アルゴリズム 88−7
(2003. 1. 20)

組合せオークションの勝者決定問題に対する発見的解法

水戸将弥 † 藤田 聡 †

† 広島大学大学院 工学研究科 情報工学専攻
〒 739-8527 広島県東広島市鏡山一丁目 4-1
{mito,fujita}@se.hiroshima-u.ac.jp

あらまし: 組合せオークションの勝者決定問題は与えられた入札集合 $\mathcal{B}$ から入札値の和が最大となる割当可能な部分集合 $\mathcal{B}'$ を見つける問題で最大重み集合詰め込み問題と等価である。また勝者決定問題は多項式時間での近似が困難であることが知られており、発見的探索では入札の適切な順序付けが高速に良い解を求めるために重要となる。本稿では入札を順序付けるための 3 つの発見的手法を提案する; 最初の 2 つの手法は競合する 2 つの入札が共有する財の個数を利用するもので、3 番目の手法はこれらの局所的な手法を再帰的に利用するものである。提案手法の性能は実験的に評価され、その結果、最初の 2 つの手法を利用することで発見的探索の anytime 性能が改善された。

On Heuristics for Solving Winner Determination Problem in Combinatorial Auctions

Masaya Mito† Satoshi Fujita†

† Graduate School of Engineering, Hiroshima University
Kagamiyama 1 chome 4-1, Higashi-Hiroshima, 739-8527 Japan

Abstract: The winner determiniation problem (WDP) in combinatorial auctions is the problem of, given a finite set of combinatorial bids $\mathcal{B}$, finding a feasible subset $\mathcal{B}'$ of $\mathcal{B}$ with a maximum revenue. WDP is known to be equivalent to the maximum weight set packing problem, and hard to approximate by polynomial time algorithms. This paper proposes three heuristic bid ordering schemes for solving WDP; the first two schemes take into account the number of goods shared by conflicting bids, and the third one is based on a recursive application of such local heuristic functions. We conducted several experiments to evaluate the goodness of the proposed schemes. The result of experiments implies that the first two schemes are particularly effective to improve the anytime performance of the resulting heuristic search procedures.

1 Introduction

Let $S = \{x_1, x_2, \ldots, x_m\}$ be a set of **goods** sold by the auctioneer. In combinatorial auctions, buyers submit a set of bids to the auctioneer, where a bidding is made on a subset of goods instead of a single good as in classical auctions, and the auctioneer selects a subset of those bids in such a way to maximize the revenue of the auctioneer. A bidder of a selected bid is called a "winner" of the auction. In this paper, we assume that each bidder can submit any number of bids, and can be a winner of several bids, without loss of generality[1]. Note that this assumption enables us to separate bids from bidders.

Let $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$ be a set of bids submitted by the bidders. Each bid $B_i \in \mathcal{B}$ is an ordered pair $\langle S_i, v_i \rangle$, where S_i is a nonempty subset of S called **bidset** (or simply "bid") and v_i is an integer referred to as the **bid value** (or simply "value"). A subset $\mathcal{B}'$ of $\mathcal{B}$ is said to be **feasible** if no two bids contained in the subset have a nonempty intersection. In addition, a bid $B_i (\in \mathcal{B})$ is said to be feasible with respect to $\mathcal{B}' (\subseteq \mathcal{B})$ if set $\mathcal{B}' \cup \{B_i\}$ is feasible. The **revenue** of subset $\mathcal{B}' (\subseteq \mathcal{B})$, denoted by $r(\mathcal{B}')$, is the sum of bid values contained in $\mathcal{B}'$. The winner determiniation problem (WDP, for short) is the problem of, given a finite set of bids $\mathcal{B}$, finding a feasible subset $\mathcal{B}'$ of $\mathcal{B}$ with a maximum revenue.

In the literature, it is known that WDP is equivalent to the maximum weight set packing problem

[1] It is known that this assumption could violate an efficient assignment of "wins" to the bidders especially when there are *substitutional* goods in S. Such a problem could be resolved by introducing dummy goods shared by substitutional bids so that any feasible selection selects those bids exclusively [3].

[2], and can be reduced to the maximum weight independent set problem (MWIS) that has been investigated extensively during the past three decades. A trivial reduction from the maximum independent set problem (MIS) [8] implies that WDP is a problem that is hard to approximate [13]; i.e., there is no polynomial time algorithm for solving WDP to have an approximation ratio $O(n^{1-\varepsilon})$ for any $\varepsilon > 0$, unless NP=ZPP [5]. On the other hand, WDP can be solved in polynomial time by restricting the class of allowable bids [8]: e.g., if $|S_i| \leq 2$ for all $\langle S_i, v_i \rangle \in \mathcal{B}$, a given instance of WDP can be solved in polynomial time by using a trivial reduction to the maximum weight matching problem (note that we can not generalize this approach to bids with larger sizes, since if the size of every bid is bounded by three, it includes the 3-dimensional matching problem as a special case). In addition, an instance of WDP could be solved in polynomial time if MWIS for the corresponding bid graph is solvable in polynomial time, as in the case of interval graphs [11] and permutation graphs [1] (a formal definition of bid graphs will be given in the next section).

