

Bundle Design in Robust Combinatorial Auction Protocol against False-name Bids

Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara

NTT Communication Science Laboratories

2-4 Hikaridai, Seika-cho

Soraku-gun, Kyoto 619-0237 Japan

email: {yokoo, yuko, matsubara}@cslab.kecl.ntt.co.jp

url: <http://www.kecl.ntt.co.jp/csl/ccrg/members/{yokoo, yuko, matubara}>

Abstract

This paper presents a method for designing bundles in a combinatorial auction protocol that is robust against false-name bids. Internet auctions have become an integral part of Electronic Commerce and a promising field for applying AI technologies. However, the possibility of a new type of cheating called a false-name bid, i.e., a bid submitted under a fictitious name, has been pointed out.

A protocol called Leveled Division Set (LDS) protocol that is robust against false-name bids has been developed. However, this protocol requires the auctioneer to define a leveled division set. A leveled division set is a series of division sets, where a division set is a set of divisions and a division is a combination of bundles of goods. We need to solve a very complicated optimization problem to construct a leveled division set in order to obtain a good social surplus.

We have developed a heuristic method for overcoming this problem. In this method, we first find a good division with a winner determination algorithm, and then construct a leveled division set by using this division as a seed. Through a simulation, we show that our method can obtain a social surplus that is very close to optimal.

1 Introduction

Internet auctions have become an especially popular part of Electronic Commerce (EC). Various theoretical and practical studies on Internet auctions have already been conducted [Monderer and Tennenholtz, 1998; Wurman *et al.*, 1998]. Among these studies, those on combinatorial auctions have lately attracted considerable attention [Fujishima *et al.*, 1999; Leyton-Brown *et al.*, 2000; Klemperer, 1999; Sandholm, 1999; Lehmann *et al.*, 1999]. Although conventional auctions sell a single item at a time, combinatorial auctions sell multiple items with interdependent values simultaneously and allow the bidders to bid on any combination of items. In a combinatorial auction, a bidder can express complementary/substitutional preferences over multiple bids. By taking into account complementary/substitutional prefer-

ences, we can increase the participants' utilities and the revenue of the seller.

However, in [Sakurai *et al.*, 1999], the authors pointed out the possibility of a new type of cheating called *false-name bids*, i.e., an agent may try to profit from submitting false bids made under fictitious names, e.g., multiple e-mail addresses. Such a dishonest action is very difficult to detect since identifying each participant on the Internet is virtually impossible. Compared with collusion [Rasmusen, 1994; Klemperer, 1999], a false-name bid is easier to execute since it can be done by someone acting alone, while a bidder has to seek out and persuade other bidders to join in collusion.

The results reported in [Sakurai *et al.*, 1999; Yokoo *et al.*, 2000a] can be summarized as follows.

- The generalized Vickrey auction protocol (GVA) [Varian, 1995] is not robust against false-name bids in combinatorial auctions.
- There exists no combinatorial auction protocol that simultaneously satisfies incentive compatibility, Pareto efficiency, and individual rationality for all cases if agents can submit false-name bids.

In this paper, we concentrate on private value auctions [Mas-Colell *et al.*, 1995]. In private value auctions, each agent knows its own evaluation values of goods, which are independent of the other agents' evaluation values. We define an agent's utility as the difference between the true evaluation value of the allocated goods and the payment for the allocated goods. Such a utility is called a *quasi-linear* utility [Mas-Colell *et al.*, 1995]. These assumptions are commonly used for making theoretical analyses tractable.

In a traditional definition [Mas-Colell *et al.*, 1995], an auction protocol is (dominant strategy) incentive compatible, if bidding the true private values of goods is the dominant strategy for each agent, i.e., the optimal strategy regardless of the actions of other agents. The revelation principle states that in the design of an auction protocol we can restrict our attention to incentive compatible protocols without loss of generality [Mas-Colell *et al.*, 1995; Yokoo *et al.*, 2000a]. In other words, if a certain property (e.g., Pareto efficiency) can be achieved using some auction protocol in a dominant strategy equilibrium, i.e., the combination of dominant strategies of agents, the property can also be achieved using an incentive compatible auction protocol.

