

On the Fastest Vickrey Algorithm

Elena Grigorieva · P. Jean-Jacques Herings ·
Rudolf Müller · Dries Vermeulen

Received: 21 May 2007 / Accepted: 27 January 2009 / Published online: 14 February 2009
© The Author(s) 2009. This article is published with open access at Springerlink.com

Abstract We investigate the algorithmic performance of Vickrey-Clarke-Groves mechanisms in the single item case. We provide a formal definition of a Vickrey algorithm for this framework, and give a number of examples of Vickrey algorithms. We consider three performance criteria, one corresponding to a Pareto criterion, one to worst-case analysis, and one related to first-order stochastic dominance. We show that Pareto best Vickrey algorithms do not exist and that worst-case analysis is of no use in discriminating between Vickrey algorithms. For the case of two bidders, we show that the bisection auction stochastically dominates all Vickrey algorithms. We extend our analysis to the study of weak Vickrey algorithms and winner determination algorithms. For the case of two bidders, we show that the One-Search algorithm stochastically dominates all column monotonic weak Vickrey algorithms and that a suitably adjusted version of the bisection algorithm, the WD bisection algorithm,

E. Grigorieva acknowledges support by the Dutch Science Foundation NWO through grant 401-01-101. P.J.J. Herings acknowledges support by the Dutch Science Foundation NWO through a VICI grant. R. Müller acknowledges support by European Commission through funds for the International Institute of Infonomics.

E. Grigorieva · R. Müller · D. Vermeulen
Department of Quantitative Economics, Maastricht University, P.O. Box 616, 6200 MD Maastricht,
The Netherlands

E. Grigorieva
e-mail: e.grigorieva@ke.unimaas.nl

R. Müller
e-mail: r.muller@ke.unimaas.nl

D. Vermeulen
e-mail: d.vermeulen@ke.unimaas.nl

P.J.J. Herings (✉)
Department of Economics, Maastricht University, P.O. Box 616, 6200 MD Maastricht,
The Netherlands
e-mail: P.herings@algec.unimaas.nl

stochastically dominates all winner determination algorithms. The WD bisection algorithm Pareto dominates all column monotonic winner determination algorithms in the n bidder case.

Keywords Single item auctions · Vickrey-Clarke-Groves implementation · Algorithms · Performance analysis

1 Introduction

Recently there has been quite some interest in designing sealed-bid auctions with limited communication [2, 3] and iterative auctions with few rounds [5, 9]. However, designs with severely limited communication or large bid levels necessarily lead to inefficient allocations for some instances, meaning that the items are not necessarily allocated to those who value them most.

In [12], we have proposed an iterative auction, called the *bisection auction*, for selling a single, indivisible item to a set of bidders who have integer valuations for this item. We have shown that in the proposed auction truth-telling is a weakly-dominant strategy and the equilibrium where all bidders follow this strategy *always* results in an efficient allocation. The bisection auction shares this property with the Vickrey auction and the English auction. However, in contrast to these auctions the bisection auction is more economical in the amount of information that bidders have to communicate about their willingness to pay. Moreover, in comparison to the English auction, it needs fewer rounds to determine the winner and the price he has to pay. In [11] we provide upper bounds on the average information revealed in the course of the auction and compare these to lower bounds for the Vickrey auction and the English auction.

Results of Green and Laffont [10] and Holmström [15] imply that under certain minor restrictions on bidders' utility functions, the Vickrey-Clarke-Groves (VCG) mechanisms are the only ones that are incentive compatible in dominant strategies and allocate efficiently. For the case of a single indivisible item any mechanism that belongs to this class is a mechanism that finds the outcome corresponding to the Vickrey auction. This outcome requires that the winner is a bidder having the highest valuation and the price he pays for the item is equal to the second-highest valuation. We can therefore interpret such an auction as an algorithm that finds the identity of all bidders with the highest valuation and the exact value of the second-highest valuation. We formally introduce Vickrey algorithms as algorithms that retrieve this information. We introduce a number of examples of Vickrey algorithms, including the direct revelation and modified direct revelation algorithm, as well as the bisection and modified bisection algorithm. Well-performing Vickrey algorithms are attractive as they lead to auctions that result in a fast allocation of the object. They also have the attractive feature that bidders only have to reveal part of their valuations.

In this paper we model a bidder's valuation by a binary code of length R . An algorithm is viewed as a series of queries that are performed on the resulting $R \times n$ matrix of zeroes and ones. A Vickrey algorithm is deemed to be fast when it needs to query only a limited number of entries (cells) in the matrix in order to find the Vickrey outcome. We study three performance criteria for Vickrey algorithms to specify what we

mean by “fast”. The first one is a Pareto criterion where one Vickrey algorithm is superior to another if it needs less steps to identify the bidders with the highest valuation and the exact value of the second-highest valuation, irrespective of the realization of the bidders’ valuations. Surprisingly, although the bisection algorithm Pareto dominates many other algorithms, including the direct revelation algorithm, we show that Pareto best algorithms do not exist.

It is widely accepted in theoretical computer science to compare the performance of two algorithms by considering the performance of the algorithms in the worst-case (see e.g. [16]). For the problem at hand, worst-case analysis makes little sense since for any algorithm there exists a realization of valuations requiring Rn queries to provide a Vickrey outcome, where n is the number of bidders and R is the length of the binary encoding of the valuations.

Therefore, we need a third, more sophisticated tool for algorithm comparison. Given an algorithm we count the number of realizations of valuations for which the number of queries is at most k , for some $0 \leq k \leq Rn$. We say that algorithm A^1 stochastically dominates algorithm A^2 if for every k the number of valuations on which algorithm A^1 finds the Vickrey outcome by performing at most k queries is not less than the number for algorithm A^2 . For the case of two bidders we show that the bisection algorithm stochastically dominates any other Vickrey algorithm. We discuss the complications that arise in extending this result to more than two bidders. Our counting of queries utilizes a representation of algorithms as binary trees. The trees are similar to those which are used to derive lower bounds on the worst-case number of comparisons of sorting algorithms (see, e.g., Chap. 9 of [4]).