In this paper, we will adopt the *heuristic search method* as a tool for obtaining a sufficiently good solution with a sufficiently short computation time. As will be discussed in Section 3, a key issue in designing efficient heuristic search procedures is how to determine the priority of bids that are sequentially examined in the given search procedures. In this paper, we propose three heuristic bid ordering schemes for solving WDP; the first two schemes take into account the number of goods shared by conflicting bids, that is generally omitted in conventional heuristic schemes, and the third one is based on a recursive application of such local heuristic functions. We conducted several experiments to evaluate the goodness of the proposed schemes, and the result of experiments implies that the first two schemes are particularly effective to improve the *anytime performance* of the resulting search procedures.

The remainder of this paper is organized as follows. Section 2 introduces necessary definitions. Related works on the heuristic search are reviewed in Section 3. Section 4 describes proposed bid ordering schemes. The goodness of the schemes is experimentally evaluated in Section 5. Finally, Section 6 concludes the paper.

2 Preliminaries

A **bid graph** G with respect to a set of bids $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$ is a 4-tuple (V, E, w_v, w_e) defined as follows:

- the set of vertices V in G coincides with $\mathcal{B}$,
- for any two vertices B_i, B_j in V, they are connected by an edge in E iff $S_i \cap S_j \neq \emptyset$,
- each vertex B_i in G is weighted by its value v_i, i.e., $w_v(B_i) \stackrel{\text{def}}{=} v_i$ for each B_i, and
- each edge $\{B_i, B_j\}$ in G is weighted by $|S_i \cap S_j|$, i.e., $w_e(B_i, B_j) \stackrel{\text{def}}{=} |S_i \cap S_j|$.

It is easy to see that a maximum weight independent set of a bid graph G with respect to $\mathcal{B}$ corresponds to a feasible subset $\mathcal{B}'$ of $\mathcal{B}$ with a maximum revenue. This reduction implies that any heuristics for solving MWIS can be used for solving WDP in a heuristic manner.

Let $N_G(v)$ denote the set of vertices adjacent to v in graph G. The cardinality of $N_G(v)$, denoted by $d_G(v)$, is referred to as the degree of vertex v in G. Note that for given bid graph G and a vertex v in it, $d_G(v)$ represents the number of bids intersecting with a given bid corresponding to vertex v. Let Δ_G denote the maximum vertex degree in graph G; i.e., $\Delta_G \stackrel{\text{def}}{=} \max_{v \in V(G)} d_G(b)$.

Let $r_A(\mathcal{B})$ denote the revenue obtained by an algorithm A for given instance $\mathcal{B}$. The **approximation ratio** ρ_A of an algorithm A is defined as

$$\rho_A \stackrel{\text{def}}{=} \inf_{\mathcal{B} \in \mathcal{U}} \frac{r_A(\mathcal{B})}{r_{Opt}(\mathcal{B})}$$

where $\mathcal{U}$ is the universe of $\mathcal{B}$, and *Opt* denotes an optimal algorithm.

3 Heuristic Search

Heuristic search is a common technique used in many heuristic algorithms for solving WDP [3, 6, 10, 13, 11, 12]. Those algorithms generally based on several basic factors, that could be summarized as follows: 1) how to determine the priority of bids, 2) how to quickly identify a feasible bid with respect to a given partial solution, and 3) how to avoid unnecessary trials that will not lead to an optimal solution. Note that the first one is an important factor even in designing efficient greedy algorithms, and the third one is closely related with the branch-and-bound method. In this paper, we are particularly interested in the first factor, and a deep investigation of the last two factors will be left as a future research.

3.1 Bid Ordering

A (nearly) optimal bid ordering could generate a (nearly) optimal solution to WDP very quickly even under heuristic search schemes. In determining an appropriate ordering of bids in $\mathcal{B}$, we can use the following (local) informations concerned with the given instance: A) the value of each bid; B) the size of each bidset; and C) the way of conflicts with the other bids in $\mathcal{B}$. Since WDP is the problem of maximizing the revenue of the auctioneer, the first factor must be taken into account in any heuristic scheme. On the other hand, as the way of considering the other two factors, several interesting techniques have been proposed in the literature.