In this paper, we extend the traditional definition of incentive-compatibility so that it can address false-name bid manipulations, i.e., we define that an auction protocol is (dominant strategy) incentive compatible, if bidding the true private values of goods by using the true identifier is the dominant strategy for each agent. Also, we say that auction protocols are robust against false-name bids if each agent cannot obtain additional profit by submitting false-name bids. If such robustness is not satisfied, the auction protocol lacks incentive compatibility.

We say an auction protocol is Pareto efficient when the sum of all participants' utilities (including that of the auctioneer), i.e., the social surplus, is maximized in a dominant strategy equilibrium. In a more general setting, Pareto efficiency does not necessarily mean maximizing the social surplus. In an auction setting, however, agents can transfer money among themselves, and the utility of each agent is quasi-linear; thus the sum of the utilities is always maximized in a Pareto efficient allocation.

An auction protocol is individually rational if no participant suffers any loss in a dominant strategy equilibrium, i.e., the payment never exceeds the evaluation value of the obtained goods. In a private value auction, individual rationality is indispensable, i.e., no agent wants to participate in an auction where it might be charged more money than it is willing to pay.

The GVA protocol [Varian, 1995] is one instance of the well-known Clarke mechanism [Mas-Colell *et al.*, 1995]. It satisfies incentive compatibility, Pareto efficiency, and individual rationality in combinatorial auctions when there exists no false-name bid; however, this protocol is not robust against false-name bids [Sakurai *et al.*, 1999].

In [Yokoo *et al.*, 2000b], we developed a combinatorial auction protocol called the Leveled Division Set (LDS) protocol. This protocol satisfies incentive compatibility and individual rationality even if agents can submit false-name bids. This protocol does not satisfy Pareto efficiency, but it can achieve a better social surplus than that obtained by a trivial protocol that always sells goods in one bundle.

However, the LDS protocol requires the auctioneer to design a leveled division set, which is a series of division sets. A division set is a set of divisions and a division is a combination of bundles of goods. The auctioneer sells goods using these bundles. The obtained social surplus or the revenue of the auctioneer can vary significantly depending on the selection of the leveled division set. Although we presented the conditions that the leveled division set must satisfy to guarantee incentive compatibility [Yokoo *et al.*, 2000b], how to construct a leveled division set that could obtain a good social surplus is still an open question.

We need to solve a very complicated optimization problem in order to construct a leveled division set in such a way that a good social surplus can be obtained. This problem is much more complicated than winner determination problems [Fujishima *et al.*, 1999; Sandholm, 1999]. In a winner determination problem, given a set of bids, we try to find a combination of bids that maximizes the sum of the values of the bids. On the other hand, to construct a leveled division set, we need some model of the possible bids, e.g., a probabil-

ity distribution of the values of the bids. We try to maximize the expected social surplus based on such a model. Also, a leveled division set is a series of division sets, where each division is a combination of bundles. Finding an optimal solution for this problem becomes infeasible when the number of goods is large.

In this paper, we develop a heuristic method to construct a good leveled division set. This method uses a winner determination algorithm as a building block, i.e., this method first finds a good division using a winner determination algorithm, and then constructs a leveled division set using this division as a seed.

In the following, we first describe the LDS protocol. Then, we show the method for constructing a good leveled division set. Finally, we show evaluation results through a simulation.

2 LDS Protocol

In the following, we define several terms and notations. To help readability, we use three different types of parentheses to represent sets: $()$, $\{\}$, and $[\]$.

- A set of agents $N = \{1, 2, \dots, n\}$
- A set of all auctioned goods $M = (1, 2, \dots, m)$
- A division, which is a set of bundles, $D = \{S \subseteq M \mid S \cap S' = \emptyset \text{ for every } S, S' \in D\}$ ¹
- For each good j , the reservation price r_j is defined.
- For a bundle S , we define $R(S)$ as $\sum_{j \in S} r_j$.

A leveled division set is defined as follows:

- Levels are defined as $1, 2, \dots, \text{maxLevel}$.
- For each level i , a division set $SD_i = [D_{i1}, D_{i2}, \dots]$ is defined.

A leveled division set must satisfy the following three conditions.

