An Approach to Sustainable Electric Power Allocation Using a Multi-Round Multi-Unit Combinatorial Auction

著者（英）	Naoki Fukuta, Takayuki Ito
journal or publication title	Lecture Notes in Computer Science
volume	7580
page range	48-63
year	2012
URL	http://id.nii.ac.jp/1476/00003635/

doi: 10.1007/978-3-642-35612-4_4(http://dx.doi.org/10.1007/978-3-642-35612-4_4)

An Approach to Sustainable Electric Power Allocation Using a Multi-Round Multi-Unit Combinatorial Auction

Naoki Fukuta[1] and Takayuki Ito[2,3]

[1] Shizuoka University,Hamamatsu Shizuoka 4328011, JAPAN,
fukuta@cs.inf.shizuoka.ac.jp,
http://whitebear.cs.inf.shizuoka.ac.jp/
[2] Nagoya Institute of Technology, Gokiso-cho Nagoya 4668555, JAPAN
[3] Tokyo University, Bunkyo-ku Tokyo 113-0033, JAPAN

Abstract. In this paper, we present a preliminary idea about applying multi-unit combinatorial auctions to an electric power allocation problem when it includes sustainable power sources and it considers guaranteeing stable continuous use of the supplied power. Multi-unit combinatorial auction is a combinatorial auction that has some items that can bee seen as indistinguishable. Theoretically, such mechanisms could be applied for dynamic electricity auctions. We try to illustrate how such a mechanism can be applied to the actual electric power allocation problem when we consider the situation that there are sustainable electric power sources and guaranteeing stable continuous use of them. An approximation mechanism has been applied for a large-scale auction problem to overcome its computational intractability.

1 Introduction

One of the main issues on using sustainable electric sources is to solve the mismatches of their production availabilities and consumption needs[13][1]. They are dynamically changing in every time, depending on the consumers' context, weather conditions, etc. Furthermore, some may want to use energy produced from sustainable ways rather than generated by ordinary ways. Of course it might be due to their ideological preferences but sometimes it would be even for economical reasons since, for example, having a badge that shows the certificate of using a certain percentage of renewable energy[4] will increase a chance to have their customers.

On the other hand, there are some investigations and innovations on effective and efficient resource allocations among many self-interested attendees. Combinatorial auctions [2], one of the most popular market mechanisms, have a huge effect on electronic markets and political strategies. For example, Sandholm et al. [18] proposed a market mechanism using their innovative combinatorial auction algorithms. Multi-unit combinatorial auction is expected to be used on

[4] e.g., see http://wwf.panda.org/how_you_can_help/live_green/renewable_energy/

many problems that includes quantitative or countable items[12]. Combinatorial auctions provide suitable mechanisms for efficient allocation of resources to self-interested attendees [2]. Therefore, many works have been done to utilize combinatorial auction mechanisms for efficient resource allocation. For example, the FCC tried to employ combinatorial auction mechanisms for assigning spectrums to companies [14]. Therefore, it is natural to consider applying an auction-based approach to an electricity power usage allocation problem in a situation that various sustainable electric power sources are widely used.

However, a naive auction-based approach will cause some serious problems (e.g., dramatic up and down of prices due to lack of a proper mechanism to stabilize)[17]. Furthermore, in such case, their bidding might be complicated (e.g., a large number of bids would be necessary to represent them) so that it is very difficult to apply it to a large-scale problem.

To overcome those issues, many approaches have been proposed. For example, Zurel et al. proposed an heuristic approach that combines approximation of LP and a local search algorithm[19]. Also we proposed a parallel greedy approach[6], a performance analysis of algorithms[7][3], and its enhancement[8]. Recently, Fukuta proposed a fast approximation mechanism that can be applied to a multi-unit combinatorial auction which has very large amount of bids so that it cannot be easily solved by existing Linear Problem(LP) solvers[5]. Also the mechanism provided a pricing mechanism that is similar to VCG(Vickery-Clarke-Groves) which increases incentives to tell the true valuations for bidders. However, there is little empirical analysis about how such fast approximation mechanism can be applied to the actual electric power allocation problem. Also, there is a need to stabilize its dramatic vibration of prices through the time.

In this paper, we present a preliminary idea about applying multi-unit combinatorial auctions to an electric power allocation problem when it includes sustainable power sources and it considers guaranteeing stable continuous use of the supplied power. Also we briefly analyze how a fast approximation mechanism for it can be applied to the problem.