B) Given bid $B_i = \langle S_i, v_i \rangle \in \mathcal{B}$, a **normalized bid value** of B_i, denoted by $\tilde{v}_i$, is defined as

$$\tilde{v}_i \stackrel{\text{def}}{=} \frac{v_i}{|S_i|^\alpha}$$

where α is a configurable parameter. A simplest heuristic for solving WDP is to select bids in $\mathcal{B}$ in a non-increasing order of $\tilde{v}_i$ in a greedy manner [6], as is commonly done in several algorithms for solving knapsack-like optimization problems. In what follows, we refer to this greedy scheme as the normalized bid value greedy method (NVG, for short). It is known that the approximation ratio of NVG (without backtracking) is at most $1/\sqrt{m}$ provided that $\alpha = 0.5$ [6].

The performance of NVG could be improved by using the technique of heuristic estimation; i.e., to modify the scheme in such a way that a bid that is likely to cause a higher revenue will be selected in a greedy manner [10, 13]. Given feasible set of bids $\mathcal{B}'(\subseteq \mathcal{B})$, let us define an estimated revenue with respect to $\mathcal{B}'$ as follows:

$$h(\mathcal{B}') \stackrel{\text{def}}{=} \sum_{x \in S'} \left\{ \max_{S_j \ni x, S_j \cap (S - S') = \emptyset} \tilde{v}_j \right\} \quad (1)$$

where S' is the set of goods that are not contained in $\mathcal{B}'$. The normalized bid value with estimation (NVE, for short) is a heuristic scheme that selects a feasible bid $B_i = \langle S_i, v_i \rangle$ that maximizes $r(\mathcal{B}') + v_i + h(\mathcal{B}' \cup \{B_i\})$ in a greedy manner. NVE is used in several heuristic schemes as a basic component [10, 13].

C) An alternative approach for solving WDP is to solve the corresponding MWIS without directly taking into account the size of each bid. A **normalized weight** of vertex u in G is defined as follows:

$$\tilde{w}_v(u) \stackrel{\text{def}}{=} \frac{w_v(u)}{(d_G(u) + 1)^\beta} \quad (2)$$

where β is a configurable parameter. The normalized weight greedy method (WG, for short) is a heuristic scheme that selects vertices in G in a non-decreasing order of $\tilde{w}_v(u)$ in a greedy manner. It is known that the approximation ratio of WG is $1/\Delta_G$ provided that $\beta = 1.0$ [9].

It should be noted that several variations of WG have also been proposed in the literature. An idea for the improvement is to use the following normalized weight instead of that in Equation (2) (the resultant heuristic scheme will be referred to as WWG hereafter):

$$\tilde{w}'_v(u) \stackrel{\text{def}}{=} \frac{w_v(u)}{(\sum_{v \in N_G(u)} w_v(v) + 1)^\beta}, \quad (3)$$

where β is a configurable parameter. Another idea is to select vertices in the *non-increasing* order of the normalized weight. It has been shown that this modification generates a scheme with approximation ratio at least $1/(\Delta + 1)$ and at most $1/\Delta$ provided that $\beta = 1.0$ [9].

3.2 Feasibility Check

The next problem we have to consider is how to quickly identify feasible bids with respect to a given partial solution $\mathcal{B}'$. Previous techniques on this problem can be classified into two categories; i.e., static and dynamic approaches.

As a static approach, a discrimination of bids into *bins* is commonly used in many heuristic search schemes [3, 10, 13]. In this method, $\mathcal{B}$ is initially partitioned into m subsets $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_m$, in the following manner: a bid $B_i = \langle \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}, v_i \rangle$ is contained into subset $\mathcal{B}_j$ iff $j = \min\{i_1, i_2, \ldots, i_k\}$. By combining this method with an appropriate bid ordering scheme, we could identify a feasible bid with higher priority very quickly, although it statically fixes the priority of "goods" contained in bids (e.g., bids containing good x_1 will always be given a higher priority than the others). Note that such a static ordering could significantly lose the flexibility of the resultant scheme. For example, we could construct an instance for which a bin-based scheme exhibits a provably bad performance. Other static approaches include schemes that discrimenate bids by their size, and select only bidsets with size smaller than $\sqrt{m}$ [4].

As a dynamic approach, a counter based scheme is proposed in [11]. In this scheme, each bid B_i is associated with a counter c_i that counts how many times the corresponding bid is excluded by the other bids, where we say that bid B_i is excluded by bid B_j when a bid B_j such that $B_i \cap B_j \neq \emptyset$ is included in $\mathcal{B}'$. All counters c_i $(1 \leq i \leq n)$ are initialized to zero. The value of c_i is incremented by one when B_i is excluded by some bid B_j, and is decremented by one when a bid that excluded B_i is removed from $\mathcal{B}'$. By using the notion of counters, we can easily check the feasibility of a bid by simply checking whether the value of c_i is zero or not.