- $SD_1 = [\{M\}]$ — the division set of level 1 contains only one division, which consists of a bundle of all goods.
- For each level and its division set, a union of multiple bundles in a division is always included in a division of a smaller level, i.e., $\forall i \geq 2, \forall D_{ik} \in SD_i, \forall D' \subseteq D_{ik}$, where $|D'| \geq 2, S_u = \bigcup_{S \in D'} S$, there exists a level $j < i$, with a division set SD_j , where $D_{jl} \in SD_j$ and $S_u \in D_{jl}$.
- For each level and its division set, each bundle in a division is not included in a division of a different level², i.e., $\forall i, \forall D_{ik} \in SD_i, \forall S \in D_{ik}, \forall j \neq i, \forall D_{jl} \in SD_j, S \notin D_{jl}$.

Figure 1 shows examples of leveled division sets. Case 1 shows one instance where there are two goods (A and B), and case 2 and case 3 show instances where there are three and four goods, respectively.

¹Note that we don't require that $\bigcup_{S \in D} S = M$ holds, i.e., satisfying $\bigcup_{S \in D} S \subseteq M$ is sufficient.

²This condition is not contradictory to the second condition, since the second condition involves the union of multiple bundles.

	case 1	case 2	case 3
level 1	[{(A,B)}]	[{(A,B,C)}]	[{(A,B,C,D)}]
level 2	[{(A),(B)}]	[{(A,B)}, {(B,C)}, {(A,C)}]	[{(A,B,C)}, {(B,C,D)}, {(A,D)}]
level 3		[{(A),(B),(C)}]	[{(A),(D),(B,C)}]

Figure 1: Example of Leveled Division Sets

To execute the LDS protocol, the auctioneer must pre-define the leveled division set and the reservation prices of goods. Each agent x declares its evaluation value $B(x, S)$ for each bundle S , which may or may not be true. We show a detailed description of the LDS protocol in the appendix. The outline of the protocol is as follows. The LDS protocol first tries to sell goods using the division in level 1, i.e., if there exists an evaluation value that is larger than $R(M)$, the goods are sold in one bundle. Otherwise, the protocol tries to sell goods using level 2 divisions, and so on. We say that the *applied level* of the LDS protocol is i if the goods are sold using a division in level i .

An intuitive explanation of why the LDS protocol satisfies incentive compatibility is as follows. Let us assume that agent x uses two false names x' and x'' to obtain bundles $S_{x'}$ and $S_{x''}$, respectively, at level i . Now, let us assume that agent x declares the evaluation value $R(S)$ for the bundle $S = S_{x'} \cup S_{x''}$ by using a single identifier. From the condition of a leveled division set, there exists a level $j < i$, where $S \in D_{jl}, D_{jl} \in SD_j$ holds. In this case, the condition in Step 1 of $LDS(j)$ is satisfied, i.e., only agent x declares evaluation values that are larger than or equal to the sum of reservation prices. Thus, agent x can obtain S by paying the minimal price $R(S)$. Therefore, by using a single identifier, the payment of agent x becomes smaller than (or equal to) the payment when agent x uses two false names.

3 Constructing Leveled Division Set

3.1 Basic Concepts

We assume that the auctioneer knows the probability distribution of the possible evaluation values of participants. The goal is to find a leveled division set and the reservation prices of goods so that the expected social surplus can be maximized.

For the number of goods m , the number of possible bundles of goods is 2^m . A division is a set of bundles and a leveled division set is a series of division sets. Clearly, enumerating all possible leveled division sets and finding the optimal one becomes infeasible when the number of goods is large. Even if we use a best-first search algorithm such as A^* , it is difficult to find an optimal leveled division without a very accurate function that estimates the expected social surplus for a partially defined leveled division set. Furthermore, we need to optimize the reservation prices of goods as well as the leveled division set.

In this paper, we make the following assumptions to attack this problem by a heuristic method.

- As in winner determination problems [Fujishima *et al.*, 1999; Sandholm, 1999], the number of bundles that participants actually need is relatively small compared with

the number of all possible bundles 2^m (sparseness of bids).

- The auctioneer knows the probability distributions of the highest evaluation values of these bundles³. Also, these probabilities are mutually independent.

With these assumptions, we can find a candidate for a division that can obtain a good social surplus using a winner determination algorithm [Fujishima *et al.*, 1999; Sandholm, 1999]. More specifically, instead of the actual value of a bid for each bundle, we use the expected value of the highest evaluation value for each bundle to solve the winner determination problem and obtain a division (we call this division *goal division*). If we can sell goods using this goal division, the obtained social surplus would be relatively good on average.

