Combinatorial Markets for Efficient Energy Management

Yoseba K. Penya
Department of Information Systems
Vienna University of Economics and
Business Administration, Austria.
yoseba.penya@wu-wien.ac.at

Nicholas R. Jennings
School of Electronics and Computer Science
University of Southampton
Southampton, UK.
nrj@ecs.soton.ac.uk

Abstract

The deregulation of the electricity industry in many countries has created a number of marketplaces in which producers and consumers can operate in order to more effectively manage and meet their energy needs. To this end, this paper develops a new model for electricity retail where end-use customers choose their supplier from competing electricity retailers. The model is based on simultaneous reverse combinatorial auctions, designed as a second-price sealed-bid multi-item auction with supply function bidding. This model prevents strategic bidding and allows the auctioneer to maximise its payoff. Furthermore, we develop optimal single-item and multi-item algorithms for winner determination in such auctions that are significantly less complex than those currently available in the literature.

1. Introduction

The deregulation of electricity markets began in the early nineties when the UK Government privatised the electricity supply industry in England and Wales. This process was then subsequently followed in many other countries. In most cases, this restructuring involves separating the electricity generation and retail from the natural monopoly functions of transmission and distribution. This, in turn, leads to the establishment of a *wholesale electricity market* for electricity generation and a *retail electricity market* for electricity retailing. In the former case, competing generators offer their electricity output to retailers and in the latter case end-use customers choose their supplier from competing electricity retailers. Here we focus on retail markets, which differ from their more traditional counterparts because energy cannot be stored or held in stock (as tangible goods can). Consequently, retailers are forced to work with consumption prognoses, which, in turn, creates a number of risks. First, producing more than is consumed is not economical. Moreover, the price of the energy mainly depends on the production cost and this typically rises with the amount of energy produced. Second, if the demand exceeds the prediction, suppliers must find additional energy to avoid a blackout. Finally, there are non-negligible costs stemming from the variation in the electricity production volume that most of the traditional types of energy generators (e.g. hydroelectric, thermoelectric, nuclear) have to face. Against this background, the desideratum is to achieve a market model where retailers have the most accurate possible prognosis and the capability of influencing or guiding customers' consumption. To this end, there have been a number of initiatives, grouped under the general banner of *Demand-Side Management* (DSM), whose main objective is to distribute the demand over time to avoid peak loads. Now, the easiest way to achieve this goal is by setting the price of the energy depending on the actual demand load. Thus, the higher the demand, the more expensive the price, and vice versa. Based upon these premises, many utility companies (UCs) already present a basic form of DSM by offering a cheaper night tariff.

Our aim in this work is to improve and extend this simple market model to permit UCs to express more complex aims and, thus, increase their influence on customers. For instance, in order to lighten the peak-time load, the supplier can offer a discount for consuming a small amount of energy at 8 am (peak-time) and a larger amount at midnight (off peak). This incentivises the customer to reschedule some tasks to midnight (e.g. the dishwasher or the washing machine). If many clients accept this compromise offer, the UC will have achieved a double goal. It will have a more accurate prognosis for 8 am and midnight and it will also have shifted some of the peak-time consumption to off peak. In e-commerce terms, this process can be seen as a *reverse combinatorial auction*. It is "reverse" because the customers pick one of the available companies and tariffs to supply their future consumption. And it is "combinatorial" because bidding for a bundle of items is typically valued differently from bidding separately for each of the constituent items (e.g. the *combination* of consuming at 8am and midnight is more appreciated, and thus rewarded, than, for instance, the combination of consuming at 10am and 11am).

While combinatorial auctions provide very efficient allocations that can maximise the revenue for the auctioneer, their main drawback is the complexity of the *clearing* process in which buyers and sellers are matched and the quantities of items traded between them are determined. Specifically, clearing combinatorial auctions is NP-hard [4]. Moreover, most work in this area deals with clearing combinatorial auctions with *atomic propositions* [6]. Thus, bids are either accepted or rejected in their entirety, which may limit the profit for the auctioneer. A more efficient solution is to allow bidding with demand/supply functions [7, 2], in which bidders submit a function to calculate the cost of the units to be bought or sold. This allows the customer to accept parts of different bids and constitutes a powerful way of expressing complex pricing policies. In our case, production costs can be easily reflected in the supply function and if bids are accepted partially, there may be more than one winner for the same auction and item. This enables customers to accept different parts of bids from different bidders so they can get energy simultaneously from several suppliers. Since the transmission and distribution grids are shared and the path followed by the electricity cannot be tracked down, it is impossible to determine the producer of the energy being consumed. Therefore, the hypothesis of customers being simultaneously supplied by several UCs does not pose any technical problems.