2 Preliminary

2.1 Multi-unit Combinatorial Auctions

Combinatorial auction is an auction that allows bidders to place bids for a combination of items rather than a single item[2]. The winner determination problem on single unit combinatorial auctions is defined as follows [2]: The set of bidders is denoted by $N = 1, \ldots, n$, and the set of items by $M = \{m_1, \ldots, m_k\}$. $|M| = k$. Bundle S is a set of items: $S \subseteq M$. We denote by $v_i(S)$, bidder i's valuation of the combinatorial bid for bundle S. An allocation of the items is described by variables $x_i(S) \in \{0, 1\}$, where $x_i(S) = 1$ if and only if bidder i wins bundle S. An allocation, $x_i(S)$, is feasible if it allocates no item more than once, for all $j \in M$.

$$\forall j \in M \quad \sum_{i \in N} \sum_{S \ni j} x_i(S) \leq 1$$

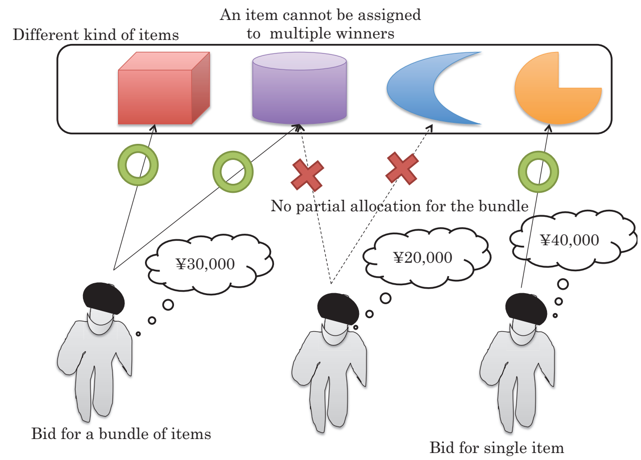

Fig. 1. An Example of (Single-Unit) Combinatorial Auction

The winner determination problem is the problem to maximize total revenue for feasible allocations $X \ni x_i(S)$.

$$\max_X \sum_{i \in N} \sum_{S \subseteq M} v_i(S) x_i(S)$$

Fig. 1 shows an example of single-unit combinatorial auction.

Note that we used simple *OR-bid* representation as the bidding language. Substitutability can be represented by a set of atomic *OR-bids* with dummy items[2].

When some items in auction can be replaceable each other, i.e., they are indistinguishable, the auction is called multi-unit auction. Multi-unit combinatorial auction is the case when some items are indistinguishable in a combinatorial auction[2].

Multi-unit combinatorial auction can be applied to electricity allocation problems, environmental exhausting right assignment problems, and other problems that considers quantitative or countable items in allocation problem[12].

Essentially, a multi-unit combinatorial auction problem equals to a single-unit combinatorial auction problem when the corresponding bidders treated some items as indistinguishable and placed substitutable bids for those items. Therefore, when we use *OR-bid* representation, a multi-unit combinatorial auction problem can be transformed into a simple single-unit combinatorial auction problem.

However, this expanding approach easily reaches an explosion of bids. For example, when we assume there are three types of item a, b, and c are auctioned and their stocks are 50, 100, and 200, respectively, a bid for bundle of item a, b and c can be expanded to $50 \cdot 100 \cdot 200 = 1,000,000$ of bids. This scale is far from tractable one. Therefore, when we need to have approximate solutions in such situation, it is demanded to realize an approximation algorithm that directly handles a multi-unit combinatorial auction problem.

2.2 Extending Lehmann's Approximation Approach

Lehmann's greedy algorithm [11] is a very simple but powerful linear algorithm for winner determination in combinatorial auctions. Here, a bidder declaring $< s, a >$, with $s \subseteq M$ and $a \in \mathcal{R}_+$ will be said to put out a bid $b =< s, a >$. The greedy algorithm can be described as follows. (1) The bids are sorted by some criterion. In [11], Lehmann et al. proposed sorting list L by descending average amount per item. More generally, they proposed sorting L by a criterion of the form $a/|s|^c$ for some number c, $c \geq 0$, possibly depending on the number of items, k. (2) A greedy algorithm generates an allocation. L is the sorted list in the first phase. Walk down the list L, allocates items to bids whose items are still unallocated.

The allocation algorithm can naturally be extended to multi-unit combinatorial auction problems. However, they did not mention about the applicability to multi-unit combinatorial auctions.

In [6],[7], and [8], we have shown that their hill-climbing approach outperforms SA[6], SAT-based algorithms[10], LP-based heuristic approximation approach[19], and a recent LP solver product in the setting when an auction has a massively large number of bids but the given time constraint is very hard. However, the algorithm is designed for single-unit combinatorial auction problems so it cannot be applied for multi-unit problems directly.

2.3 Winner Approximation and Pricing

It is crucial for a combinatorial auction mechanism to have proper pricing mechanism. In VCG(Vickery-Clarke-Groves) mechanism, prices that winners will pay will be given as follows[15]. A payment p_n for a winner n is calculated by

$$p_n = \alpha_n - \sum_{i \neq n, S \subseteq M} v_i(S) x_i(S)$$

Here, the right part of the right side of the equation denotes the sum of all bidding prices of won bids, excluding the bids that are placed by the bidder n. The left part of the right side of the equation, α_n is defined by

$$\alpha_n = max \sum_{i \neq n, S \subseteq M} v_i(S) x_i(S)$$

for a feasible allocation $X \ni x_i(S)$. This means that the α_n is the sum of all bidding prices of won bids when the allocation is determined as if a bidder n does not place any bids for the auction.