In a Vickrey algorithm, the identity of all bidders with the highest valuation and the exact value of the second-highest valuation is determined. An alternative problem is the one where the identity of at least one bidder with the highest valuation and the exact value of the second-highest valuation is determined. An algorithm for this alternative problem is called a weak Vickrey algorithm. We introduce the One-Search algorithm and show the partial result that for the case of two bidders the One-Search algorithm is stochastically dominant within the class of so-called column monotonic weak Vickrey algorithms. It also follows that the One-Search algorithm strictly stochastically dominates the bisection algorithm for the weak Vickrey case.

Yet another interesting problem is the winner determination problem. It concerns the determination of the identity of all bidders with the highest valuation. We adjust the bisection algorithm to make it a suitable algorithm for solving this problem and call it the WD bisection algorithm. For the case of two bidders we show that the WD bisection algorithm stochastically dominates all winner determination algorithms. Moreover, for the case with n bidders, the WD bisection algorithm Pareto dominates all column monotonic winner determination algorithms. Nevertheless, it is not obvious to show that it dominates all winner determination algorithms in the weaker stochastic dominance sense.

Counting the number of bits that have to be communicated in a decentralized function computation has been studied extensively in computer science under the term *communication complexity*, see [17]. However, to the best of our knowledge, this literature has only studied worst-case or average-case communication. Our results show that there is a potential for a measure that allows to differentiate between algorithms

that are equivalent in terms of worst-case communication, and that provides a finer differentiation as average-case analysis. Also related to our problem is literature on bit probing and cell probing, see for example [20]. This literature deals with a trade-off between an enlarged binary encoding of the problem, and performance savings in solving a decision problem. Again, this literature limits algorithm comparison to worst-case and average-case analysis.

The paper is organized as follows. In Sect. 2 we give a formal definition of a Vickrey algorithm and we give a number of examples of such algorithms. In Sect. 3 we introduce two frequently used performance criteria and study how particular Vickrey algorithms perform. We show that Pareto best algorithms do not exist and argue that worst-case analysis is not very helpful. Section 4 introduces the performance criterion of stochastic dominance and shows that for the case of two bidders the bisection algorithm stochastically dominates all Vickrey algorithms. Section 5 introduces the One-Search algorithm and shows that this algorithm stochastically dominates all column monotonic weak Vickrey algorithms for the case of two bidders. Section 6 derives results for the winner determination problem and discusses the complications that arise when $n \geq 3$. Section 7 concludes.

2 Auctions as Algorithms

Suppose there is an auctioneer who wants to sell a single indivisible item to an individual in a set $N = \{1, \dots, n\}$ of bidders, where $n \geq 2$. Each bidder i has a valuation for the item that is given in a binary encoding of length $R \geq 2$.¹ Thus, the valuation of bidder i is a vector $v_i = (v_{ri})$, $r = 1, \dots, R$, representing a value $\sum_{r=1}^R v_{ri} 2^{R-r}$, where each element v_{ri} is a binary digit, so is equal to 0 or 1. The set $V_i = \{0, 1\}^R$ denotes the set of all possible realizations of bidder i 's valuation. A realization of the valuations of all bidders is therefore a matrix $v = (v_i)_{i \in N}$, where column i corresponds to the valuation v_i of bidder i . The set $V = \prod_{i \in N} \{0, 1\}^R$ is the set of all possible realizations of bidders' valuations. The valuation of a bidder is private information, so is only known to the bidder himself.

If the auctioneer wants to allocate the object in an incentive compatible and efficient way, he should use a Vickrey-Clarke-Groves mechanism. If in addition we also require the mechanism to be individual rational, the auctioneer needs to find a bidder with the highest valuation, and charge this bidder a price equal to the second-highest valuation.² The major advantage of this mechanism is that it gives the bidders the appropriate incentives to reveal their valuation truthfully.

In terms of the matrix v , the task of the auctioneer is to identify a column whose value exceeds the values of all other columns, and to determine the value of the

¹The cases $n = 1$ and $R = 1$ are trivial and not of much interest.

²This follows from theorems by Green and Laffont [10] and Holmström [15], though these theorems require that the set of valuations is convex. Using a graph theoretic approach as in [13], one can easily see that one has slightly more flexibility in the case of integer valuations in order to make the efficient mechanism incentive compatible and individual rational: the set of payments that may be charged to the winner is given by the closed interval $[k - 1, k]$, where k is the second highest valuation.

second-highest column.³ We are interested in the fastest algorithm to do so. An algorithm is represented as a sequence of queries by the auctioneer about the value of a particular coordinate of the matrix v . The well-known 2nd price, sealed bid auction would ask for the values of all coordinates of v .⁴ We will show that there are faster algorithms.

To define precisely what we mean by an algorithm, we need some more notation. For $k \in \mathbb{N}$, a history h of length k is a sequence of k binary digits, so $h = (b^1, \dots, b^k)$, where $b^j \in \{0, 1\}$, $j = 1, \dots, k$. The length of a history h is denoted by $\ell(h)$. The initial history is given by $h^0 = \emptyset$ and has length zero. The history g is a subhistory of h , denoted $g \leq h$, if the first $\ell(g)$ digits of h yield the history g . The history g is a proper subhistory of h , denoted $g < h$, if $g \neq h$ and $g \leq h$. The set $C = \{1, \dots, R\} \times N$ denotes the set of matrix coordinates, in this paper also referred to as entries or cells.

Definition 2.1 Consider a triplet $A = (H, \sigma, \varphi)$, where H is a collection of histories, and $\sigma: H \rightarrow \{0, 1\}$ is the *stopping function*. The set of non-terminal histories is $H^0 = \{h \in H \mid \sigma(h) = 0\}$. The function $\varphi: H^0 \rightarrow C$ is the *query function*. The triplet $A = (H, \varphi, \sigma)$ is an *algorithm* if:

1. $\emptyset \in H$.
2. If $h \in H$ and $\sigma(h) = 0$, then $(h, 0) \in H$ and $(h, 1) \in H$.
3. If $h \in H$ and $\sigma(h) = 1$, then $(h, 0) \notin H$ and $(h, 1) \notin H$.
4. If $h \in H$ and $g \leq h$, then $g \in H$.