However, to sell goods using this goal division, we need to include additional divisions in earlier levels to satisfy the conditions of the leveled division set described in the previous section. Therefore, we put the goal division at level 3, and put additional divisions at level 2. By setting the reservation price of each good in an appropriate range, we can expect that goods will not be sold at level 1 or level 2, and we can sell goods using the goal division.

For example, let us assume that there are five goods A, B, C, D and E . Also, let us assume that by running a winner determination algorithm using the average value of each bundle, we obtain a goal division $\{(A), (B), (C), (D), (E)\}$. Then, we set level 3 of the leveled division set as $\{[(A), (B), (C), (D), (E))]\}$.

To satisfy the conditions of the leveled division set, we must put all unions of multiple bundles in this goal division at level 1 or level 2. There are four bundles in this goal division, so we must create unions of two bundles and three bundles and put them in the divisions at level 2. If two newly created bundles, each of which is a union of two bundles, do not have any goods in common, we can construct a division using these bundles. For example, we can put divisions $\{(A, B), (C, D, E)\}, \{(A, C), (B, D, E)\}$ and $\{(A, D, E), (B, C)\}$ in level 2. Then, all unions of two bundles appear in level 2. For a bundle that is a union of three bundles, we create a division that consists of only this bundle, and add them to level 2, e.g., we add $\{(A, B, C)\}, \{(A, B, D, E)\}, \{(A, C, D, E)\}$, and $\{(B, C, D, E)\}$.

Also, we can add a new goal division in level 3, as long as the division does not contain a bundle that appears in level 2. For example, we can add a division

³If we assume that the auctioneer has knowledge of the probability distributions of the evaluation values of agents, the probability distributions of the highest evaluation values can be calculated based on this knowledge. Since the LDS protocol is incentive compatible, we can assume that agents truthfully bid their true evaluation values.

$\{(A), (B), (C, D), (E)\}$, since it does not contain any bundles in level 2. To add this division in level 3, we must add divisions $\{(A, B, C, D)\}, \{(A, B, E)\}, \{(A, C, D), (B, E)\}$, and $\{(A, E), (B, C, D)\}$ in level 2.

It must be noted that even if the above assumptions do not hold, i.e., there might be unexpected bids or bids might be correlated, the LDS protocol using the leveled division set obtained by this method satisfies incentive compatibility (and the robustness against false-name bids). The violations of the above assumptions might degrade the obtained social surplus, but the protocol satisfies incentive compatibility as long as the required conditions of the leveled division set are satisfied.

3.2 Details of Method for Constructing Leveled Division Set

We show the details of the method for constructing a leveled division set. Let us represent the set of bundles that participants actually need as BS . For each bundle $S \in BS$, we assume that the auctioneer knows the probability distribution of the the highest evaluation value $B_{max}(S)$. Let us represent the expected value of $B_{max}(S)$ as $E(B_{max}(S))$. To help readability, for a division $D = \{S_1, S_2, \dots\}$, we represent a set of unions of k elements of D as $U(D, k)$.

We execute the following procedure. In the initial state, the level 1 division set SD_1 is $\{\{M\}\}$, and the level 2 and level 3 division sets SD_2 and SD_3 are empty sets \square .

Step 1: Find a goal division D that maximizes $\sum_{S \in D} E(B_{max}(S))$ using a winner determination algorithm⁴. Add D to SD_3 . We exclude a division that is already an element of SD_3 . Also, we exclude a division D where any element of $U(D, k)$ for all $k \geq 2$ is also an element of BS . This is because we need to put each element of $U(D, k)$ at level 2. Therefore, if any element is also a member of BS , there might be a relatively high evaluation value for this element and we would need to sell goods using the level 2 division. Also, we exclude a division D that contains a bundle that is already included in level 2. This is because we cannot put such D at level 3 by the conditions of the leveled division set.

Step 2: For all $D' = \{S'\}$, where $t = |D|$, $S' \in U(D, t-1)$, we add D' to SD_2 . Furthermore, for all $D'' = \{S_1, S_2\}$, where $S_1 \in U(D, u)$, $S_2 \in U(D, t-u)$, $u \geq 2$ and $S_1 \cap S_2 = \emptyset$, we add D'' to SD_2 .

Step 3: When adding another goal division, go to Step 1, otherwise, terminate the procedure.

The appropriate number of goal divisions in level 3 varies depending on the problem setting. In the problem settings used in the next section, we found that improvement of the obtained social surplus was saturated after adding about 20 goal divisions.