In [15], Nisan et al. showed that optimal allocations should be used for VCG-based pricing to make the auction incentive compatible (i.e., revealing true valuations is the best strategy for each bidders). Also, Lehmann et al. showed that VCG-based pricing with approximate winner determination will not make the auction incentive compatible even when it is assumed that all bidders are single-minded(i.e., each bidder can only place single bid at each auction)[11].

To overcome this issue, Lehmann et al. prepared a special pricing mechanism that can only be applied for their approximate greedy winner determination[11]. However, this pricing mechanism can only be applied to their allocation algorithm but it cannot be applied to other approximation allocation algorithms. Also the mechanism is incentive compatible only when single-minded bidders are assumed[11].

The main problem in which VCG-based pricing is applied to approximation allocation of items is that there are the cases that: (1) the price for a won bid is rather higher than the bid price, and (2) the price for a won bid is less than zero, it means the bidder will win the items and also will obtain some money rather than paying for it[15]. In the situation of (1), it breaks individual rationality (i.e., the one will not pay a higher price than the placed bid when the one won the bundle of items). Also the situation of (2) is not preferable for both auctioneers and sellers.

2.4 Approximation for Multi-unit Combinatorial Auctions

In this section, we briefly describe the approximation allocation algorithm for multi-unit combinatorial auctions proposed in [5], as follows.

The inputs are *Alloc*, *L*, and *Stocks*. *L* is the bid list of an auction. *Stocks* is the list of the number of auctioned units for each distinguishable item type. *Alloc* is the initial greedy allocation of items for the bid list.

```
function LocalSearch(Alloc, L, Stocks)
 RemainBids:= L - Alloc;
 sortByLehmannC(RemainBids);
 for each b ∈ RemainBids
  RestStocks:=getRestStocks({b}, Stocks);
  AllocFromWinners:=greedyAlloc(RestStocks, Alloc);
  RestStocks:=
     getRestStocks(AllocFromWinners + {b}, RestStocks);
  AllocFromRest:=
    greedyAlloc(RestStocks, RemainBids − {b});
  NewAlloc:=
    {b} + AllocFromWinners + AllocFromRest;
  if price(Alloc) < price(NewAlloc) then
    return LocalSearch(NewAlloc,L,Stocks);
 end for each
 return Alloc
```

The function *sortByLehmannC*(*Bids*) has an argument *Bids*. The function sorts the list of bids *Bids* by descending order of Lehmann's weighted bid price. The result are directly stored (overwritten) to the argument *Bids*. The function *getRestStocks*(*Bids*, *Stocks*) has two arguments : *Bids* and *Stocks*. The function returns how many unit of items will remain after allocating the items in *Stocks* to the list of bids *Bids*. The function *greedyAlloc*(*Stocks*, *Bids*) has two arguments : *Stocks* and *Bids*. The function allocates the items in *Stocks* to the list of bids *Bids* by using Lehmann's greedy allocation, and then the winner bids are returned as the return value. The function *price* calculates the sum of bidding prices for bids specified in the argument.

The optimality of allocations got by Lehmann's algorithm (and the following hill-climbing) deeply depends on which value was set to the bid sorting criterion c. Again, in [11], Lehmann et al. argued that $c = 1/2$ is the best parameter for approximation when the norm of the worst case performance is considered. However, the optimal values for each auction are varied from 0 to 1 even if the number of items is constant. Therefore, an enhancement has been proposed for this kind of local search algorithms by using parallel searches for multiple sorting criterion c[6]. Although the proposed enhancement is primarily designed for single-unit combinatorial auctions, this approach can be applied to the above mentioned approximation algorithm for multi-unit combinatorial auctions. In the algorithm, the value of c for Lehmann's algorithm is selected from a pre-defined list. It is reasonable to select c from neighbors of 1/2, namely, $C = \{0.0, 0.1, \ldots, 1.0\}$. The results are aggregated and the best one (i.e., that has the highest revenue) is selected as the final result.

