j_i} < q_j$ or $\sum_{i=1}^n s^j_{i,t^j_i} > q_j$:*
>     *Continue; // Jump to the next $\langle t_i^j\rangle$ tuple.*
>   - *Sort $\{\omega_i(\langle t_i^j\rangle)\cdot \pi^j_{i,t^j_i}\}$ increasingly.*
>   - *For $k = 1$ to $n$ do:*
>     - *If $\sum_{i=1}^k e^j_{i,t^j_i} + \sum_{i=k+1}^n s^j_{i,t^j_i} > q_j$:*
>       - *Set:* $\begin{cases} r_i^j &= e^j_{i,t^j_i}, \forall 1 \le i \le k_j - 1 \\ r_i^j &= s^j_{i,t^j_i}, \forall k_j+1 \le i \le n \\ r^j_{k_j} &= q_j - \sum_{i=1, i\neq k_j}^n r_i^j \end{cases}$
>       - *End $k$ for loop.*
> - *Compare $P(\langle r_i^j\rangle)$ with the price of the best allocation found so far.*

**Figure 2: Clearing algorithm for multi-unit combinatorial case with piece-wise linear supply function bids.**

The above lemma leads directly to the following corollary:

COROLLARY 2. *The dominant set $D$ must contain an optimal allocation.*

LEMMA 4. *The number of elements in the set $D$ is not more than $\prod_{i=1}^n \prod_{j=1}^m (N_i^j + 1)$.*

PROOF. Consider an allocation $\langle r_i^j\rangle$ in $D$. By lemma 2, for each $\bar{j}$ ranging from 1 to $m$, the number of possible values of a tuple $\langle r_i^{\bar{j}}\rangle_{i=1}^n$ is not more than $\prod_{i=1}^n (N_i^{\bar{j}} + 1)$. Thus, the number of possible values of $\langle r_i^j\rangle$ is not more than $\prod_{i=1}^n \prod_{j=1}^m (N_i^j + 1)$. □

With these lemmas in place, we can now present our algorithm for the combinatorial case (see figure 2), which, as before, searches through all allocations of the dominant set $D$ and chooses the most profitable valid one. We can now analyse the algorithm to assess its properties.

THEOREM 3. *The algorithm is guaranteed to find the optimal allocation.*

PROOF. Same as that of theorem 1. □

THEOREM 4. *The complexity of the algorithm is $O(mn \cdot (K+1)^{mn})$, where $K$ is the upper bound on the number of segments of $P_i^j$.*

PROOF. Same as that of theorem 2. □

Note that this is a worst-case analysis. In many real-life scenarios, each bidder is likely to provide only a strict subset of the set of goods/services, not all of them. So if bidder $i$ does not provide an item $j$, then $N_i^j = 0$, meaning the number $\prod_{i=1}^{n}\prod_{j=1}^{m}(N_i^j+1)$ is much smaller than $(K+1)^{mn}$. For example, given the values suggested in [4] (that are claimed to resemble real-life problems in the domain of e-commerce), the complexity of our algorithm reduces to $O(mn \cdot 3^{\frac{m(n+4)}{2}})$. While this is certainly not an average case analysis, it provides an indication of the complexity that may be encountered in practice.

## 4. RELATED WORK

As noted in section 1, most of the work on auction clearing to date has concentrated on the atomic proposition case which is neither as compact nor as economically efficient as demand/supply bidding. However, some work has examined more general settings. For instance, [9] considered multi-unit single-item auctions with bids in the form of supply/demand curves. By limiting these curves to a specific type (linear and piecewise linear curves[14]), they were able to analyse the complexity and suggest an algorithm for clearing.[15] However, this work does not deal with the multi-unit combinatorial case.

Other researchers, such as [3] and [4], have further considered multi-unit combinatorial reverse auctions with supply curves. They showed that in the case where the supply curves are piecewise linear, the clearing problem can be modelled as a Linear Program and solved using Linear Programming techniques. However, in this work, bidders submit separate supply curves for different items, and they assumed additive separability. This means that their auctions are not truly *combinatorial* in nature as the correlation between items is ignored.[16] [2] have developed algorithms for multi-unit combinatorial reverse auctions with demand/supply function bidding when the bidding functions exhibit specific properties (free disposal and sub-additive pricing). These settings are truly combinatorial as the bidders submit the bidding functions for combinations of items. The algorithms run in polynomial time and produce solutions that are shown to be within a finite bound of the optimal. However, they are not guaranteed to find the optimal allocation.

## 5. CONCLUSIONS AND FUTURE WORK

This paper presents, for the first time, optimal clearing algorithms for multi-unit single-item and multi-unit combinatorial auctions where bids are expressed through supply/demand functions. Specifically, we consider the class of supply/demand functions where the demand/supply curves for each individual commodity are piece-wise linear (an important and often considered case). This means our algorithms enable us to deal with a more general case than any previous work in this area. Moreover, we believe this degree of expressiveness is important for obtaining the maximum benefit from combinatorial auctions in practical settings. For the future, we aim to evaluate the algorithms empirically with real-life scenarios and to reduce the complexity of the algorithms which, we believe, can be achieved using standard combinatorial search techniques such as Branch-and-Bound and heuristics.

[14]Their concepts of linear and piecewise linear curves are different from ours, as they consider the unit price function, not the total price function. Thus, when they speak of a linear unit price function, this means a quadratic total price function.

[15]They provided an algorithm for the linear case only, not for the piecewise linear case.

[16]Note that in the case where the assumption of additive separability is adopted, it is possible to use our single-item algorithm to clear the auction, by repeating the algorithm for every item. This gives an optimal allocation algorithm whose complexity is $O(mn \cdot (K+1)^n)$.

## 6. REFERENCES


[1] V. D. Dang and N. R. Jennings. Optimal clearing algorithms for forward and reverse combinatorial auctions with demand/supply function bidding. Technical report, University of Southampton, 2002.

[2] V. D. Dang and N. R. Jennings. Polynomial algorithms for clearing multi-unit single item and multi-unit combinatorial reverse auctions. In *Proc. 15th European Conf. on AI*, pages 23–27, 2002.

[3] A. Davenport and J. Kalagnanam. Price negotiations for procurement of direct inputs. IBM Research Report RC 22078, 2001.

[4] M. Eso, S. Ghosh, J. Kalagnanam, and L. Ladanyi. Bid evaluation in procurement auctions with piece-wise linear supply curves. IBM Research Report RC 22219, 2001.

[5] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and ap-proximate approaches. In *Proc. 16th Intl. Joint Conf. on AI*, pages 548–553, 1999.

[6] R. Gonen and D. Lehmann. Optimal solutions for multi-unit combinatorial auctions: Branch and bound heuristics. In *Proc. 2nd ACM Conf. on Electronic Commerce*, pages 13–20, 2000.

[7] N. Nisan. Bidding and allocation in combinatorial auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce*, pages 1–12, 2000.

[8] T. Sandholm. An algorithm for optimal winner determination in combinatorial auctions. In *Proc. 16th Intl. Joint Conf. on AI*, pages 542–547, 1999.

[9] T. Sandholm and S. Suri. Market clearability. In *Proc. of 17th Intl. Joint Conf. on AI*, pages 1145–1151, 2001.

[10] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Winner determination in combinatorial auction generalizations. In *Proc. 1st Intl. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 69–76, 2002.