Against this background, this paper advances the state of the art in two main ways. First, we present, for the first time, an energy retail market designed as a system of reverse combinatorial auctions with supply function bidding. This novel market allows customers to increase their profit and provides UCs with a mechanism to influence customers' behaviour. Second, we develop new optimal clearing algorithms tailored to electricity supply functions that perform better than the existing more general clearing algorithms. The remainder of the paper is organized as follows. Section 2 details the overall market design. Section 3 presents the single-item and the multi-item clearing algorithms and analyses their complexity and optimality. Section 4 examines the results of comparing the multi-item algorithm with the only other optimal multi-item one defined in the literature. Section 5 discusses related work. Finally, Section 6 concludes and outlines the avenues of future work.

2. Electricity Retail Markets

This section discusses the nature of current electricity retail markets and outlines the design of our solution.

2.1. Requirements

Currently, most customers only partially enjoy the benefits of a deregulated market. They typically sign mid-term contracts with a single supplier and the tariffs do not reflect the pressure of competition. Moreover, whereas classical capitalist pricing policies encourage demand by applying discounts on quantity (the more you buy, the cheaper the unit price becomes), actual electricity contracts often include a threshold above which the consumption becomes more expensive. To move to a more dynamic environment where the benefits of competition can be more fully realised, we put forward the following requirements for our market design. The arrangement of customers' electricity supply from multiple UCs should be achieved by having contracts that specify the provision of an amount of energy for a certain period of time (say one hour). These contracts should not necessarily be exclusive and, thus, customers may have agreements with different companies for the same hour if this is the best thing to do. Finally, we assume customers auction, on a daily basis, their next 24 hours consumption divided into 24 items (representing one hour each). They subsequently receive bids from the UCs and make their decision for the next 24 hours, which is a trade-off between the very static situations of today and the possibility of auctioning on a per minute basis for the coming minute.

2.2. Market Design

The requirements detailed above can be best met by structuring the market as a reverse auction. Furthermore, we assume customers don't issue any bids but simply choose among those offered by the UCs. An exchange (in which multiple buyers and sellers submit their bids and offers to an independent auctioneer that decides the winners [6]) was rejected because it scales poorly. In practice, the number of customers may be up to tens of thousands, each of which is selling 24 items, and with combinatorial bidding, clearing such an exchange becomes intractable very fast. Unlike exchanges, reverse auctions have the advantage that they may be performed in parallel. This means the complexity can be divided between the number of customers because instead of one big auction, many *smaller* ones are carried out at the same time. For these reasons, we have designed our system as a series of simultaneous reverse auctions despite the risk of *overbooking*. That is, although the UCs issue their own tariffs they cannot control the number of customers that choose them. So the demand could exceed their capability. However, in this case, we allow overbooked UCs to buy the additional energy from non-overbooked ones (also see section 6). As combinatorial bidding is permitted, UCs submit their *special* discounts together with the usual hour tariffs. In this case, having 24 hours (or items) means that there may be up to 2^{24} different combinations of discounts. This is obviously a worst-case scenario because, in practice, our experience in the domain indicates that UCs are highly unlikely to issue a different discount for each possible combination. Moreover, we decided that the auctions should be sealed (to reveal the least possible information) and single-round (to minimise communication and other delays). The auctions also need to be both multi-item and multi-unit. As each item is the supply of electricity in one hour, there are 24 items to allocate in an auction. In addition, each bidder may not allocate the whole consumption within an hour but rather just a portion of it (i.e. some units).