The interpretation of an algorithm is as follows. First, nature selects an instance v in V . The algorithm A starts without any information, represented by the history $h^0 = \emptyset$. If $\sigma(\emptyset) = 1$, the algorithm has solved the problem and stops. Otherwise, $\sigma(\emptyset) = 0$ and the algorithm performs a *query*, meaning that it opens cell $\varphi(h^0)$ of v , which generates the history $h^1 = (v_{\varphi(h^0)})$, where $v_{\varphi(h^0)}$ is the value found in cell $\varphi(h^0)$. If $\sigma(h^1) = 1$, the algorithm stops. Otherwise, the algorithm performs another query and opens cell $\varphi(h^1)$, which generates the history $h^2 = (h^1, v_{\varphi(h^1)})$. In general, after k steps history h^k is generated. If $\sigma(h^k) = 1$, the algorithm stops. Otherwise, the algorithm opens cell $\varphi(h^k)$, which generates the history $h^{k+1} = (h^k, v_{\varphi(h^k)})$, and so forth.⁵

³A column is said to be the second highest one if there is at least one other column whose value is at least as high as the value of the column in question, and at most one other column whose value is strictly higher than the value of the column in question.

⁴This auction is usually called the Vickrey auction. We deliberately do not use this name though to avoid confusion with other notions in this paper that bear the prefix “Vickrey”.

⁵Our terminology such as “the algorithm opens a cell”, suggests direct access to bits. However, we do not think of an algorithm as an auctioneer who has direct access to the bits that encode the bidders’ valuations. Rather, we think of an algorithm as being an, admittedly rigorous, restriction of the query language of the auctioneer. The goal of this rigor is to create a setting in which the encoding of the valuation is unambiguous, giving exact meaning to the data in the valuation matrix. This is of course a restriction on the bidding language of the auctioneer, but it does not require the auctioneer to have direct access to the encoding of the valuations, and there are several ways to relax this initial assumption. Furthermore, such severe restrictions on the bidding language are common practice in electronic auction design, where

The notation of a history only includes the announced binary digits, but not the cell that contained them. This is without loss of generality, since the algorithm itself can be used to determine the queried cell. Thus, to include data regarding the queries itself—not just the answers—into histories will needlessly complicate our notation. It suffices just to remember which answers were given in which order.

Definition 2.2 An algorithm $A = (H, \sigma, \varphi)$ is a *Vickrey algorithm* if the stopping criterion σ is defined in such a way that the algorithm does not stop before the identity of all bidders with the highest valuation and the exact value of the second-highest valuation are found.⁶

The identity of all bidders with the highest valuation has been found if for each bidder it either can be decided that he has the highest valuation or it can be decided that he does not have the highest valuation. Consider an algorithm $A = (H, \sigma, \varphi)$ that has generated a history h . We associate to such a history the minimal valuation of bidder $i \in N$, $\underline{v}^i(h)$, by assigning a 0 to all unopened cells of bidder i , and the maximal valuation of bidder i , $\bar{v}^i(h)$, by assigning a 1 to all unopened cells of bidder i . At h it can be decided that a bidder i' has the highest valuation if $\underline{v}^{i'}(h) \geq \max_{i \in N \setminus \{i'\}} \bar{v}^i(h)$, and it can be decided that i' does not have the highest valuation if $\bar{v}^{i'}(h) < \max_{i \in N \setminus \{i'\}} \underline{v}^i(h)$.

The exact value of the second-highest valuation has been found if the entire valuation of a bidder that is known to have the second-highest valuation is known. More precisely, consider an algorithm $A = (H, \sigma, \varphi)$ that has generated a history h . At h the second highest valuation is known if there is a bidder $i' \in N$ with:

1. $\underline{v}^{i'}(h) = \bar{v}^{i'}(h)$;
2. there is at least one bidder $i \neq i'$ such that $\underline{v}^i(h) \geq \bar{v}^{i'}(h)$;
3. there is at most one bidder $i \neq i'$ such that $\bar{v}^i(h) > \bar{v}^{i'}(h)$.

Notice that the requirement $\underline{v}^{i'}(h) = \bar{v}^{i'}(h)$ is equivalent to the statement that all cells of bidder i have been opened. We present two examples of Vickrey algorithms.

Definition 2.3 The *direct revelation algorithm* $A^s = (H^s, \sigma^s, \varphi^s)$ is the algorithm corresponding to the 2nd price sealed bid auction, where first bidder one is queried for his valuation by asking for increasing values of r the value of cell $(r, 1)$, next the second bidder is queried, and so on, and so forth, would be defined as follows. The set H^s of histories consists of all sequences with less than or equal to Rn binary digits. For $h \in H$, $\sigma^s(h) = 0$ if $\ell(h) \leq Rn - 1$ and $\sigma^s(h) = 1$ if $\ell(h) = Rn$. For $h \in H^0$, $\varphi^s(h) = (r, i)$, where i is the largest integer less than or equal to $(\ell(h)/R) + 1$, and $r = \ell(h) + 1 - (i - 1)R$.

the binary nature of the medium dictates precisely such restrictions on the auctioneer's queries. The only real implicit assumption we make here is simply that bidders report truthfully, truthful bidding being a dominant strategy under any Vickrey algorithm, and therefore plausible bidder behavior.

⁶We require a Vickrey algorithm to find the identity of *all* bidders with the highest valuation to give them equal chance to get the object in case of ties. In Sect. 5 we show that the results depend on this exact formulation. Only requiring that at least one bidder with highest valuation needs to be identified will be shown to yield different optimal algorithms.

The direct revelation algorithm continues to make queries even after all the information needed has been retrieved. A superior alternative therefore is the modified direct revelation algorithm defined as follows.

Definition 2.4 The *modified direct revelation algorithm* $A^{ms} = (H^{ms}, \sigma^{ms}, \varphi^{ms})$ is the algorithm that stops as soon as the identity of all bidders with the highest valuation and the exact value of the second highest valuation are found.

Vickrey algorithms that stop as soon as the identity of all bidders with the highest valuation and the exact value of the second highest valuation are found, are called *proper*. The direct revelation algorithm is not proper, whereas, by definition, the modified direct revelation algorithm is.

Acyclic algorithms never perform the same query, i.e. never ask a bidder to report a specific digit of his valuation more than once. More precisely, an algorithm $A = (H, \sigma, \varphi)$ is *cyclic* if there are histories $g, h \in H$ with $g < h$ such that $\varphi(g) = \varphi(h)$. Otherwise it is *acyclic*.