3.3 Search Method

In a search tree for solving WDP, each node corresponds to a bid, and a path from the root to a leaf corresponds to a feasible solution. Note that the search tree is an ordered tree (i.e., children of a node are totally odered), and in our setting, the depth of a node in the tree is at most $m(=|\mathcal{B}|)$, where the depth of the root is defined to be zero.

Limited discrepancy search (LDS, for short) is a generic technique to realize time-efficient search schemes [14]. An LDS-based search scheme for solving WDP is proposed in [10]. The basic idea of LDS is to control the width of expansion made at each internal node by parameter D, that is referred to as the **discrepancy**. Each internal node v in the seach tree is associated with an integer d_v called discrepancy number, the value of which is determined as follows: 1) the discrepancy number of the root node is zero; and 2) if a node v has a discrepancy number d_v and a successor u of v is given the ith order among all children of v, the discrepancy number of u is determined as $d_u = d_v + (i-1)$. In LDS-based schemes, a node will be expanded only when its discrepancy number does not exceed D. Note that in LDS, the trade-off between the execution time and the quality of solution could be controlled by appropriately adjusting the value of parameter D, or by consecutively increasing D from zero.

4 Proposed Heuristics

In this paper, we propose three bid ordering methods for solving WDP that could be used in general heuristic search schemes with an appropriate feasibility check mechanism and heuristic search techniques.

The first idea is to refine the normalized weight used in WG. A refined normalized weight of vertex u in G, denoted by $\phi(u)$, is defined as follows:

$$\phi(u) \stackrel{\text{def}}{=} \frac{w_v(u)}{(\sum_{v\in N_G(u)} w_e(u,v)+1)^{\alpha}}, \qquad (4)$$

where α is a configurable parameter. A refined normalized weight greedy method (EW, for short) is a heuristic scheme that selects vertices in G in a non-decreasing order of $\phi(u)$ in a greedy manner. Note that this extension of WG is intended to take into account the edge weight of a given bid graph, that is usually assumed to be uniform in normal MWIS or MIS. The above idea could further be extended by taking into account the weight of adjancent vertices. More concretely, we can use the following heuristic function ψ instead of ϕ used in EW (the resultant heuristic scheme will be referred to as EWVE, in what follows):

$$\psi(u) \stackrel{\text{def}}{=} \frac{\phi(u)}{(\sum_{v\in N_G(u)} w_v(v)+1)^{\beta}}, \qquad (5)$$

where β is a configurable parameter.

Our next idea is to evaluate the appropriateness of a vertex as a candidate vertex by comparing its priority with that of its neighboring vertices. Let $f(u)$ be any estimated value of a vertex u corresponding to a bid; we can use either w_v, $\tilde{w}_v$, ϕ, or ψ as a candidate for function f. A **potential** of vertex u with respect to function f is defined as:

$$\omega(u) \stackrel{\text{def}}{=} \frac{f(u)}{(\sum_{v\in N_G(u)} f(v)+1)^{\gamma}}, \qquad (6)$$

where γ is a configurable parameter. The potential based greedy method (PG, for short) is a heuristic scheme that selects vertices in G in a non-increasing order of $\omega(u)$ in a greedy manner. Note that PG is intended to collect a global information compared with the other schemes.

5 Experimental Evaluation

5.1 Class of Instances

We conducted several experiments to evaluate the goodness of proposed bid ordering methods. All experiments are conducted in the following environment: CPU: 2.53GHz PentiumIV; Memory: 1024MB; OS: FreeBSD 4.7; and all programs are written in C language.

In the experiments, we used the following five classes of instances, each of which models bid distributions observed in real-world applications. The first two classes are benchmarks used in [13], and the remaining three classes are extracted from [7].

Random: Each bid B_i is constructed by selecting k_i goods from S without replacement and by assigning a bid value v_i to it, where k_i is a random value drawn from $\{1,2,\ldots,m'\}$, where $m' \le m$, and v_i is a random value drawn from $\{1,2,\ldots,Max\}$.

Uniform: Modify Random in such a way that the size of each bid is fixed to a constant k.

Paths: Let $H_1 = (V_1, E_1)$ be a graph that models an interconnection network such as truck routes and natural gas pipeline networks. A vertex in V_1 is associated with a point in the Euclidean plane, and an edge in E_1 is associated with a line segment connecting two points in V_1. In this class of instances, an edge corresponds to a good, and it is requested that a bid should be submitted on a set of goods that form a simple path in graph H_1.