Another important component to set is the price paid by the winner. We do not want to have a first price auction because it offers incentives for strategic behaviour (i.e. the participants act according to beliefs formed about others' values and types, which does not assure them of maximising their payoff). To circumvent this, we choose a uniform second price for combinatorial auctions (Vickrey-Clarke-Groves) since this has the dominant strategy of bidders bidding their true valuations of the goods [5, 3]. The price paid by the winner is not directly specified in the bid because bidders submit a supply function. Thus, the customer must calculate the energy it wants to consume within a time slot (i.e. the units of that item to be auctioned) and then decide the cheapest combination with the supply functions submitted (i.e. number of units to be allocated with each bidder. Therefore, the bids are accepted partially. To this end, we use the compact notation introduced in [2], where bidders submit for a certain item a piece-wise linear supply function P composed of n linear segments. Each segment $l, 1 \leq l \leq n$, is described by a starting quantity s_l, an ending quantity e_l, a unit price π_l, and a fixed price C_l. Thus, if a customer wants to buy q units of that item from the supplier, it will pay $P_l = \pi_l \cdot q + C_l$ if $s_l \leq q \leq e_l$. Additionally, bidders submit a correlation function, ω, which shows the reward or penalty of buying a number of items together (it is this that makes the bidding truly combinatorial). For instance, $\omega_1(A, B) = 0.95$ would mean that if buying x units of items A and y units of B (i.e. consuming x Kw at time A and y Kw at time B), the price paid will have a 5% discount. Thus, if the unit price of item A is p_a and the unit price of item B is p_b, the total price would be $0.95 \cdot ((x \cdot p_a) + (y \cdot p_b))$.

Currently, there is only one optimal algorithm to solve this problem. Specifically, the one presented by Dang and Jennings in [2] (described in more detail in section 5). However, we believe this is inapplicable in our scenario because it scales poorly (as we show in section 4). Therefore, with the market described above in place, the next step is to design a clearing algorithm that solves the winner determination problem more efficiently and allows it to be actually applied in realistic contexts.

3. Optimal Clearing Algorithms

This section details the optimal single-item (sPJ) and the optimal multi-item clearing algorithm (mPJ) that we have developed for the electricity retail market described in section 2. Furthermore, we analyse their complexity, prove their optimality, and analyse strategies to keep them tractable. First of all, let us introduce some basic definitions that will be used thereafter:

Definition 1 *A **single allocation** is a set <time-slot t, supplier s, amount q, price p> meaning that s wants to pay p to buy q units of energy to be consumed at time t.*

Definition 2 *An **allocation** is a list containing a number (between one and the number of suppliers) of single allocations that detail the supply of electricity to be provided to the customer at a given time-slot.*

Definition 3 *A **more profitable allocation** from two alternatives is the one that for a given total demand q, has the lower total price p.*

Definition 4 *An **optimal allocation** is one in which the demand constraint is satisfied and there is no more profitable allocation.*

Definition 5 *An **optimal day allocation** is a set of 24 optimal allocations, each of which corresponds to a different item (i.e. there is an optimal allocation for each hour).*

The clearing algorithms we present in this section are related in that the multi-item one is a consecutive and iterative processing of the single-item one (i.e. the result of the multi-item algorithm is obtained by executing the single-item one with different values). Specifically, clearing a single-item case implies finding the optimal allocation for that item, so this enterprise deals only with the supply functions submitted to one item. The multi-item case has a broader remit (an optimal day allocation) and, thus, it also takes into account the relationships between the different items of the optimal allocations (i.e. the correlation functions). Let us first start with the explanation of the single-item case.

3.1. The Optimal Single-Item Clearing Algorithm

Clearing a single-item algorithm with piece-wise supply function bids involves determining the amount to be allocated to each submitted bid function. In essence, in each loop the algorithm selects one segment of each supply function (the one corresponding to the already allocated demand) and allocates k units to the segment with the best price (i.e. the lowest price for k units after applying any relevant discount on the amount). The loop is repeated until the demand is satisfied. Note that the value of k is dynamically assigned in each loop to guarantee the optimality of the algorithm. Specifically, it always has the ending quantity value (e_l) of the shortest segment being evaluated at that moment.