To realize a pricing mechanism that receives little effect from the winners bid prices, we use the following algorithm. The inputs are *Alloc*, *L*, and *Stocks*. *L* is the bid list of an auction. *Stocks* is the list of the number of auctioned units for each distinguishable item type. *Alloc* is the initial allocation of items for the bid list that is obtained by the previously defined *LocalSearch* function.

```
function transformToSWPM(Alloc, L, Stocks)
 RemainBids:= L - Alloc;
 sortByLehmannC(RemainBids);
 clear(payment);
 for each b ∈ Alloc
  RestStocks:=getRestStocks(Alloc − {b}, Stocks);
  AllocForB:=greedyAlloc(RestStocks, RemainBids);
  NewAlloc:=Alloc-{b} + AllocForB;
  if price(Alloc) < price(NewAlloc) then
   return transformToSWPM(NewAlloc,L,Stocks);
  else payment_b = price(NewAlloc) − price(Alloc − {b})
end for each
return (Alloc,payment)
```

The above algorithm computes the price to be paid for each winner bid. The payment price for a winner bid b is denoted by $payment_b$, and it's value is obtained by $price(NewAlloc) - price(Alloc - \{b\})$. When the obtained payment price is higher than the bidding price of the winner bid, the algorithm discards the winner bid and place the items to *AllocForB*. To the end, the algorithm produces modified allocations *Alloc* and their payment prices *payment* that satisfies budget constraints for bidders.

For simplicity of description, the above algorithm is written with single-minded bidders assumption. To extend the algorithm without the assumption can be realized by just replacing $\{b\}$ with the all bids that come from the bidder of $\{b\}$.

3 Applying to Electric Power Allocation

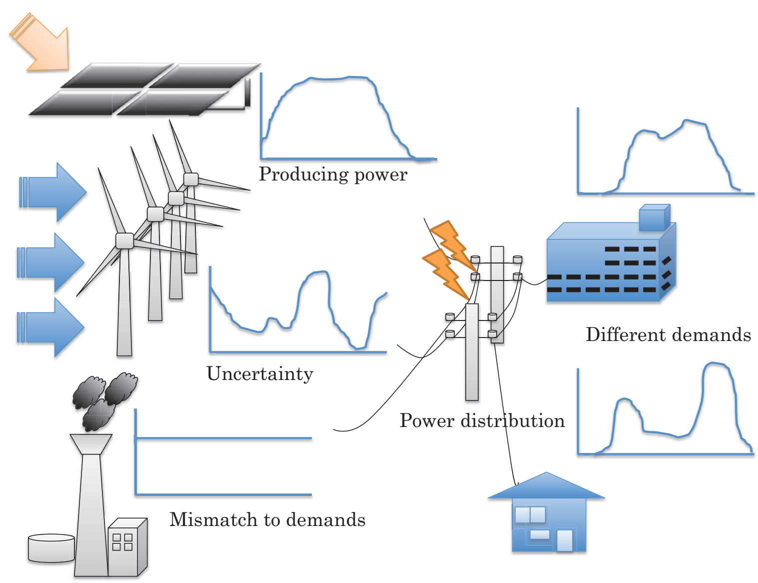

Fig. 2. Mismatches among Electricity Generation and Consumption Needs

In this section, we present how an electric power allocation problem can be transformed into *a sequence of* multi-unit combinatorial auctions[5].

3.1 The Allocation Model

First of all, we would start from a very simple case.

[5] A preliminary idea has been presented in [9]

Single power source, one time slot: Here we assume there is a single power source and allocate electric power consumption rights in a small time unit t. In this case, each bidder should place a bid that denotes the amount of necessary electric power and the possible highest price to be paid for it. For example, one can place a bid for the use of 50W in duration t with 0.04 Euro. This case is identical to a multi-unit (single-item) auction. When there are multiple preferences for different amounts of necessary electric power, e.g., 50W with 0.04 Euro but when it is for 40W the price becomes 0.02 Euro, such two bids can be placed at once, but each of them should include a dummy virtual item i_d in their bundle of items. Here, the dummy item i_d would be assigned to each bidder and which stock is always one. Therefore, the bidder only wins each of them at once but does not win both of them. This case is a simple case of a multi-unit combinatorial auction.

Single power source, multiple time slots: In some cases, we may need to use electricity continuously during a certain time period. For example, when operating a cloth washing machine, it takes a certain time period for its operation but the power supply should not be stopped during the operation time. In such case, we can include multiple time slots in an auction. A right to use electricity during multiple slots will be actually allocated when its first timeslot is reached to the current. Then, the occupied electric power will be removed from the auction in the next time. Otherwise the auction has the tentative winners but they will not be the final winners and it will continue the auction to accept further bids for unallocated electricity. This is a multi-unit combinatorial auction and also partly behaves as an ascending auction[6]. In this paper, we call it a *Multi-Round* auction approach.

Multiple power sources, single time slot: We can consider the difference of power sources in bidding. For example, let there are two power sources p_a and p_b at a time slot t. We can only place a bid for power source p_a but not for p_b when p_a is the preferable power source(e.g., a solar power generator) but p_b is not (e.g., a nuclear generator). Also we can place a bid for a mixture use of p_a and p_b, e.g., 500W from p_a and 50W from p_b, within a single combinatorial bid. This case is similar to a case in *single power source, multiple time slots* but it places for different power sources in a same time slot. Also one can place a set of combinations of such mixtures but at most only one mixture can be won, by using a dummy item which is described before.