Obviously an acyclic algorithm always ends within Rn performed queries. Both the direct revelation algorithm and the modified direct revelation algorithm are acyclic.

Another Vickrey algorithm is the algorithm corresponding to the bisection auction proposed in [12]. The bisection auction has R rounds, with the auctioneer announcing a price in each round. The first price equals 2^{R-1} , the middle of the initial interval $[0, 2^R)$. Bidders report their demand at the current price by sealed bids. A *yes*-bid stands for the announcement to be willing to buy at the current price, a *no*-bid for the contrary. As a function of these bids, the auctioneer announces the price of the next round.

In case there are at least two bidders submitting a *yes*-bid, the price goes up to the middle of the upper half interval, i.e. the interval $[2^{R-1}, 2^R)$. The bidders that are allowed to participate actively in the next round are the ones that said *yes* and they are competing for the object in the price range $[2^{R-1}, 2^R)$. The other bidders drop out of the auction.

In case there is at most one bidder saying *yes*, attention shifts to the lower half interval, i.e. the interval $[0, 2^{R-1})$ and the price goes down to the middle of this interval. Two different things can happen now. First, if no one has submitted a *yes*-bid then all active bidders remain active in the next round. In the other case there is a single bidder that submitted a *yes*-bid. This bidder now becomes the winner and he gets the object. Nevertheless the auction doesn't end, but enters a price-determination phase. The active bidders in the next round are the ones that were active in the previous round minus the winner. The remaining active bidders are competing on the lower half interval $[0, 2^{R-1})$. The winner is no longer active, and the auctioneer by default considers him to say *yes* to all prices that are proposed beyond the moment he became the winner. Apart from this, the way it is decided whether the price should go up or down is not any different from the way this is decided in the winner-determination phase. In each round depending on submitted bids we subsequently restrict attention to either the lower or the upper half of the current interval.

Iterating this procedure will eventually yield a winner and a price. When in no round precisely one bidder said *yes*, at least two bidders will still be active after R

rounds, and the object is assigned by a lottery to one of them. The price is uniquely determined because in each round the length of the current interval goes down by a factor of two, so after R rounds the resulting interval is of length 1, and since it is a half-open interval, it contains exactly one integer. This integer is declared to be the price the winner of the auction has to pay for the object.

The bisection auction implicitly defines the bisection algorithm, denoted by B , and opens cells of the matrix v in R steps. In step r the algorithm B opens cells in the r th row of v . Inside a step the cells can be opened in an arbitrary order, but to be specific we require this order to correspond to the ranking $\{1, \dots, n\}$ of the bidders. To define cells to be opened in a step we introduce sets A_r , W_r , and Y_r , with as interpretation the set of active bidders, the set of winning bidders, and the set of bidders saying *yes*. Initially, $A_1 = N$ and $W_1 = \emptyset$. The set Y_1 will be determined by the algorithm.

In step r the bisection Algorithm B :

1. Opens cell (r, i) for all $i \in A_r$, with (r, i^1) before (r, i^2) when $i^1 < i^2$;
2. Defines $Y_r = \{i \in A_r \mid v_{ri} = 1\}$;
3. Defines A_{r+1} and W_{r+1} as follows:
 - if $|Y_r| = 0$, then $W_{r+1} = W_r$ and $A_{r+1} = A_r$;
 - if $|Y_r| = 1$ and $W_r = \emptyset$, then $W_{r+1} = Y_r$ and $A_{r+1} = A_r \setminus Y_r$;
 - if $|Y_r| = 1$ and $W_r \neq \emptyset$ or if $|Y_r| > 1$, then $W_{r+1} = W_r$ and $A_{r+1} = Y_r$.

After R steps the bisection algorithm opens all cells of all valuations from the set A_{R+1} . The valuation of a bidder in this set equals the second highest valuation. If W_{R+1} is non-empty, it contains the bidder with the highest valuation. Otherwise, the set A_{R+1} contains at least two bidders, all of them having the highest valuation [12].

Definition 2.5 The *bisection algorithm* $B = (H^B, \sigma^B, \varphi^B)$ is the algorithm defined in previous paragraphs.

Example 2.6 Consider the case where $n = 2$, $R = 3$, and

$$v = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}.$$

The bisection algorithm will open the cells $(1, 1)$, $(1, 2)$, $(2, 1)$, $(2, 2)$, and $(3, 2)$ and generates the terminal history $h = (0, 0, 1, 0, 0)$. For this instance v , it needs one step less than both the direct revelation algorithm and the modified direct revelation algorithm.

It can easily be checked that the bisection algorithm B is acyclic. Also, for $n = 2$, it is proper. The following example however shows that the bisection algorithm is not proper for $n \geq 3$.

Example 2.7 Consider the case where $n = 3$, $R = 2$, and

$$v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

The bisection algorithm will open the cells (1, 1), (1, 2), (1, 3), (2, 2), and (2, 3) and generates the terminal history $h = (1, 0, 0, 1, 0)$. For this instance v , the bisection auction needs five steps. However, it could have terminated after four steps, since at that point the identity of all bidders with the highest valuation and the exact value of the second-highest valuation were known.

Definition 2.8 The *modified bisection algorithm* $B^m = (H^{mB}, \sigma^{mB}, \varphi^{mB})$ is the algorithm that follows the same procedure as the bisection algorithm, but stops as soon as the identity of all bidders with the highest valuation and the exact value of the second-highest valuation are found.

The modified bisection algorithm is acyclic and, by definition, proper. For $n = 2$, the bisection algorithm and the modified bisection algorithm coincide.

3 Performance Criteria for Algorithms

In this section we discuss two widely accepted criteria for the comparison of algorithms, namely the Pareto criterion, and the worst-case criterion. We argue that in our setting both criteria are not very useful. The Pareto criterion does exclude clearly inferior algorithms such as cyclic and non-proper algorithms, but does not discriminate much beyond these obvious inferiorities, and it turns out that a Pareto-best Vickrey algorithm does not exist. The worst-case criterion is also not useful because all acyclic algorithms turn out to be worst-case equivalent.