Let us now illustrate this procedure with the example of Figure 1. Assume there are three potential buyers 1, 2, and 3 that submit their supply functions $s1$, $s2$, and $s3$ for a certain item (i.e. the consumption in one hour). In the first loop, the algorithm processes the segments $s1_1$, $s2_1$, and $s3_1$. Since the shortest of the three is segment $s1_1$(i.e. $e_1^1 < e_1^2 < e_1^3$), $k = e_1^1$ and the algorithm compares $s1_1(e_1^1)$, $s2_1(e_1^1)$, and $s3_2(e_1^1)$. Suppose the price of $s3_1(e_1^1)$ is less than the price of $s1_1(e_1^1)$ and $s2_1(e_1^1)$; then, it selects $s3_1$ to supply these first e_1^1 units. In the second loop, the algorithm processes the segments $s1_1$, $s2_1$, and $s3_1$(but starting from e_1^1) and gives k the value of $e_1^3 - e_1^1$ because it is less than e_1^1 and e_1^2. Then,

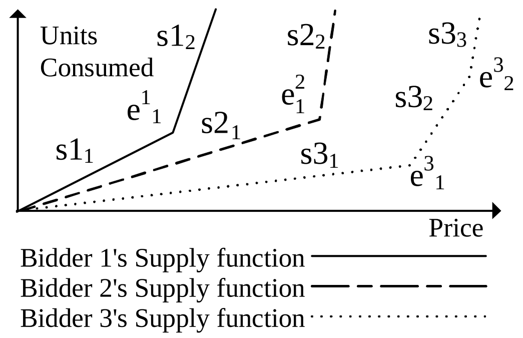


Figure 1. Linear piece-wise supply functions submitted to a single item.

it compares $s1_1(e_1^3 - e_1^1)$, $s2_1(e_1^3 - e_1^1)$, and $s3_1(e_1^1)$, and so on. The algorithm continues until the amount of allocated units is equal to the demand.

As we can see, the algorithm evaluates one function per bidder in each step so it has a complexity $O(m)$ per loop, where m is the number of bidders. As the loop is repeated k times, where k is the number of segments of the function with the highest number of them, the overall complexity is $O(km)$. A safe way to reach an optimal allocation is to select for each unit the segment that offers the best price (i.e. $k = 1$). However, it is not necessary to repeat the process for each single unit since price and discount are constant in each segment. So, as long as the segments evaluated in each loop are the same (unit price and fixed price remain unchanged), the winner will also be the same. Thus, in each loop it is only necessary to compare the price of allocating the lowest ending quantity of the segments being processed, repeating this process until the demand is satisfied. Therefore, sPJ (detailed in Figure 2) always finds the most profitable optimal allocation.

3.2. The Optimal Multi-Item Clearing Algorithm

This algorithm, detailed in Figure 3, is more complex since it cannot simply be generalised from the single-item one. If there were no correlations, it would be sufficient to run the sPJ case once for each item. However, the existence of correlations poses the problem of the inconsistent application of discounts. First, if a supplier bids for two items and offers a reduction if both bids get accepted, no reduction should be applied if only one of them succeeds. Second, functions become different after applying a discount. For example, assume P_l is a piece-wise supply function for the item l and it is included in the correlation $\omega(l, ...) = x$. Then, P_l' is the new supply function with the value $P_l' = xP_l$. Thus, the optimal allocation of a set of functions in which P_l is included may not be the same as the one in which everything else is the same but with P_l' instead of P_l.

In this way, mPJ must process all possible combinations of discounted and non-discounted functions and check that discounts are applied consistently. To this end, we use a

Input: m supply functions f and $demand$.

• Pre-loop: initialise needed variables: $allocated$ to keep the total allocated demand, the list $allocation$ showing the allocated demand per bidder, and the temporal storage variable k.

• Loop: in each loop, until the demand is satisfied, select the segment with the lowest gradient and allocate the minimum ending quantity units.

```
While (allocated $<$ demand) do
  k = select the minimum ending quantity
  if (demand - allocated $<$ k) then
    winner = select the minimum $f_m$(k)
    allocated += k
    allocation[winner] += k
  else
    winner = select the $f_m$ with lowest gradient
    allocated += demand - allocated
    allocation[winner] += demand - allocated
```

Output: $allocation$, the variable detailing the amount allocated to each bidder.