Multiple power sources, multiple time slots: This is the most complicated case and it can be an extension of both *Single power source, multiple time slots* case and *Multiple power sources, single time slot* case. For example, one can place a bid for a period from t_1 to t_2 with the use of 500W from power source p_a and 20W from power source p_b. Also in such biddings, the actual power usage for each time slot can be varied rather than a simple combination of fixed values for each power sources.

Note that, in the above model, we do not consider how the obtained revenue should be distributed to the power suppliers. Also it is assumed that a power

[6] An analysis about this type of auction has been presented in [4]

supplier has the responsibility to produce the certain power at the specified time slot when it is allocated to consumers. When the power source cannot supply sufficient power in the time slot, the power source should buy the power from sufficient power sources via another auction. Therefore if there does not exist enough power supply from sufficient power sources in the market at the moment the power may not be supplied.

3.2 Example

Here we will give some examples for our approach. Since an important issue in using sustainable power sources is their uncontorollability and uncertainty of power generation levels(Fig.2), we will show how this issue can be captured in and handled by our proposed allocation model.

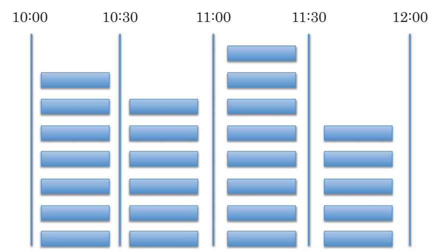

Fig. 3. Expected Power Generation Level in Each Time-Slot

Fig. 3 shows an example of the expected power generation levels at time slot from 10:00 to 12:00 in each 30 minutes. For example, in a time slot between 10:00 and 10:30, we have 7 units of electric power supply. Since the level may depend on its time, the supply level is 1 unit greater than the slot between 10:30 and 11:00.

Fig. 4 shows an example of winner determination result based on a combinatorial biddings to the expected power supplies. Here, we can place a bid for a set of each 3 units at the slots from 10:00 to 12:00 (i.e., the red case). Of course a bid for single time slot can be placed (e.g., the purple and the orange cases). Here, in this case, when the first time slot reaches the current time, these units are allocated to the winners for the slot (i.e., the purple and the red). Note that the red bid is for a bundle of multiple time slots, the units for other bundled time slots are also allocated to the red (Fig. 4:right).

Here, we consider what may happen when there is an update of expected power supply levels. Fig. 5 shows an example of such situation. Here, in Fig. 5(left), 1) there is a decrease of unit between 11:00 and 11:30, and 2) another bid (the blue one) appears. In this case, the green bid has no longer assigned to use any units but rather the blue and another purple one obtained new allocations, since the total revenue of the auction is increased by doing so. Then,

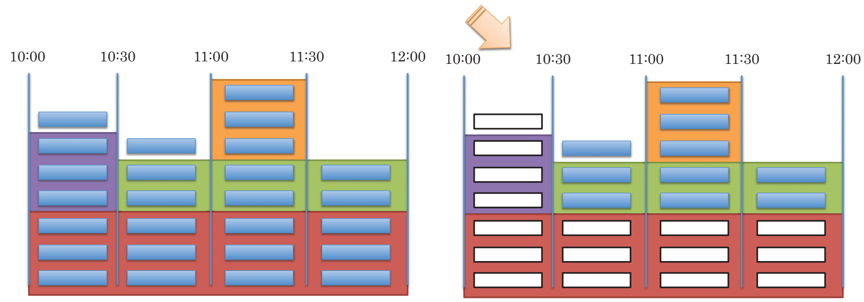

Fig. 4. Initial Bidding and Winner Determination

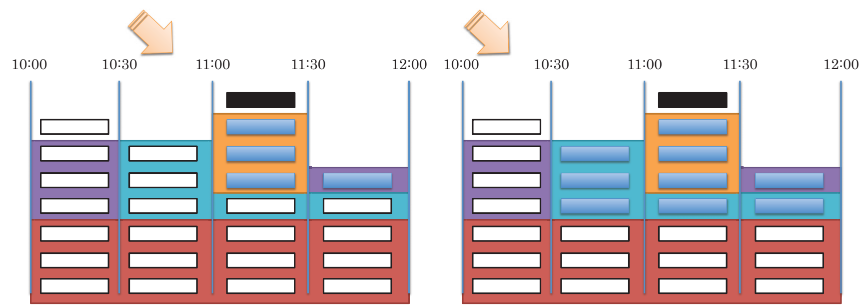

Fig. 5. Reallocation Due to Changes of Expected Supply Levels

after reaching the time of the next slot (i.e., from 10:30 to 11:00), the units are allocated to winners of the slot as we described before (i.e., Fig. 5:right).