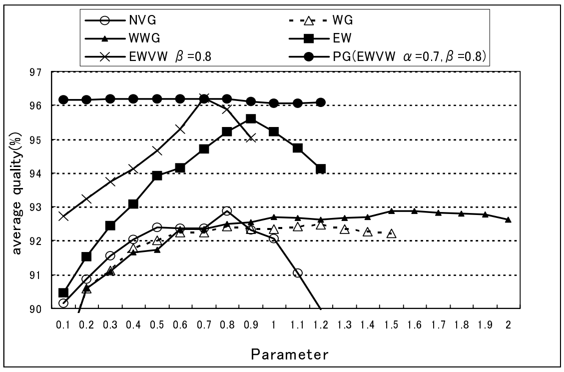


Figure 1: The effect of configurable parameters to the performance of bid ordering schemes.

Regions: Let H_2 be a graph with vertex set S and edge set E_2, where S corresponds to a set of points in the Euclidean plane (e.g., drilling right of a mining point), that is weighted by the offset value shared by all bidders, and E_2 represents the geometrical adjacency of those goods in the plane, that is also weighted by their strength. In this class of instances, a bid B_i is sequentially constructed as follows: initially, it contains a single good, and in each iteration, it selects a good probabilistically, where the probability of selecting a good adjacent to goods contained in the bid is set to be higher than that of the other goods. The value of B_i is determined as a linear combination of the following factors: sum of bidder's preferences and the offset values of each good contained in the bid, and $|S_i|^{1.2}$.

Arbitrary: Modify Regions in such a way that every pair of vertices is connected by an edge and the weight of edge is not uniform.

5.2 Preliminary Experiments

Before proceeding to a detailed analysis, we conducted preliminary experiments to determine an appropriate value of several configurable parameters used in heuristic bid ordering schemes (e.g., α, β, or γ). In the experiments, we used an LDS-based scheme with a fixed descrepancy (i.e., $D = 0$ or 3), and under the scheme, we evaluated the goodness of heuristics in terms of the average ratio to an optimal solution[2], where we generated 20 instances for each class. Figure 1 illustrates the result. According to the result, we fix the value of parameters as in Table 1, in the following experiments.

[2] An optimal value is used as a reference except for Uniform, since we could not obtain an optimal solution to Uniform instances within 48 hours. Instead of that, for Uniform, we used a trivial upper bound on the maximum revenue as a reference, that can be calculated as in Equation (1).

Table 1: Parameters used in the experiments.

NVG	$\alpha = 0.8$
WG	$\beta = 1.2$
WWG	$\beta = 1.6$
EW	$\alpha = 0.9$
EWVW	$\alpha = 0.7,\ \beta = 0.8$
PG(MNWG)	$\gamma = 0.7$

As the next step, we conducted a series of experiments to make a *rough comparison* of bid ordering schemes. Table 2 summarizes the result. In this table, NVEr denotes an NVE with *recalculation* of estimated revenue at each node in the search tree, and NVEb denotes an NVE with a quick descremenation of feasible bids by using the notion of bins (see Section 3.2, for detail).

From the table, we can make several interesting observations: First, among three schemes based on the normalization of bid values by their size, 1) NVErb is better than NVE for $D = 3$, while it is worse than NVE for $D = 0$, and 2) NVErb is better than NVG for $D = 3$ in three classes of instances, and a significant improvement on the average revenue (compared with NVG) is observed by increasing D, in all of the five classes (it implies that the use of bins really improves the performance of NVE for $D = 3$). Next, among four schemes based on local informations on bid graph, WG exhibits almost the same performance with WWG. Finally, we could observe that PW(EWVW) does not significantly improve the performance of EWVE while it uses a much more global information than the others.

According to the above observations, in the following experiments, we focus our attention to the following five heuristic schemes: NVEb, NVErb, WWG, EW, and EWVW.