Figure 2. The sPJ clearing algorithm.

brute-force strategy for identifying all the possibilities. Here, all possible bids from each bidder are combined with all possible bids from the rest of the bidders. However, it is not necessary to evaluate all the combinations since some of them are repeated. For instance, Table 1 shows an auction with two suppliers (1 and 2) and two items (a and b). In this case, there is one possible correlation for each bidder, $\omega^1(a, b) = x$ and $\omega^2(a, b) = y$. Thus, clearing the multi-item case implies evaluating the combinations where supplier 1 and 2 bid normally for item a (so the single-item clearing algorithm is run with supply functions $P_a^1 - P_a^2$); supplier 1 bids for item a and b with discount and supplier 2 bids normally for item b (so the single-item algorithm clears item a with supply function xP_a^1 and item b with xP_b^1 and P_b^2), and so on.

Table 1. Single-item evaluations with two items and two bidders, repeated in bold.

	P_a^2	yP_a^2 yP_b^2	P_b^2	-
P_a^1	$P_a^1 - P_a^2$	$P_a^1 - yP_a^2$ $\boldsymbol{yP_b^2}$	$\boldsymbol{P_a^1}$ $\boldsymbol{P_b^2}$	$\boldsymbol{P_a^1}$
xP_a^1 xP_b^1	$xP_a^1 - P_a^2$ $\boldsymbol{xP_b^1}$	$xP_a^1 - yP_a^2$ $xP_b^1 - yP_b^2$	$\boldsymbol{xP_a^1}$ $xP_b^1 - P_b^2$	$\boldsymbol{xP_a^1}$ $\boldsymbol{xP_b^1}$
P_b^1	$\boldsymbol{P_a^2}$ $\boldsymbol{P_b^1}$	$\boldsymbol{yP_a^2}$ $P_b^1 - yP_b^2$	$P_b^1 - P_b^2$	$\boldsymbol{P_b^1}$
-	$\boldsymbol{P_a^2}$	$\boldsymbol{yP_a^2}$ $\boldsymbol{yP_b^2}$	$\boldsymbol{P_b^2}$	-

Input: j supply functions s_j, j correlation functions ω_j and demand q_i for each item i.

• Pre-loop: Initialise variable *day-set* to keep the optimal allocation for each item, *item-set* to keep a group of supply functions to be evaluated by the single-item clearing algorithm, $all - item - sets$ to keep already processed sets of supply functions, and a boolean variable ok.

• Loop: For each item calculate the optimal allocation of a possible set of supply functions and then check whether the selected discounts are applicable.

```
Do
  for each item i
    for each supplier $s_j$
      add next $s_j^i$ to item-set
    if item-set not in all-item-set then
      optimal-allocation = single_item_algorithm(item-set)
      store item-set in all-item-sets
    add optimal-allocation to day-set
  ok = check constraints (day-set, $\omega_j$).
  If ok then compare day-set with best so far
  until all the combinations are explored
```

Output: *day-set*, a set of i optimal allocations (one for each item) with the lowest total price.

Figure 3. The mPJ clearing algorithm.

This brute-force strategy evaluates all possible bid combinations (without repeating some of them) and, therefore, it always finds the most profitable optimal day allocation. However, it also scales poorly. First, the number of possible combinations depends on the number of items. In our case, with 24 items, there are 2^{24} different combinations. Second, it also rises exponentially as the number of bidders grows: with n items, and two bidders, 2^{2n}; with three bidders 2^{3n}, and so on. In the extreme situation with two bidders submitting a different supply function for each one of the 24 items and 2^{24} correlations, there are $2 \cdot 2^{48}$ possible combinations. This is, $n \cdot (2^n)^m$, where n is the number of items and m the number of bidders. For instance, in the example of Table 1, there are $2 \cdot (2^2)^2 = 32$ possible combinations, but half of the combinations do not need to be re-calculated (in bold format in Table 1). Thus, if bidders bid for all items and submit all possible correlations, the number of times that the multi-item algorithm clears the single-item one is $n \cdot (2^n - 2^{n-1})^m = n \cdot (2^{n-1})^m$. Therefore, the complexity is $O(kmn \cdot 2^{(n-1)\cdot m})$, where n is the number of items, m the number of suppliers and k the number of segments of the supply function with more segments. Note, however, that this is a pathological worst-case scenario, which is highly unlikely to happen in practice. Furthermore, as we discuss bellow, it can be mitigated against by constraining the agent's bidding behaviours.