In this way, the rights to use electricity power supplies are allocated to winners of the *current* slot and their bundled units in other slots, repeatedly.

4 Discussion

4.1 The Way of Performance Evaluation

Since there is no standard benchmark setting to evaluate electric power allocation performance that considers the consumers' preferences of power sources, we need to prepare a certain setting that reflects the important characteristics of the allocation problem that are discussed in this paper.

To our best knowledge, in [12], a similar problem setting has been introduced as a combinatorial auction problem. However, it does not consider such consumer's preferences of power sources since such problem setting was not realistic enough when [12] was published. Furthermore, there is little analysis about the change of our life when we have a rich amount of such sustainable electric energy sources. Therefore it is hard to prepare a set of realistic profiles that incorporates actual users' and producers' behaviors. As a first step, it is meaningful to use a general evaluation dataset that somewhat reflects the characteristics, e.g., multi-unit problem which large amount of bids, rather than preparing a pseudo simulation of such a world.

For general evaluation of winner determination performance on combinatorial auction, LeytonBrown et al. proposed CATS benchmark testsuite[12]. However, even if multi-unit auction is referred in [12], CATS suite does not include any data generation algorithm for multi-unit combinatorial auction. Therefore, we extended the existing auction problem generation algorithm to support multi-unit auctions by the following way.

Extending CATS standard dataset to multi-unit problems: Each auction problem generation algorithm in CATS generates artificial bids for a fixed size of items. The generated auction problems are single-unit combinatorial auction problems where each item in the auction has only one stock and these items are distinguished each other. When we consider each item has many stocks in a small size auction, the allocation problem could be rather much easier than that of single stocks since many conflicts (i.e., the situation that some bids placed to a set that includes an identical item) among bids can be automatically solved by allocating items to such conflicting bids. So, in the situation, many bids could win the items and only a limited number of bids might fail to win the items. However, when there are a huge number of bids in a single-unit combinatorial auction, the problem could be complex enough even when we assume there are a small number of stocks for each item in the auction. Here, as described in the previous section, we extend the dataset produced by CATS workbench by adding number of stocks for each non-dummy item in an auction. We call this 'a number of stocks for each item' approach.

This representation is also useful for representing items that can be shared with a limited number of people. For example, when we represent a fact that a radio frequency band can be shared by three devices at a time, the stocks for the item (i.e., the number of shared users for the bandwidth) is set to 3 in the auction. Also this representation does not have to generate a large number of bids even when the number of stocks is large. Another representation could be based on a representation of indistinguishable relationships among items but this representation inevitably generates a large number of definitions for such relationships. Therefore, we use 'the number of stock for each item' approach here.

The actual preparations of datasets have been done as follows. We used the bid distributions (i.e., the way to generate bids for items) that are defined and usable for generating the auction problems with a specified number of bids. Here we choose 20,000 bids for each auction so we choose the bid distribution L2, L3, L4, L6, L7, arbitrary, matching, paths, regions, and scheduling[7]. We prepared 100 auction problems for each bid distribution for both the size of 20,000 bids

[7] The reason why there are some missing number (e.g., L1, and L5,) is mainly the difficulty of generating the necessary number of bids by such bid distributions[12].

in an auction. We used those settings to make the results comparable to other papers[7]. The names for bid distributions are borrowed from [12][8]. Here, to keep the meaning of data generation algorithms, we choose fixed values for those stocks (e.g., every item has 4 stocks). We chose a fixed value, 16 for these stocks[9].

Note that, as mentioned before, in multi-unit auction problems, some bids that could be treated as dominated bids(e.g., having a higher price of bid for the same bundle of items) in single-unit auction problems could be winners of the auction. Therefore, we did not eliminate such bids to generate the original single-unit auction problems by CATS.

Compared algorithms: In [5], we have compared the following search algorithms: greedyL(C=0.5) uses Lehmann's greedy allocation algorithm[11] with parameter ($c = 0.5$). HC(c=0.5) uses a local search in which the initial allocation is Lehmann's allocation with $c = 0.5$ and conducts the hill-climbing search[6]. HC-3 uses the best results of the hill-climbing search with parameter ($0 \leq c \leq 1$ in 0.5 steps)[6],[7]. MHC(c=0.5) and MHC-3 are the proposed multi-unit enabled algorithms extended from HC(c=0.5) and MHC-3, respectively. greedyO means a simple greedy allocation of the received bids by the input order.

In this paper, due to the limited space of the paper, we only show the comparison to greedyO, greedyL(c=0.5), and our approach(MHC-3-para-100ms). Since the value in CPLEX was very low in the setting, we omitted it in the results. Further analysis can be found in [5].

Comparison criteria: Since it is really difficult to obtain the maximum revenue for an auction problem, we have compared algorithms with the values computed by average revenue ratio[7]. We use the same approach to evaluate performances of algorithms on single-unit auction problems.