5.3 Bidset with a Uniform Structure

As the first step of investigation, we examined the performance of schemes for instances with a uniform bid structure (i.e., Random and Uniform given in Section 5.1). More concretely, we used the following three suites of instances in the experiments (each suite consists of 20 randomly generated instances): Uniform with bid size three (U3), Uniform with bid size ten (U10), and Random with average bid size three (R3). Basic properties of those suites are summerized in Table 3. In the experiments, we measured the *anytime performance* of schemes for the first 60 seconds under the following LDS-based scheme: D is initialized to zero, and after complet-

Table 2: Average revenue of each heuristics for a fixed descrepancy D (%(D=0)/%(D=3)).

suite	NVG	NVEr	NVErb	WG	WWG	EW	EWVW	PG(EWVW)
random	90.9/96.3	90.2/92.6	88.7/97.9	95.2/98.5	95.7/99.5	94.5/98.3	95.4/98.8	95.2/98.8
uniform	76.6/80.2	76.6/78.0	74.8/82.8	80.0/85.0	82.1/86.5	80.1/84.7	82.0/86.2	81.1/86.4
paths	94.0/97.1	91.6/92.9	94.6/98.5	93.7/99.0	94.6/98.9	95.2/98.6	94.8/99.1	94.5/99.1
region	86.5/95.7	87.4/93.0	85.7/94.2	90.5/91.5	89.6/91.4	90.8/98.5	93.0/99.1	92.5/99.1
arb	85.1/95.2	82.1/90.4	81.3/94.4	86.5/88.5	86.5/88.1	88.3/97.9	91.2/97.8	89.9/97.6
avg	86.6/92.9	85.6/89.4	85.0/93.6	89.2/92.5	89.7/92.9	89.8/95.6	91.3/96.2	90.6/96.2

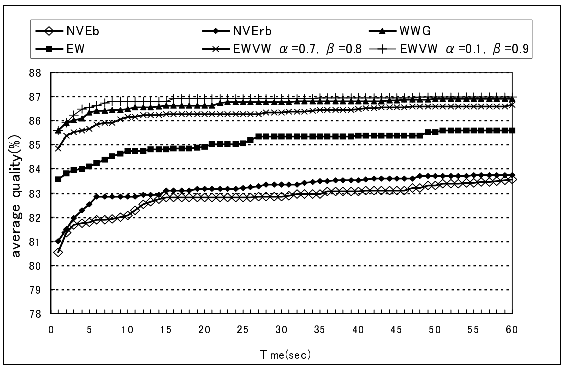
Table 3: Properties of instances with uniform bid structure (m means the average number of goods, n means the avrage number of bids, $|S|$ means the average bid size, and d and w_e mean the average degree and edge weight in the corresponding bid graph, respectively).

suite	m	n	$\|S\|$	d	w_e
U3	100.0	500.0	3.0	43.5	1.03
U10	100.0	500.0	10.0	331.9	1.50
R3	100.0	500.0	3.0	57.0	1.06

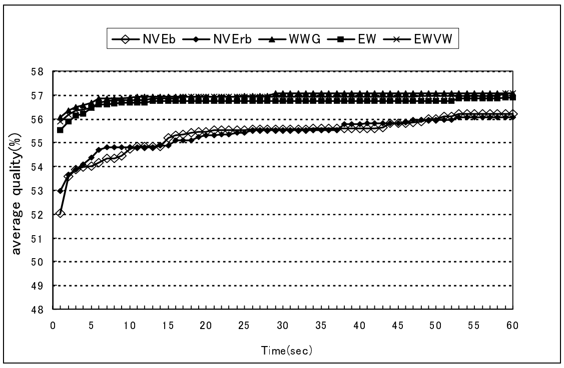
ing the search for the current D, the value of D is incremented by one, and it restarts the heuristic search with the new descrepancy D.

Figure 2 (a) shows the result for U3. From the figure, we can observe that EW exhibits a worse performance than WWG. The badness of EW is probably because of the uniformity of edge weight in U3 (recall that EW is a heuristic based on the normalization by the edge weight). In fact, in U3, 97 % of edges are weighted by one, and only 3 % of edges are weighted by two. On the other hand, for scheme EWVW, that takes into account both edge and vertex weights, we could make the following observations: 1) it exhibits a worse performance than WWG for parameters $\alpha = 0.7$ and $\beta = 0.8$ that gives almost the same "weighting" to both weights, but 2) it exhibits a better performanmce for parameters $\alpha = 0.1$ and $\beta = 0.9$ that gives more "weighting" to the vertex weights. The above observation implies that, in order to fully enjoy the combination of edge and vertex weights in EWVW, we have to clarify the impact of edge weight in bid ordering heuristics in more detail. A natural conjecture concerned with this question is "EW (and EWVW) can exhibit a good performance when there is a sufficiently large variance in the edge weight of the given bid graph."