Given an algorithm A , $\ell_A: V \rightarrow \mathbb{N} \cup \{\infty\}$ denotes the function that assigns to each instance v in V the length of the history after which the algorithm stops, or equivalently the number of queries that A performs on v . Among all Vickrey algorithms we want to find the one with the most favorable ℓ_A .

One approach consists of preferring an algorithm A^1 to an algorithm A^2 if for all $v \in V$, $\ell_{A^1}(v) \leq \ell_{A^2}(v)$. We refer to this concept as Pareto preferred.

Definition 3.1 An algorithm A^1 is *Pareto preferred* to an algorithm A^2 if for all $v \in V$, $\ell_{A^1}(v) \leq \ell_{A^2}(v)$. The algorithm A^1 is *strictly Pareto preferred* to A^2 if A^1 is Pareto preferred to A^2 , and A^2 is not Pareto preferred to A^1 . A Vickrey algorithm is *Pareto best* if it is Pareto preferred to any other Vickrey algorithm.

Not all Vickrey algorithms are candidates for being Pareto best. As the next proposition states, neither non-proper Vickrey algorithms nor cyclic Vickrey algorithms are Pareto best. The proof is easy, and therefore omitted.

Proposition 3.2 *For any Vickrey algorithm A that is not proper there is a Vickrey algorithm A' that is strictly Pareto preferred to A . For any cyclic Vickrey algorithm A there is a Vickrey algorithm A' that is strictly Pareto preferred to A .*

Thus both cyclic and non-proper Vickrey algorithms are not Pareto best. However, the Pareto criterion has only limited further use beyond these straightforward observations. There do exist rather silly algorithms that accidentally perform very well for

particular instances. An example is the reversed modified direct revelation algorithm, that starts by opening cell (R, n) , next cell $(R - 1, n)$, and continues in this way until all cells of bidder n are opened. The algorithm then continues with the same operations for bidders $n - 1, n - 2$, and so on, until the identities of all bidders with the highest valuation, and the exact value of the second highest valuation have been found.

Now consider the case considered in Example 2.6 with $n = 2, R = 3$, and

$$v = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}.$$

For this instance, the length of the direct revelation algorithm and the modified direct revelation algorithm is six, the length of the bisection algorithm and the modified bisection algorithm is five, and the length of the reversed modified direct revelation algorithm is four. Indeed, this algorithm opens cell $(3, 1)$ after history $(0, 0, 0)$, concludes that bidder 1 is the unique bidder with the highest valuation, and knows the second highest valuation to be equal to zero.

This already indicates that the criterion of being Pareto best is not appropriate for the comparison of algorithms in our setting. The following result shows that Pareto best Vickrey algorithms do indeed not exist.

Theorem 3.3 *Pareto best Vickrey algorithms do not exist.*

Proof We present the proof for the case $n = 2$. For $n \geq 3$ the proof runs along similar lines. Based on Proposition 3.2, we can restrict our attention to proper acyclic algorithms when searching for a Pareto best Vickrey algorithm. Consider the instances

$$v^1 = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad v^2 = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Any Pareto best algorithm will only open all cells of the second column together with cell $(R, 1)$ for v^1 and the all cells of the first column together with cell $(R, 2)$ for v^2 . Therefore, any Pareto best algorithm has to start with opening a cell in row R . However, applying the same line of reasoning to the instances

$$v^3 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad v^4 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}$$

shows that any Pareto best algorithm starts by opening a cell in the first row. It follows that Pareto best Vickrey algorithms do not exist. \square

Another approach that is often taken when assessing the performance of an algorithm is to perform a worst-case analysis. A Vickrey algorithm A^1 is *worst-case*

preferred to a Vickrey algorithm A^2 if $\max_{v \in V} \ell_{A^1}(v) \leq \max_{v \in V} \ell_{A^2}(v)$. A Vickrey algorithm A is *worst-case best* if A is worst-case preferred to any other Vickrey algorithm.

In a worst-case analysis of the algorithms under consideration, an upper bound is given for the number of queries the algorithm performs for a fixed number of bidders and length of valuations. Clearly, for any two Vickrey algorithms A^1 and A^2 , A^1 is worst-case preferred to A^2 whenever A^1 is Pareto preferred to A^2 . However, worst-case analysis does not provide any insights when applied to Vickrey algorithms, as all acyclic Vickrey algorithms are equivalent to each other according to this criterion.

Theorem 3.4 *Let A^1 and A^2 be two acyclic Vickrey algorithms. Then A^1 is worst-case equivalent to A^2 .*

The proof is simple. Any acyclic Vickrey algorithm terminates in less than or equal to Rn steps, and the upper bound is attained for all instances v with $v^i = v^j$ for all $i, j \in N$, as the only way to decide whether a bidder has the highest valuation is to open all his cells.

4 Stochastic Dominance of the Bisection Algorithm

The Pareto criterion and worst-case analysis are of no use for comparison of Vickrey algorithms. The criterion we advocate relies on the concept of the characteristic function of an algorithm.

Definition 4.1 The *characteristic function* of an algorithm A is the function $F_A: \mathbb{N} \rightarrow \{0\} \cup \mathbb{N}$, where $F_A(k)$ equals the number of instances in V for which $\ell_A(v) \leq k$.

Consider an acyclic Vickrey algorithm A . Since the number of elements in V equals 2^{Rn} , and opening all cells definitely yields the required information, it holds that $F_A(k) = 2^{Rn}$ for all $k \geq Rn$. When all bidders have identical valuations, any Vickrey algorithm will open all cells. Therefore $F_A(Rn) - F_A(Rn - 1) \geq 2^R$. Also, the characteristic function of an algorithm enables us to identify the number of instances for which the algorithm terminates in exactly k steps, since this number equals $F_A(k) - F_A(k - 1)$.

Definition 4.2 An algorithm A^1 *stochastically dominates* an algorithm A^2 if $F_{A^1}(k) \geq F_{A^2}(k)$ for all $k \in \mathbb{N}$. An algorithm A^1 *strictly stochastically dominates* A^2 when A^1 stochastically dominates A^2 , but A^2 does not stochastically dominate A^1 .

Notice that an algorithm A^1 stochastically dominates an algorithm A^2 precisely when the graph of F_{A^1} is never below the graph of F_{A^2} .