Here, we use another approach that is based on the optimality ratio to the best one in the average on each bid distribution.

Let A be a set of algorithms, L be a dataset generated for this experiment, and $revenue_a(p)$ such that $a \in A$ be the revenue obtained by algorithm a for a problem p such that $p \in L$, the average revenue ratio $ratioM_a(L)$ for algorithm $a \in A$ for dataset L is defined as follows:

$$ratioM_a(L) = \frac{\sum_{p \in L} revenue_a(p)}{\max_{m \in A}(\sum_{p \in L} revenue_m(p))}$$

Here, we use $ratioM_a(L)$ for my comparison of algorithms on multi-unit auction problems. We also showed actual computation time for obtaining the approximation allocations.

Evaluating pricing performance: In addition to above-mentioned comparisons, we compared the performance of the proposed pricing mechanism. Since the pricing mechanism itself may modify allocations, we compared the algorithms in $ratioM$, and execution time to complete allocations and pricing.

[8] For more details about each bid distribution, see [12]

[9] Actually we conducted our experiments in four fixed values, 2, 4, 16, and 256 for the number of stocks. Due to limited space of the paper, we only presented the results for 16 stocks. Further detailed analysis can be found in [5]

Table 1. Detailed Pricing Performance on Multi-Unit Auctions (20,000bids-256items,with dominated bids,stocks=16)

	MHC-3-para-100ms		greedyL(c=0.5)		greedyO	
L2	1.0000	(157)	0.9994	(57)	0.6932	(6057)
L3	1.0000	(744)	0.9988	(764)	0.7064	(36503)
L4	1.0000	(414)	0.9705	(23761)	0.8664	(66774)
L6	1.0000	(292)	0.9497	(7207)	0.7380	(34904)
L7	1.0000	(475)	0.9771	(364)	0.7886	(1091)
arbitrary	1.0000	(13273)	0.9577	(4071)	0.8883	(6483)
matching	1.0000	(19633)	0.9996	(22137)	0.9718	(118207)
paths	1.0000	(95337)	0.9969	(84245)	0.9889	(49906)
regions	1.0000	(26031)	0.9731	(14288)	0.9461	(18883)
scheduling	1.0000	(140)	1.0000	(51)	0.9663	(58)
average	1.0000	(15650)	0.9823	(15695)	0.8554	(33886)

(each value in () is time in milliseconds)

Preliminary experiment environment: We implemented algorithms in a C program for the following experiments. The experiments were done with above implementations to examine the performance differences among algorithms. The programs were employed on a Mac with Mac OS X 10.4, a CoreDuo 2.0GHz CPU, and 2GBytes of memory.

4.2 Preliminary Analysis

Table 1 shows the performance *ratioM* of approximate winner determination when the proposed pricing mechanism is applied for each approximate allocation obtained by the shown approximate winner determination algorithms. Although the actual execution time for the pricing mechanism deeply depends on the number of winners in each auction problem, the average of total execution time on MHC-3-para-100ms is rather faster than that on greedyO, and also it is slightly faster than that on greedyL(c=0.5). Furthermore, the performance *ratioM* of MHC-3-para-100ms is higher than the others. This shows that the combination of MHC-3-para-100ms and the proposed pricing mechanism can work better than other combinations on the experiment setting. When we apply our algorithm to a sustainable electric power auction, we may obtain 17 percent of energy gain compared to a simple fast-in fast-allocate mechanism(greedyO) and even about 2 percent better than sort-and-allocate approach(greedyL) although its average computation time is slightly short. Note that this evaluation is only for starting discussions and more sophisticated way of evaluation should be considered in the future research.

4.3 Issues Left

As we described before, in the proposed auction model, we do not consider how the obtained revenue should be distributed to the power suppliers. When we consider this issue, the problem becomes to *combinatorial exchange*[16]. In this paper we do not provide any idea to solve this issue.

Also in this model, it is assumed that a power supplier has the responsibility to produce the certain power at the specified time slot when it is allocated to consumers. This may not be realistic enough, for example, when a big accident happens (e.g., a bid disaster removes a large amount of solar power stations). In such cases, it would be very hard to buy a sufficient amount of electric power from the market so the allocations for the consumers would be discarded.

Since the auction used in the model can be seen as a combination of one-shot auction and ascending auction, it is difficult to present a strict theoretical analysis for the auction. It would be more complex than that of ordinary online auctions(i.e., typically they allocate only one item for each round but in the proposed model many items can be allocated at once).

Also this does not reflect any regal issues and moral and ethical issues. The model may enforce people who don't have enough money to have little opportunity to obtain electricity when the total power supply level is low. Also the model may produce the situation that a plant may blackout even when it would produce serious environmental damages. The model does not consider such social costs that should be paid by the society itself.