For solving a winner determination problem, we don't have to find the optimal solution. Since we use the expected value instead of the actual value, we cannot say that the optimal

⁴For simplicity, we assume $|D| \geq 3$ and $\bigcup_{S \in D} S = M$ hold. If D is a set of two bundles, we can put D into SD_2 . Also, if some goods are not included in D , we must add these goods to D .

solution for the winner determination problem always gives the best goal division. Therefore, using an approximate algorithm (such as [Sakurai *et al.*, 2000]) would be sufficient.

4 Evaluations

In this section, we show simulation results that demonstrate the quality of the obtained leveled division set using our heuristic method. We determine a bundle S that participants actually need and the probability distribution of the highest evaluation value $B_{max}(S)$ by the following method.

- For each bundle S , we determine the number of goods in S (represented as $|S|$) by using a binomial distribution $B(p, m)$, i.e., the probability that the number of goods is j is given by $p^j(1-p)^{m-j}m!/(j!(m-j)!)$ [Fujishima *et al.*, 1999]. Next, we randomly choose $|S|$ goods included in S . Also, we choose $E(B_{max}(S))$ randomly from within the range of $[(1-\epsilon)|S|, (1+\epsilon)|S|]$. Then, we set the distribution of $B_{max}(S_k)$ as a uniform distribution $[(1-q)E(B_{max}(S)), (1+q)E(B_{max}(S))]$, where $0 \leq q \leq 1$.

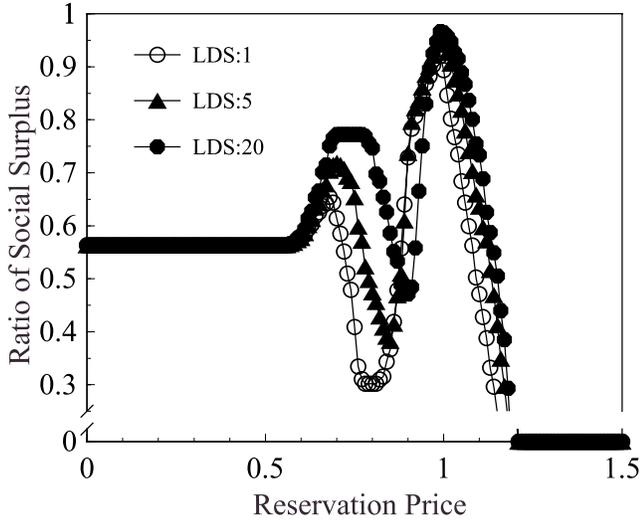
If q is very small, the auctioneer has very good knowledge of the highest evaluation value of S . If q is relatively large compared with ϵ , the auctioneer has little knowledge of the possible highest evaluation value of S , and the probability that the goal division can actually achieve a Pareto efficient social surplus becomes low.

Figure 2 (a) shows the average ratio of the obtained social surplus to the Pareto efficient social surplus by varying the reservation price⁵. In Figure 2, we set $m = 10$, the number of bundles $|BS| = 200$, $p = 0.2$, $\epsilon = 0.1$, and $q = 0.1$. Each data point is the average of 100 problem instances, in which the evaluation values of bundles are generated based on the probability distribution determined by the above method. In Figure 2 (a), we plot the results where we put only one goal division (LDS:1), 5 goal divisions (LDS:5), and 20 goal divisions (LDS:20). Also, Figure 2 (b) shows the trends of how the applied levels change according to the reservation price in LDS:20. More specifically, for each value of the reservation price, we show the cumulative frequency of three levels in which the goods are sold,