3.3. Constrained Bidding

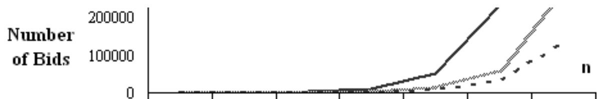
In order to prevent such pathological scenarios from occurring, it is possible to constrain the choice of possible discounts so bidders can submit only a certain number of correlations. This type of restriction has already been successfully applied to atomic propositions bidding, where when limiting the allowable combinations to tree structures or sequential combinations, the NP-hard winner determination problem can be solved in polynomial time [6]. In a similar vein, mPJ can also take advantage of such an approach. Specifically, we can constrain the number of correlations to a value c. Thus, bidder 1 can issue, for instance, the following: $\omega_1^1(n_1, n_2 \ldots n_i)$, $\omega_2^1(n_1, n_2 \ldots n_i)$ $\ldots \omega_c^1(n_1, n_2 \ldots n_i)$ where i is the number of items included in each discount (for the sake of simplicity, let us suppose it is a fixed number less than n, the number of items, but big enough to allow the bidder to be sufficient flexible in its offering).

In this way, the single-item algorithm sPJ will be executed, with m bidders, $i \cdot c^m$ times (again, supposing that i is fixed) and the complexity of the mPJ algorithm will drop to $O(ki \cdot c^m)$. Unfortunately, in this case, the mPJ algorithm cannot skip evaluating half of the combinations (as in section 3). With this constrained discount choice, the reduction depends much more on the specific discount combinations chosen. For instance, if the combinations include many items (i.e. i is bigger), the single-item algorithm will be executed more often than if the combinations only include two items each. In short, there is no way to accurately determine it *a priori*. Similarly, restricting the available amounts assigned to the discount increases the number of repeated combinations. Thus, if a supplier offers the same reduction for accepting two different items (e.g. $\omega(a, b) = \omega(c, d)$), the number of repeated combinations would increase further and the complexity would continue decreasing.

4. Evaluation of the mPJ clearing Algorithm

In this section we present the results of comparing the performance of our mPJ algorithm (introduced in section 3.2) with the only other optimal algorithm for this class of problem. Specifically, our benchmarks are the algorithm mDJ presented by Dang and Jennings in [2] (described in more detail in section 5, hereafter referred to as "sDJ" for the single-item one and "mDJ" for multi-item) and the constrained bidding variant of our algorithm (as detailed in section 3.3). In this later algorithm, we have set the maximum number of bids to be issued as half of the maximum possible ($c = 2^{n-1}$) and the maximum number of items included in a correlation as the number of items ($i = n$).

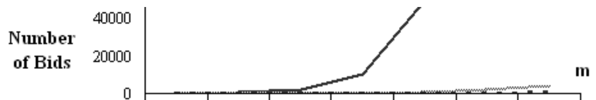
The comparison shown in Figure 4 details how the complexity (defined in terms of X, the number of bids) scales when the number of items n increases for a constant number of bidders m. As can be seen, mDJ soon becomes intractable (i.e. prohibitively high complexity), mPJ scales better, and the constrained variant presents the best profile for our purposes. This would have been even clearer if we had not set the value of c and i depending on the number of items n (as detailed above). With a fixed c and i, the constrained variant would had presented a flat line, whereas mDJ and mPJ would had grown exponentially because in contrast to mDJ and mPJ, the constrained variant does not depend directly on the number of items being auctioned.



	2	3	4	5	6	7	8
mDJ	64	384	2048	10240	49152	229376	1048576
mPJ	16	96	512	2560	12288	57344	262144
Constr. bid	8	48	256	1280	6144	28672	131072

Figure 4. Complexity evolution with n increasing and m steady ($m = 2$).

Figure 5 tests how the algorithms react to the increment of m (bidders) when n (items) remains steady. Again, mDJ becomes intractable as soon as it did in Fig. 4, whereas mPJ and its constrained variant present a significantly better performance profile. The main reason for this behaviour is the sensitivity of mDJ to the increment of both n and m (while mPJ is only sensitive to the increment of n, as seen in Fig. 4). For mDJ, a larger number of items and clients means a larger number of single-allocations to form the set from which the allocations will be formed. Whereas for mPJ, more clients means more correlations to clear, but half of which need not be processed since they are repeated.