We use this criterion to compare Vickrey algorithms. The most favorable instance is one where in the first row one cell contains a 1 and the other entries are 0, since the bidder with the highest valuation is then known. If entry $v_{1,i} = 0$, but $v_{r,i} = 1$ for all $r \geq 2$, then opening the cells in column i determines the exact value of the

second-highest valuation. Any Vickrey algorithm has to open all the cells of the bidder with the second-highest valuation. We therefore obtain that for all Vickrey algorithms $F_A(k) = 0$ for all $k \leq R + n - 2$, and there are Vickrey algorithms A such that $F_A(R + n - 1) > 0$.

Definition 4.3 A Vickrey algorithm A is *stochastically dominant* when A stochastically dominates any other Vickrey algorithm.

Definition 4.2 is equivalent to the definition of first-order stochastic dominance, as it is defined in e.g. [7] and [18], when we assume a uniform distribution of instances. Indeed, for the case at hand, it would state that an algorithm A^1 first-order stochastically dominates an algorithm A^2 if for every number of queries k , the probability of stopping after at most k queries is not smaller for A^2 than it is for A^1 . Rescaling the characteristic function along the y-axis by dividing by the total number of instances 2^{Rn} , we get the function that for every number of queries shows the probability that the algorithm performs at most as many queries if instances were randomly chosen. A direct consequence is that, when we assume instances to be uniformly distributed, an algorithm A^1 that (strictly) stochastically dominates an algorithm A^2 has a (strictly) shorter average running time than the algorithm A^2 .

Although not used in this paper, another interesting notion directly related to stochastic dominance is the Lorenz criterion, well known in utility theory and welfare economics. The Lorenz criterion was developed in 1905 by the American economist Max Lorenz to measure inequality of income over a population. An allocation a_1 of goods over consumers Lorenz dominates another allocation a_2 of goods over consumers when, for each k , under a_1 the sum of the utilities of the k poorest consumers is at least as high as the sum of the utilities of the k poorest consumers under a_2 .⁷ The connection to stochastic dominance is evident.

Applications of stochastic dominance can be found in operations research, see for example [14, 19], and in game theory, see for example [8, 21]. Applications of the Lorenz criterion can for example be found in [1, 6].

The next result claims that stochastic dominance is weaker than the notion of Pareto best, but stronger than the notion of worst-case best. The proofs are obvious and therefore omitted.

Theorem 4.4 Let A^1 and A^2 be two algorithms. If A^1 is Pareto preferred to A^2 , then A^1 stochastically dominates A^2 . If A^1 stochastically dominates A^2 , then A^1 is worst-case preferred to A^2 .

We prove that in the case of 2 bidders, i.e. $N = \{1, 2\}$, the bisection algorithm is stochastically dominant. Recall that the bisection algorithm coincides with the modified bisection algorithm when $n = 2$, so that the bisection algorithm is proper in this

⁷“Poorest” here means “having the lowest level of utility”.

case. We use the notation

$$v = \begin{bmatrix} a_1 & b_1 \\ \vdots & \vdots \\ a_R & b_R \end{bmatrix}.$$

Definition 4.5 Let v be an instance in V . Row r of v is a *double* if $a_r = b_r$ and is a *single* otherwise.

Let $r_A: V \rightarrow \mathbb{N}$ be the function that assigns to each instance v in V the number of rows r of v for which algorithm A opens both cells $(r, 1)$ and $(r, 2)$ before it stops. Let $G_A: \mathbb{N} \rightarrow \mathbb{N}$ be the function that assigns to each k the number $G_A(k)$ of instances in $v \in V$ for which $r_A(v) \leq k$. Since a Vickrey algorithm A should find the exact value of the second-highest valuation and thus should open all the cells of a bidder with the second-highest valuation, the number of queries that an acyclic Vickrey algorithm A performs on an instance v is equal to the number of rows that A opens in V before it stops plus R .⁸

$$\ell_A(v) = r_A(v) + R.$$

For any two acyclic Vickrey algorithms A^1 and A^2 it therefore holds that $G_{A^1}(k) \geq G_{A^2}(k)$ for all k is equivalent to $F_{A^1}(k) \geq F_{A^2}(k)$ for all $k \in \mathbb{N}$. To prove that an acyclic Vickrey algorithm A^1 stochastically dominates an acyclic Vickrey algorithm A^2 , it suffices to show that $G_{A^1}(k) \geq G_{A^2}(k)$ for all $k \in \mathbb{N}$.

Lemma 4.6 Assume $n = 2$. Let A be an acyclic Vickrey algorithm. If $v \in V$ contains R doubles, then $r_A(v) = R$. For any other v , A opens at least one single before terminating.

Proof The first statement of the theorem is obvious. Consider the case where v has at least one single. If a Vickrey algorithm does not open any single, it cannot decide for both bidders whether they have the highest valuation or not. This contradicts the definition of a Vickrey algorithm. \square

If we do not require an algorithm to find the identity of all bidders with the highest valuation, then the statement of Lemma 4.6 does not hold. Consider the following example. Suppose an algorithm has performed five queries, resulting in

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & * \end{bmatrix}.$$

The information revealed is enough to find the second highest valuation, which is equal to the valuation of bidder 1, as well as the identity of a bidder with the highest valuation, bidder 2. If we are satisfied with finding only a bidder with the highest valuation we can stop without opening the remaining cell. This implies that the algorithm,

⁸We say that a row is open when all cells in that row are opened.

before it stops, neither opens all R rows nor a single. Section 5 treats algorithms that only identify some bidder with highest valuation.

Lemma 4.7 *Assume $n = 2$. Let A be an acyclic Vickrey algorithm. Then, for $k < R$,*

$$G_A(k) \leq \sum_{j=1}^k 2^{2R-j}.$$

Proof Consider the algorithm A' that is identical to A as long as no single is opened, and stops as soon as the first single is opened. Notice that A' is not a Vickrey algorithm, since it may terminate before the required information has been retrieved. Since $G_A(k) \leq G_{A'}(k)$ holds for all k , it suffices to show that

$$G_{A'}(k) = \sum_{j=1}^k 2^{2R-j}$$

for $k < R$. We construct a set M_d of instances for which the first row opened by A' has distinct values, and a set M_e of instances for which the first row opened by A' has equal values. Start with a set of instances M equal to V and let M_e and M_d be empty sets. Go through the following procedure. Take an instance v from M . Consider the step in algorithm A' where for the first time a row is opened. Without loss of generality in this step cell b_r is opened. Define the instance v' as follows:

$$\begin{aligned} a'_j &= a_j, & \text{for all } j, \\ b'_j &= b_j, & \text{for all } j \neq r, \\ b'_r &\neq b_r. \end{aligned}$$

Obviously, the instance v' belongs to M .