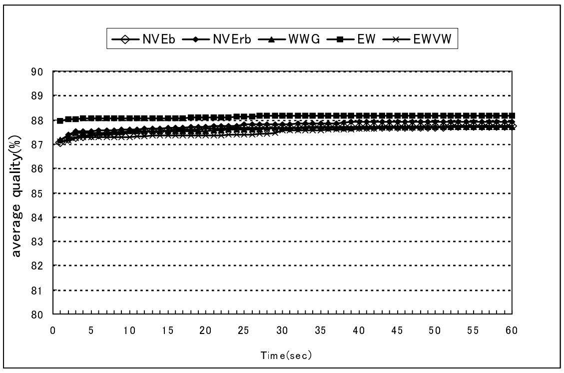
In order to verify the above conjecture, we conducted an additional experiment for suites U10 and R3 (U10 is a suite with average edge weight 1.5 and R3 is a suite with average edge weight 1.06). Figures 2 (b) and (c) illustrate the result. As is shown in



(a) U3.

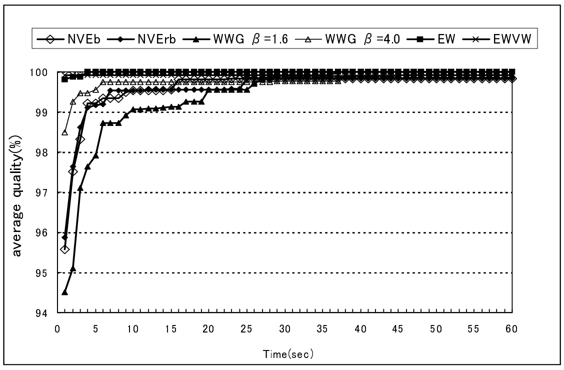


(b) U10.

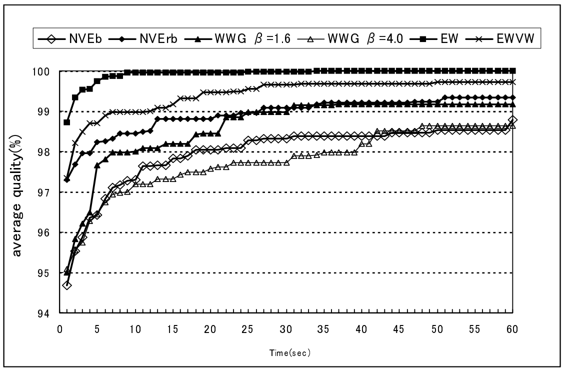


(c) R3.

Figure 2: Anytime performance of heuristics for uniform instances.



(a)



(b)

Figure 3: Anytime performance for Regions.

the figures, the performance of EW really increases by enlarging the variance of edge weight in the instance; more concretely, the gap between EW and WWG for the first three seconds reduces from 2.3 % to 0.3 % by increasing the bid size in Uniform from three to ten (see (b)), and the relative position of those schemes interchanges by replacing suite U3 by R3 (see (c)).

5.4 Stability for Non-Uniform Bid Structure

Next, we examined the performance of schemes for instances with a non-uniform bid structure. In the following, we merely illustrate results for Regions, since 1) Path is easily solved by all schemes, and 2) Arbitrary exhibits almost the same behavior with Regions.

Figure 3 shows the anytime performance for two suites of Regions, where the second one is constructed from the first one by doubling the number of goods (other parameters are kept identical). From the figures, we can observe that EW exhibit a very good performance in both cases, that is probably because the variance of edge weight is sufficiently large. In fact, by doubling the number of goods, the variance of edge weight reduces, that really degrades the convergence to an optimal solution by EW.

In contrast to that, the performance of WWG is very sensitive to the selection of parameter β; in fact, in Figure 3 (a), it exhibits a very bad performance when $\beta = 1.6$, while it is satisfactory when $\beta = 4.0$. Conversely, in Figure 3 (b) (i.e., by doubling the number of goods), $\beta = 1.6$ becomes a better selection than $\beta = 4.0$. It is probably because the relative importance of edge weight significantly varries by doubling the number of goods, and more importantly, WWG is inherently sensitive to the change of those factors.

5.5 Scalability

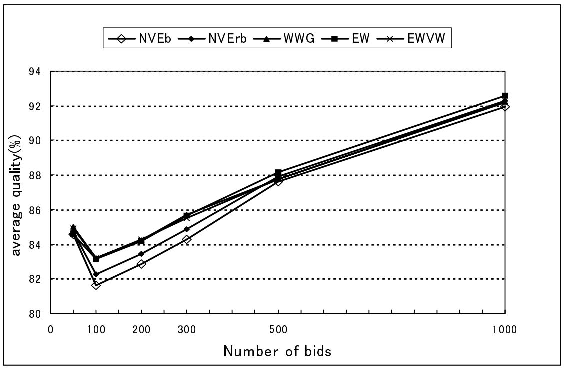
Finally, we examined the scalability of schemes with respect to bid number n. We examined the performance of schemes for Random over 100 goods with average bid size three, by varying the number of bids n from 50 to 1000 (for each n, we randomly generated 20 instances, as before). The performance is measured by the execution time and the quality of the solution, where the bid ordering schemes are implemented over an LDS-based scheme with a fixed descrepancy $D = 3$.