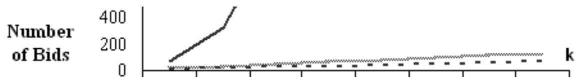


	2	3	4	5	6	7	8
mDJ	64	384	2048	10240	49152	229376	1E+06
mPJ	16	48	128	320	768	1792	4096
Constr. bid	8	16	32	64	128	256	512

Figure 5. Complexity evolution with m increasing and n steady ($n = 2$).

Similarly, Figure 6 illustrates the behaviour of the algorithms when both n (items) and m (bidders) increase. Again, mDJ performs worse than the others. Its $n = 2$ series is almost equivalent to the $n = 3$ of our multi-item algorithm. The best results are again achieved by the constrained variant (as we would expect).

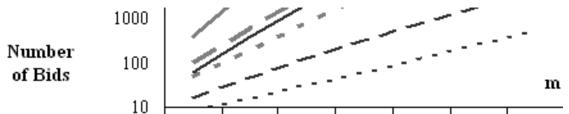
Finally, Figure 7 depicts the dependence of each algorithm on k, the number of units allocated in each iteration of the single-item algorithm. In this dimension both our algorithm and its constrained variant perform again well. For mDJ, increasing k implies increasing the number of single allocations that may be combined with each other (therefore



	1	2	3	4	5	6	7	8
mDJ	64	324	1024	2500	5184	9604	16384	26244
mPJ	16	32	48	64	80	96	112	128
Constr. Bid	8	16	24	32	40	48	56	64

Figure 6. Complexity evolution with n and m increasing.

the algorithm grows exponentially with k as the base). In contrast, for mPJ increasing k just implies that the single-item algorithm is going to process more steps (therefore the algorithm grows linearly with k as the factor).



	2	3	4	5	6	7	8
mDJ n=2	64	384	2048	10240	49152	229376	1E+06
mDJ n=3	384	4608	49152	491520	5E+06	4E+07	4E+08
mPJ n=2	16	48	128	320	768	1792	4096
mPJ n=3	96	576	3072	15360	73728	344064	2E+06
Constr. Bid n=2	8	16	32	64	128	256	512
Constr. Bid n=3	48	192	768	3072	12288	49152	196608

Figure 7. Complexity evolution with n and m steady and k increasing ($n, m = 2$).

Note that the complexity of the constrained variant can be further reduced depending on the values of i and c. With the values, we assigned to i and c for these comparisons, it is only m times less complex than mPJ (since $c = 2^{n-1}$, $i = n$ and $O(ki \cdot c^m)$, then the complexity after substitution of c and i is $O(kn \cdot 2^{(n-1)m})$). The genuine advantage of the constrained variant can be found when there are higher values of n and m. Thus, based on our beliefs about the likely operation of the retail energy market some "typical" values might be to have 24 items (e.g. 24 hours) and around 20 bidders (e.g. 20 UCs trying to sell their energy). Therefore, if we set $k = 1$ and restrict the number of possible correlations to 10, each one with 5 items (which experience indicates will provide UCs with enough *persuasive* power), the results are clear: mDJ presents a complexity of $1,498E + 147$, our mPJ $1,429E + 141$ and the constrained variant $5E + 20$. In our opinion, this means the constrained variant is sufficiently close to the optimal to be useful, but is still sufficiently tractable to be practicable.

5. Related Work

There has been comparatively little previous work in combinatorial energy markets, but there is a much larger literature on clearing algorithms for combinatorial auctions. However, these two strands of work have not been brought together before. The work of Ygge [8] is seminal in the area of agents and energy management. Specifically, he combines power load management with market-oriented programming. He introduces a hierarchical structure of *HomeBots*, intelligent agents that represent every load in the system and buy the energy in a system of forward non-combinatorial auctions. With only one energy supplier, his approach places all the initiative on the *HomeBots* so the UCs cannot express their preferences for having more or less demand at a certain time. We address this shortcoming by allowing combinatorial bidding.