Clearly, either $a_r = b_r$ or $a'_r = b'_r$. If $a_r = b_r$, then define

$$\begin{aligned} M_e &:= M_e \cup \{v\}, \\ M_d &:= M_d \cup \{v'\}, \end{aligned}$$

and otherwise

$$\begin{aligned} M_e &:= M_e \cup \{v'\}, \\ M_d &:= M_d \cup \{v\}. \end{aligned}$$

In both cases $M := M \setminus \{v, v'\}$.

Repeat the procedure from the beginning until the set M is empty. It is clear from the construction that when the procedure ends we have $|M_e| = |M_d| = 2^{2R-1}$. For any instance from M_d , algorithm A' stops after opening one row. For $k < R$, let $V(k)$ be the set of instances for which algorithm A' stops after opening k rows. By Lemma 4.6, it holds that $V(1) = M_d$ and $|V(1)| = 2^{2R-1}$, so $G_{A'}(1) = |V(1)| = 2^{2R-1}$.

If $R \geq 3$, then apply the same procedure to the set $M := M_e$ and consider the step in algorithm A' where the second row is opened. Note that for any instance from M_e the first row opened is a double. At the end of the procedure we have sets M_d

and M_e . By Lemma 4.6, $V(2) = M_d$, the set of instances for which algorithm A' stops after the second row is opened. The cardinality of this set is 2^{2R-2} . It follows that $G_A(2) = 2^{2R-1} + 2^{2R-2}$.

Iterating this procedure we can show that for any $k < R$ it holds that $|V(k)| = 2^{2R-k}$. We thereby obtain the desired result that, for $k < R$,

$$G_{A'}(k) = \left| \bigcup_{j=1}^k V(j) \right| = \sum_{j=1}^k 2^{2R-j}. \quad \square$$

The bisection algorithm opens rows in a top-down fashion as long as no single is opened. As soon as the first single is opened the algorithm is able to determine the identity of the highest valuation and at the same time the identity of the second highest. Then the algorithm opens the remaining cells of the second highest valuation.

Theorem 4.8 *Assume $n = 2$. The bisection algorithm B is a stochastically dominant Vickrey algorithm.*

Proof First of all, observe that after algorithm B finds the first single no new row is opened. It holds that $V_k(B) = V_k(B')$, where B' is identical to B as long as no single is opened, and stops as soon as the first single is opened. From the proof of Lemma 4.7 it follows that for $k < R$, $|V_k(B)| = 2^{2R-k}$ and

$$G_B(k) = \sum_{j=1}^k 2^{2R-j}.$$

Using the result of Lemma 4.7, for any Vickrey algorithm A and any k we have

$$G_A(k) \leq G_B(k)$$

and consequently

$$F_A(k) \leq F_B(k). \quad \square$$

5 Weak Vickrey Algorithms

In this section we show that the results from the previous section are sensitive with respect to small changes in the algorithmic problem. We change our auction problem as follows. We no longer require that all bidders with the highest valuation are identified, but only the identity of at least one bidder with the highest valuation has to be determined. We have seen in the discussion following Lemma 4.6 that for $n = 2$ an algorithm for this problem might not have to open any single, even when the instance contains singles. We call algorithms for this new problem *weak Vickrey algorithms*.

Definition 5.1 An algorithm is a *weak Vickrey algorithm* if the stopping criterion σ is defined in such a way that the algorithm does not stop before the identity of at least one bidder with the highest valuation and the exact value of the second-highest valuation are found.

If a weak Vickrey algorithm stops at history h , it holds that there is at least one bidder i' for which $\underline{v}_{i'}(h) \geq \max_{i \in N \setminus \{i'\}} \bar{v}_i(h)$, i.e. the identity of at least one bidder with the highest valuation is known. Moreover, there should be a bidder $i'' \neq i'$ such that $\underline{v}_{i''}(h) = \bar{v}_{i''}(h)$, and there is at most one bidder $i \neq i''$ such that $\bar{v}_i(h) > \bar{v}_{i''}(h)$, i.e. the exact value of the second-highest valuation is known. For the criteria of Pareto best and worst-case best, the results for weak Vickrey algorithms are identical to those for Vickrey algorithms as the next two theorems show.

Theorem 5.2 *Pareto best weak Vickrey algorithms do not exist.*

Proof Assume first that $n = 2$ and consider the instance

$$v^1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}.$$

Let A be a weak Vickrey algorithm that opens one of the cells $(r, 1)$ for $1 \leq r \leq R$ in the first step. Since a weak Vickrey algorithm has to open all cells in column 2, we find $\ell_A(v^1) \geq R + 1$. However, the algorithm that starts by opening all cells in the second column terminates after R steps. A Pareto best weak Vickrey algorithm should therefore start by opening a cell in column 2. A completely symmetric argument leads to the conclusion that it should start by opening a cell in column 1, and we have obtained a contradiction. The argument can easily be extended to the case of n bidders. \square

Theorem 5.3 *Let A^1 and A^2 be two acyclic weak Vickrey algorithms. Then A^1 is worst-case equivalent to A^2 .*

Proof Any acyclic weak Vickrey algorithm terminates in less than or equal to Rn steps. Any acyclic weak Vickrey algorithm attains the upper bound for an instance v with $v_{ri} = 1$ for all i and r . \square

Let us now turn to the case of 2 bidders. It is easy to see that for this case a weak Vickrey algorithm is equivalent to an algorithm finding the exact value of the lowest valuation. When the exact value of the lowest valuation is found, then there is at least one bidder for which all cells are opened. The other bidder must then be a bidder having the highest valuation. We define the *One-Search algorithm*, B^* , as follows.

One-Search

Let l_1, l_2 be the last cell opened of bidder 1 and 2 respectively.

Set $l_1 = l_2 = 0$.