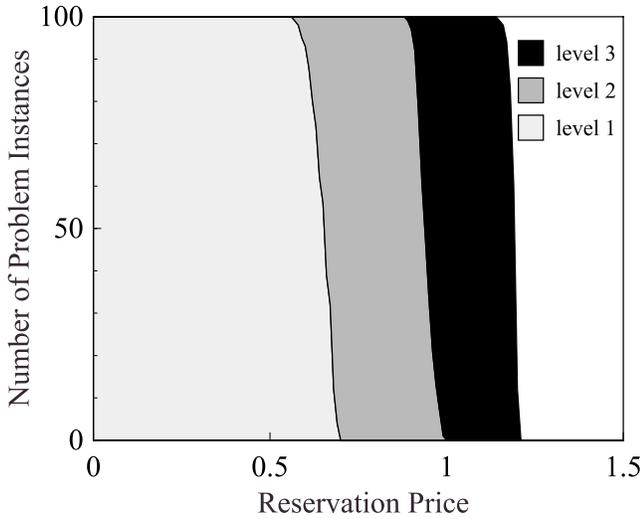
As shown in Figure 2 (b), when the reservation price is low, the applied level is 1, i.e., the LDS protocol sells goods using the level 1 division, which means that the protocol always sell goods in one bundle to a single agent. When the reservation price is around 0.7, the applied level becomes 2. Also, the social surplus drops when the reservation price is around 0.9. This is because only one bundle in a level 2 division (which consists of two bundles) is sold in some problem instances.

If we set the reservation price in an appropriate range (in this case, around 1.0), the applied level becomes 3 and the protocol can sell goods using the goal divisions; thus the obtained social surplus becomes very good (more than 90% of the optimal) even if we put only one division at level 3. By

⁵We use an identical reservation price for all goods since in current problem settings, the evaluation values of goods are basically similar. If the evaluation values of goods can be very different, we need to set different reservation prices for different goods.



(a) Obtained Social Surplus



(b) Applied Levels (LDS:20)

Figure 2: Evaluation Results ($m=10$, $q=0.1$)

putting 20 goal divisions at level 3, the obtained social surplus becomes very close to the optimal. We found the improvement of the obtained social surplus becomes saturated after adding 20 goal divisions.

In Figure 3, we show the results of LDS:20, where we set $q = 0.15, 0.2$, and 0.25 . As expected, when q becomes relatively large compared with $\epsilon = 0.1$, the probability that the goal division can actually obtain a Pareto efficient social surplus becomes very low and the obtained social surplus of the LDS protocol becomes low. However, even when $q = 0.2$, the LDS protocol can obtain a social surplus that is around 82% of the optimal.

Figure 4 shows the results obtained when the number of goods becomes large, i.e., we set $m = 200$, $|BS| = 1000$, $p = 0.2$, $\epsilon = 0.2$, and $q = 0.3$. Each data point is the average of 50 problem instances. We can see that even though

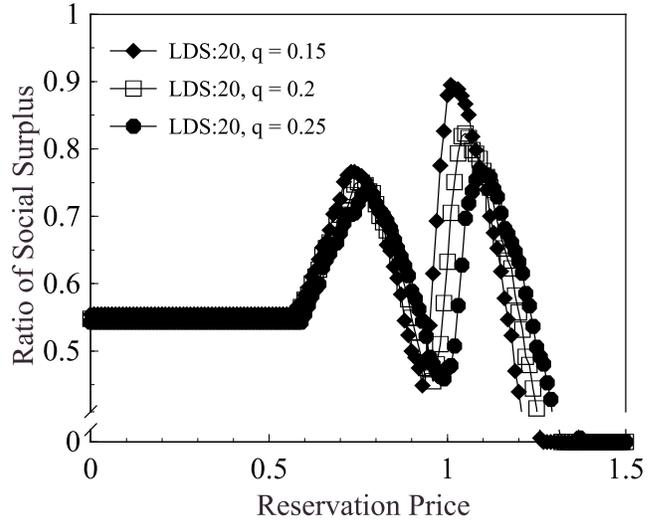


Figure 3: Obtained Social Surplus ($m=10$, $q=0.15, 0.2, 0.25$)

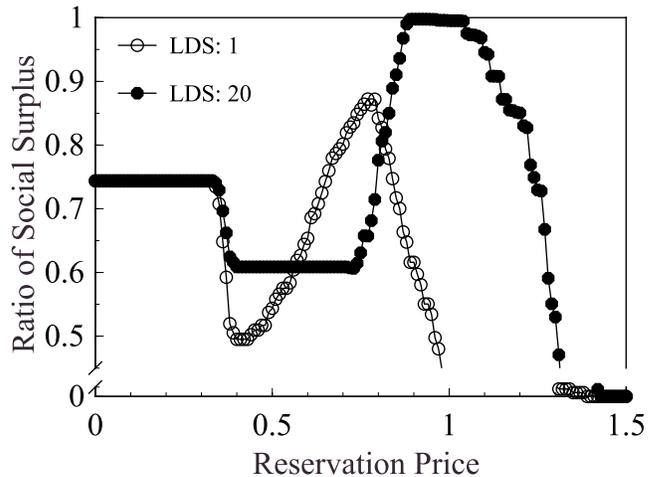


Figure 4: Obtained Social Surplus ($m=200$, $q=0.3$)

q is relatively large, the obtained social surplus is very close to the optimal. In this case, the sizes of bundles in the goal division tend to be large, e.g., a bundle of 40 goods. Therefore, the possibility that goods are sold by level 2 divisions, which contain very large bundles (e.g., 80 goods), becomes very small. Thus, we can easily set the reservation price so that the applied level becomes 3.