Recently, there has been an enormous amount of research in combinatorial auctions [6], but most of this has focused on atomic propositions that may limit the choice (and hence the profit) to the auctioneer. Addressing this limitation, several authors have developed algorithms that deal with demand/supply bidding [7]. Moreover, [1] developed a single-item and a multi-item algorithm for multi-unit combinatorial reverse auctions with demand/supply functions that run in polynomial time (but that are not guaranteed to find the optimal solution). In [2] the same authors present another two algorithms for the same environment but they are optimal. The strategy they use consists in defining a dominant set containing an increasingly sorted group of single allocations and searching within this dominant set for the combinations that form the most profitable day allocation. The complexity in a worst case scenario is $O(n \cdot (k+1)^n)$ in the single-item case and $O(mn \cdot (k+1)^{mn})$ in the multi-item (where n, m, and k have the same meaning as in the previous section). In comparison to this work, sPJ is less general than sDJ (because it only clears continuous piece-wise supply functions), but both our algorithms present significantly lower computational complexity even in a worst-case scenario ($O(km)$ in the single-item case and $O(kmn \cdot 2^{(n-1)\cdot m})$ in the multi-item). That is even if, for instance $k = 1$, our mPJ algorithm is still 2^{n-1} times less complex than mDJ.

6. Conclusions and Future Work

The deregulation of the electricity industry offers new opportunities for providers and consumers. In this environment, customers can choose their suppliers to get cheaper energy and suppliers can compete to increase the number of their customers and, subsequently, their profits. To make this happen in practice, however, efficient electricity markets need to be developed. To this end, traditionally, energy management techniques have presented the two different sides with their own purposes and measures. On one hand, suppliers and retailers aim to smooth the overall energy consumption to avoid sudden peak loads. On the other hand, customers intend to reduce their energy bills without giving up freedom (meaning they can use energy at any time). Our system addresses both needs. It helps to reduce peak loads and to distribute them amongst less-loaded time slots. Specifically, by including off-peak hours in the discounts, UCs reward customers that consume electricity off-peak. Thus, they have an additional tool for energy management besides setting off-peak prices lower than peak ones. Moreover, the use of combinatorial auctions helps to produce efficient allocations of goods because combinatorial bidding allows the expression of more complex synergies between auctioned items [4]. Together with the use of supply functions and non-atomic propositions, consumers are able to accept energy from diverse UCs simultaneously, which, in turn, helps them to maximise their benefits.

Against this background, this paper presents, for the first time, an electricity retail market as a system of simultaneous reverse combinatorial auctions with supply-function bidding. Furthermore, we have developed the novel single and multi-item clearing algorithms sPJ and mPJ that are optimal, as well as a strategy to keep the multi-item algorithm within tractable ranges for the real-world problem we face. Future work will focus on evaluating the whole electricity market system and on reducing the complexity of the multi-item clearing algorithm with additional restrictions on combinatorial bidding. Further, we will focus on failure-scenarios and how to keep the demand in secure ranges to avoid blackouts or massive overbooking of the system. Finally, the likely pricing strategies of the suppliers need a more detailed study to determine how to maximise their revenue.

Acknowledgments

MThe authors would like to thank Viet Dang, Raj Dash and Alex Rogers from the IAM Group at the University of Southampton for their support and assistance.

References

[1] V. D. Dang and N. R. Jennings. Polynomial algorithms for clearing multi-unit single item and multi-unit combinatorial reverse auctions. In *Proceedings of ECAI '02 (Lyon France)*, pages 23–27, Lyon France, 2002.

[2] V. D. Dang and N. R. Jennings. Optimal clearing algorithms for multi-unit single item and multi-unit combinatorial auctions with demand/suppy function bidding. In *Proceedings of ICEC'03*, pages 25–30, Pittsburgh PA, 2003.

[3] R. K. Dash, D. C. Parkes, and N. R. Jennings. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):20–47, 2003.

[4] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the IJCAI'99*, pages 548–553, Stockholm Sweden, 1999.

[5] T. Groves. Incentives in teams. *Econometrica*, 41:617–31, 1973.

[6] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, 2002.

[7] T. Sandholm and S. Suri. Market clearability. In *Proceedings of the IJCAI'01*, pages 1145–1151, Seattle WA, 2001.

[8] F. Ygge. *Market-oriented programming and its application to power load management*. PhD thesis, Department of Computer Science, Lund University, 1998.