Figure 4 shows the result, where (a) represents the quality of solution normalized by a trivial upper bound and (b) represents the execution time. As is shown in Figure 4 (a), EW exhibits the best performance among others for sufficiently large $n(\geq 500)$. It should be worth noting that the normalized quality of the solution increases by either increasing or decreasing bid number n, that is because: 1) the instance becomes easy for small n's, and 2) the increase of m increases the redundancy in selecting a feasible subset of bids with a higher revenue, that can improve the relative quality of the solution. On the other hand, as is shown in (b), three schemes based on the calculation over the bid graph (i.e., WWG, EW, and EWVW) take a longer execution time than the other two schemes, that is probably because of the necessity of re-ordering of feasible bids at each point of the heuristic search.

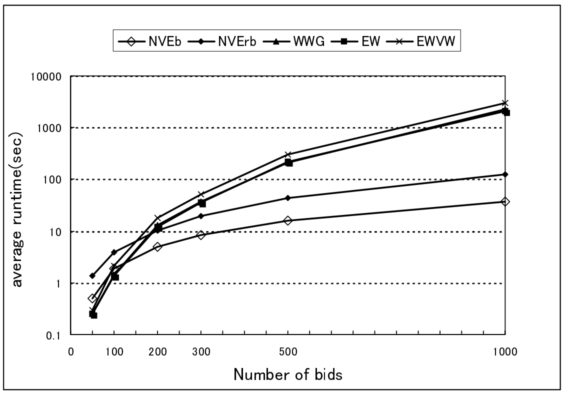
However, the increase of the total execution time for a fixed D does not directly affect the anytime performance of the overall scheme; i.e., by using EW, we could still obtain a better solution at anytime of the execution of the heuristic search.

6 Concluding Remarks

In this paper, we studied the winner determination problem in combinatorial auctions, and proposed



(a) Quality of solutions.



(b) Average runtime.

Figure 4: Scalability of heuristics for R3.

three heuristic bid ordering schemes for solving the problem. An extensive comparison of those schemes with conventional ones, with respect to the quality of solutions, execution time, and the anytime performance was conducted, and the result of experiments implies that the first two schemes EW and EWVW, that take into account the *number* of goods shared by conflicting bids, exhibit a particularly good and stable behavior compared with the other schemes. Although several conventional schemes (e.g., WWG) could also exhibit a good performance under an appropriate selection of configurable parameters, the optimal value of parameters severely depends on several factors, such as the number of bids and the bid size. Hence, at least when the characteristics of the given instance is not clear, it should be a good way to use our proposed schemes as a basic scheme in solving WDP.

References

[1] M.-S. Chang, F.-H. Wang. Efficient algorithms for the maximum weight clique and maximum weight independent set problems on permutation graphs. *Information Processing Letters*, 43(6): 293-295, 1992.

[2] S. de Vries, R. Vohra. Combinatorial auctions: A survey. *to appear in INFORMS Journal on Computing*, 15, 2003.

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

[4] M. M. Halldórsson. Approximation of weighted independent set and hereditary subset problems. *Journal of Graphs Algorithms and Applications*, 4(1): 1–16, 2000.

[5] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182: 105–142, 1999.

[6] D. Lehmann, L. Ita O'Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. *J. Assoc. Comput. Mach.*, 49(5): 577–602, 2002.

[7] K. Leyton-Brown, M. Pearson, and Y. Shoham. Towards a universal test suite for combinatorial auction algorithms. In *ACM Conf. on Electronic Commerce*, pages 66–76, 2000.

[8] M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8): 1131–1147, 1998.

[9] S. Sakai, M. Togasaki, K. Yamazaki. A note on greedy algorithms for maximum weighted independent set problem. *to appear in Discrete Applied Mathematics.*

[10] Y. Sakurai, M. Yokoo, K. Kamei. An efficient approximate algorithm for winner determination in combinatorial auctions. In *ACM Conf. on Electronic Commerce*, pages 30–37, 2000.

[11] T. Sandholm and S. Suri. Improved algorithms for optimal winner determination in combinatorial auctions and generalizations. In *Proc. AAAI*, pages 90–97, 2000.

[12] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proc. IJCAI'01*, pages 1102–1108, 2001.

[13] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2): 1–54, 2002.

[14] W. D. Harvey, M. L. Ginsberg. Limited discrepancy search. In *Proc. IJCAI'95*, Vol.1, pages 607–615, 1995.