Figure 5 shows the results obtained when adding some noise to problem instances. More specifically, we add several unexpected bids, i.e., the bids submitted by agents whose evaluation values do not correspond to the model the auctioneer has. A noise bid is generated using basically the same method for generating original bids. In Figure 5, the x-axis shows the number of additional noise bids. The parameter settings are identical to those of Figure 4, and we show the results of LDS:20, where the reservation price is 1. We can

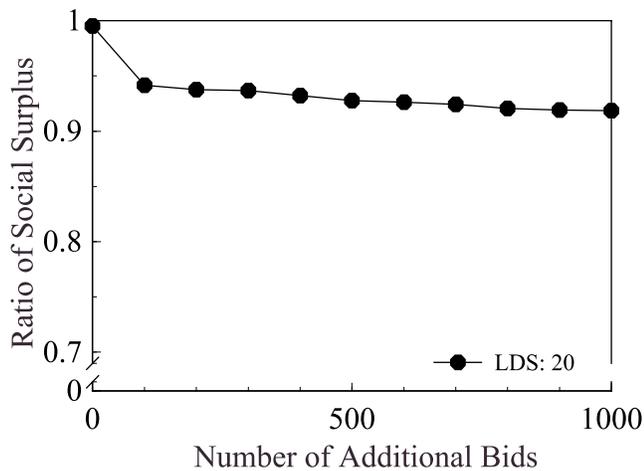


Figure 5: Effect of Adding Noise

see that our method is very robust against additional noise bids. Even if we add 1000 noise bids, which is equal to $|BS|$, the obtained social surplus is around 92% of the optimal.

5 Conclusions

In this paper, we developed a heuristic method for constructing a good division set that can be used in the LDS protocol. This method uses a winner determination algorithm to find a good division, and then constructs a leveled division set by using this division as a seed. Evaluation results showed that the LDS protocol using the leveled division set constructed by our method can obtain a social surplus that is very close to optimal when the auctioneer has reasonably good knowledge of the highest evaluation values of agents. Furthermore, the protocol can still obtain a good social surplus when the knowledge of the auctioneer is relatively poor. Our future works include evaluating this method in more realistic problem settings based on real application problems such as [Leyton-Brown *et al.*, 2000], and extending a double auction protocol that is robust against false-name bids [Yokoo *et al.*, 2001] so that it can handle multiple items.

References

- [Fujishima *et al.*, 1999] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computation complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 548–553, 1999.
- [Klemperer, 1999] Paul Klemperer. Auction theory: A guide to the literature. *Journal of Economic Surveys*, 13(3):227–286, 1999.
- [Lehmann *et al.*, 1999] Daniel Lehmann, Liadan Ita O’Callaghan, and Yoav Shoham. Truth revelation in approximately efficient combinatorial auction. In *Proceedings of the First ACM Conference on Electronic Commerce (EC-99)*, pages 96–102, 1999.
- [Leyton-Brown *et al.*, 2000] Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of the Second ACM Conference on Electronic Commerce (EC-00)*, pages 66–76, 2000.
- [Mas-Colell *et al.*, 1995] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
- [Monderer and Tennenholtz, 1998] Dov Monderer and Moshe Tennenholtz. Optimal auctions revisited. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 32–37, 1998.
- [Rasmusen, 1994] Eric Rasmusen. *Games and Information*. Blackwell, 1994.
- [Sakurai *et al.*, 1999] Yuko Sakurai, Makoto Yokoo, and Shigeo Matsubara. A limitation of the Generalized Vickrey Auction in Electronic Commerce: Robustness against false-name bids. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 86–92, 1999.
- [Sakurai *et al.*, 2000] Yuko Sakurai, Makoto Yokoo, and Koji Kamei. An efficient approximate algorithm for winner determination in combinatorial auctions. In *Proceedings of the Second ACM Conference on Electronic Commerce (EC-00)*, pages 30–37, 2000.
- [Sandholm, 1999] Tuomas Sandholm. An algorithm for optimal winner determination in combinatorial auction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 542–547, 1999.
- [Varian, 1995] Hal R. Varian. Economic mechanism design for computerized agents. In *Proceedings of the First Usenix Workshop on Electronic Commerce*, 1995.
- [Wurman *et al.*, 1998] Peter R. Wurman, Michael P. Wellman, and William E. Walsh. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In *Proceedings of the Second International Conference on Autonomous Agents (Agents-98)*, pages 301–308, 1998.
- [Yokoo *et al.*, 2000a] Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. The effect of false-name declarations in mechanism design: Towards collective decision making on the Internet. In *Proceedings of the Twentieth International Conference on Distributed Computing Systems (ICDCS-2000)*, pages 146–153, 2000.
- [Yokoo *et al.*, 2000b] Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. Robust combinatorial auction protocol against false-name bids. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 110–115, 2000.
- [Yokoo *et al.*, 2001] Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. Robust double auction protocol against false-name bids. In *Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS-2001)*, 2001. (to appear).