5 Conclusions

In this paper, we discussed about a preliminary idea and an analysis about a dynamic electric power auction when there are sustainable power sources and consumers have their preferences to use. We illustrated how such an auction can be formalized as a variant of multi-unit combinatorial auctions when we only consider the allocation of aggregated electricity. We discussed about a possible performance based on a standard evaluation dataset which rather does not consider actual power use scenarios. Also we discussed the potential advantages and issues left in the presented analysis. Further analysis and implementations will be presented in future work.

Acknowledgements

The work was partly supported by Japan Cabinet Founding Program for NEXT Generation World-Leading Researchers(NEXT Program), and Grants-in-Aid for Young Scientists(B) 22700142.

References

1. Carrasco, J.M., Bialasiewicz, J.T., Guisado, R.C.P., Leon, J.I.: Power-electronic systems for the grid integration of renewable energy sources: A survey. IEEE Trans. Industrial Electronics 53(4), 1002–1016 (2006)
2. Cramton, P., Shoham, Y., Steinberg, R.: Combinatorial Auctions. The MIT Press (2006)

3. Fukuta, N., Ito, T.: Periodical resource allocation using approximated combinatorial auctions. In: Proc. of The 2007 WIC/IEEE/ACM International Conference on Intelligent Agent Technology(IAT2007). pp. 434–441 (2007)
4. Fukuta, N., Ito, T.: Approximated winner determination for a series of combinatorial auctions. In: Proc. of 1st International Conference on Agents and Artificial Intelligence(ICAART2009). pp. 400–407 (Jan 2009)
5. Fukuta, N.: Toward a vcg-like approximate mechanism for large-scale multi-unit combinatorial auctions. In: Proc. IEEE/ACM/WIC International Conference on Intelligent Agent Technology(IAT2011). pp. 317–322 (Aug 2011)
6. Fukuta, N., Ito, T.: Towards better approximation of winner determination for combinatorial auctions with large number of bids. In: Proc. of The 2006 WIC/IEEE/ACM International Conference on Intelligent Agent Technology(IAT2006). pp. 618–621 (2006)
7. Fukuta, N., Ito, T.: Fine-grained efficient resource allocation using approximated combinatorial auctions–a parallel greedy winner approximation for large-scale problems. Web Intelligence and Agent Systems: An International Journal 7(1), 43–63 (2009)
8. Fukuta, N., Ito, T.: An experimental analysis of biased parallel greedy approximation for combinatorial auctions. International Journal of Intelligent Information and Database Systems 4(5), 487–508 (2010)
9. Fukuta, N., Ito, T.: Toward combinatorial auction-based better electric power allocation on sustainable electric power systems. In: Proc. International Workshop on Sustainable Enterprise Software(SES2011). pp. 392–399 (Sep 2011)
10. Hoos, H.H., Boutilier, C.: Solving combinatorial auctions using stochastic local search. In: Proc. of the Proc. of 17th National Conference on Artificial Intelligence (AAAI2000). pp. 22–29 (2000)
11. Lehmann, D., O'Callaghan, L.I., Shoham, Y.: Truth revelation in rapid, approximately efficient combinatorial auctions. Journal of the ACM 49, 577–602 (2002)
12. Leyton-Brown, K., Pearson, M., Shoham, Y.: Towards a universal test suite for combinatorial auction algorithms. In: Proc. of ACM Conference on Electronic Commerce (EC2000). pp. 66–76 (2000)
13. MacKay, J.: Sustainable Energy - Without Hot Air. UIT Cambridge (2009)
14. McMillan, J.: Selling spectrum rights. The Journal of Economic Perspectives (1994)
15. Nisan, N., Ronen, A.: Computationally feasible VCG mechanisms. In: Proc. of ACM Conference on Electronic Commerce. pp. 242–252 (2000), citeseer.ist.psu.edu/nisan00computationally.html
16. Parkes, D.C., Cavallo, R., Elprin, N., Juda, A., Lahaie, S., Lubin, B., Michael, L., Shneidman, J., Sultan, H.: Ice: An iterative combinatorial exchange. In: The Proc. 6th ACM Conf. on Electronic Commerce (EC2005) (2005)
17. Ramchurn, S.D., Vytelingum, P., Rogers, A., Jennings, N.: Agent-based control for decentralized demand side management in the smart grid. In: Proc. 10th International Conference on Autonomous Agents and Multiagent Systems(AAMAS2011). pp. 5–12 (May 2011)
18. Sandholm, T., Suri, S., Gilpin, A., Levine, D.: Cabob: A fast optimal algorithm for winner determination in combinatorial auctions. Management Science 51(3), 374–390 (March 2005)
19. Zurel, E., Nisan, N.: An efficient approximate allocation algorithm for combinatorial auctions. In: Proc. of the Third ACM Conference on Electronic Commerce (EC2001). pp. 125–136 (2001)