Appendix: Details of LDS Protocol

For a division $D = \{S_1, S_2, \dots\}$ and one possible allocation of goods G , we say G is allowed under D if each bundle in D is allocated to different agents in G . Also, we allow that some bundle is not allocated to any agent. In that case, we assume that the bundle is allocated to a dummy agent d , whose evaluation value of the bundle S is equal to $R(S)$. For each level i and its division set $SD_i = [D_{i1}, D_{i2}, \dots]$, we represent a union of all allowed allocations for each element of SD_i as SG_i . The declared evaluation value of agent x for an allocation G (represented as $v_x(G)$) is defined as $B(x, S)$ if S is allocated to agent x in G , otherwise $v_x(G) = 0$. Also, we define the evaluation value of a dummy agent d for an allocation G as the sum of the reservation prices of goods that are not allocated to real agents in G .

The winners and payments are determined by calling the procedure $\text{LDS}(1)$. $\text{LDS}(i)$ is a recursive procedure defined as follows. Note that the procedures in $\text{GVA}(i)$ are equivalent to those in the GVA [Varian, 1995], except that the possible allocations are restricted to SG_i .

Procedure $\text{LDS}(i)$

Step 1: If there exists only one agent $x \in N$ whose evaluation values satisfy the following condition: $\exists D_{ik} \in SD_i, \exists S_x \in D_{ik}$, where $B(x, S_x) \geq R(S_x)$, then compare the results obtained by the procedure $\text{GVA}(i)$ and $\text{LDS}(i+1)$, and choose the one that gives the larger utility for agent x . If the condition of Step 1 is also satisfied for $\text{LDS}(i+1)$, then also compare with the results of $\text{GVA}(i+1)$ and $\text{LDS}(i+2)$, and so on. When choosing the result of $\text{LDS}(i+1)$, we don't assign any good, nor transfer money, to agents other than x , although the assigned goods for agent x and its payment are calculated as if goods were allocated to the other agents.

Step 2: If there exist at least two agents $x_1, x_2 \in N, x_1 \neq x_2$ whose evaluation values satisfy the following condition: $\exists D_{ik} \in SD_i, \exists D_{il} \in SD_i, \exists S_{x_1} \in D_{ik}, \exists S_{x_2} \in D_{il}$, where $B(x_1, S_{x_1}) \geq R(S_{x_1}), B(x_2, S_{x_2}) \geq R(S_{x_2})$, then apply the procedure $\text{GVA}(i)$.

Step 3: Otherwise: call $\text{LDS}(i+1)$, or terminate if $i = \text{max_level}$.

Procedure $\text{GVA}(i)$: Choose an allocation $G^* \in SG_i$ such that it maximizes $\sum_{y \in N \cup \{d\}} v_y(G)$. The payment of agent x (represented as p_x) is calculated as $\sum_{y \neq x} v_y(G_{\sim x}^*) - \sum_{y \neq x} v_y(G^*)$, where $G_{\sim x}^* \in SG_i$ is the allocation that maximizes the sum of all agents' (including the dummy agent d) evaluation values except that of agent x .