While the second highest valuation is not found do the following

 Start from $l_1 + 1$ to open cells for bidder 1 until the next 1 is found

 if no 1 found in rows $l_1 + 1$ to R

 bidder 2 is winner and bidder 1 has second highest valuation

 else

 Set l_1 to the row in which a 1 is found for bidder 1

 Start from $l_2 + 1$ to open cells for bidder 2 until a 1 is found or cell l_1 of bidder 2 is opened

 if a 1 is found in rows $l_2 + 1$ to $l_1 - 1$

 bidder 2 is winner and bidder 1 has second highest valuation

 if no 1 is found in rows l_2 to l_1

 bidder 1 is winner and bidder 2 has second highest valuation

 if a 1 is found in row l_1

 set l_2 equal to l_1 and continue

 open all cells of bidder with second highest valuation.

As an example, for

$$v = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix},$$

the algorithm proceeds as follows. It will find the first 1 in row 2 of bidder 1. At that point it holds that $l_1 = 2$. It will then search for a 1 in cells of bidder 2 in rows 1 to 2. As no 1 is found, it will conclude that bidder 1 is the winner and leave the while loop. It will finally open row 3 of bidder 2.

There is a number of minor modifications that would not change the performance of the One-Search algorithm. For example, for the current definition of the algorithm there are always at least as many cells opened for bidder 1 as for bidder 2 (as long as the winner is not found). The reason for this is that, whenever there are as many cells opened for bidder 1 as for bidder 2, the algorithm either stops, or it proceeds with opening cells for bidder 1. The algorithm would not be essentially affected if it would continue with opening a cell of bidder 2 instead.

So far we did not find a proof that the One-Search algorithm stochastically dominates all other weak Vickrey algorithms, nor did we find a counterexample. We show the partial result that the One-Search algorithm is stochastically dominant in the class of all column monotonic weak Vickrey algorithms.

Definition 5.4 An algorithm $A = (H, \sigma, \varphi)$ is column monotonic if it is acyclic and for every history $h \in H$ with $\varphi(h) = (r, i)$ either $r = 1$ or there is a history $g < h$ such that $\varphi(g) = (r - 1, i)$.

Theorem 5.5 Assume $n = 2$. The One-Search algorithm B^* stochastically dominates all column monotonic weak Vickrey algorithms.

The proof requires a similar preparation as the proof of Theorem 4.8. Let $r_A : V \rightarrow \{0\} \cup \mathbb{N}$ be the function that assigns to each instance v in V the number of rows r of v for which algorithm A opens both cells $(r, 1)$ and $(r, 2)$, containing a_r and b_r respectively, before it stops. Let $G_A : \{0\} \cup \mathbb{N} \rightarrow \{0\} \cup \mathbb{N}$ be the function where $G_A(k)$ equals the number of instances in V for which $r_A(v) \leq k$. Since a weak Vickrey algorithm A should find the exact value of the second-highest valuation, we have again $\ell_A(v) = r_A(v) + R$. For any two acyclic weak Vickrey algorithms A^1 and A^2 it therefore holds that $G_{A^1}(k) \geq G_{A^2}(k)$ for all k is equivalent to $F_{A^1}(k) \geq F_{A^2}(k)$ for all $k \in \mathbb{N}$.

The first step in the proof of Theorem 5.5 is to derive an explicit formula for function G in case of the One-Search algorithm.

Lemma 5.6 *Assume $n = 2$. Let B^* be the One-Search algorithm. For $0 \leq k \leq R$,*

$$G_{B^*}(k) = 2^R + \sum_{j=1}^k 2^{2R-j}.$$

Proof Algorithm B^* does not open any row before terminating if and only if $a_r = 0$ for all r . Such is the case for 2^R instances.

Next, take k with $1 \leq k \leq R - 1$. There are three cases in which B^* opens exactly k rows:

- A. When $a_r = b_r$ for $1 \leq r \leq k - 1$, $a_k = 0$, $b_k = 1$, and $a_r = 1$ for some row $r > k$. We denote the number of instances satisfying these conditions by $g_1(k)$. A straightforward calculation shows that $g_1(k) = 2^{k-1} \cdot 2^{R-k} \cdot (2^{R-k} - 1) = 2^{2R-k-1} - 2^{R-1}$.
- B. When $a_r = b_r$ for $1 \leq r \leq k - 1$, $a_k = b_k = 1$, and $a_r = 0$ for all $r > k$. We denote the number of instances satisfying these conditions by $g_2(k)$. A straightforward calculation reveals that $g_2(k) = 2^{k-1} \cdot 2^{R-k} = 2^{R-1}$.
- C. When $a_r = b_r$ for $1 \leq r \leq k - 1$, $a_k = 1$, and $b_k = 0$. We denote the number of instances satisfying these conditions by $g_3(k)$ and find that $g_3(k) = 2^{k-1} \cdot 2^{R-k} \cdot 2^{R-k} = 2^{2R-k-1}$. The result follows by observing that $g_1(k) + g_2(k) + g_3(k) = 2^{2R-k}$.

Finally, algorithm B^* opens R rows precisely when $a_r = b_r$ for $1 \leq r \leq k - 1$ and $a_k = 1$. These are exactly 2^R cases. For the sake of completeness, we add up all instances considered so far, and get

$$2^R + \sum_{k=1}^{R-1} 2^{2R-k} + 2^R = 2^{2R},$$

which confirms that we have exhausted all possibilities. \square

We show next that the expression derived in Lemma 5.6 is an upper bound for G_A for any column monotonic weak Vickrey algorithm A , thereby proving Theorem 5.5.

Lemma 5.7 Assume $n = 2$. For any column monotonic algorithm A , for $0 \leq k \leq R$, it holds that

$$G_A(k) \leq 2^R + \sum_{j=1}^k 2^{2R-j}.$$

Proof Our proof is subdivided in two claims.

Claim 1 Let A^1 be a column monotonic weak Vickrey algorithm. Then A^1 is stochastically dominated by a column monotonic weak Vickrey algorithm A^2 with the property that for every instance v the cell $(r, 2)$ may be queried at a history h in which cell $(r, 1)$ is not opened only if it is known that bidder 2 has the lowest valuation (i.e., $\bar{v}_2(h) \leq \underline{v}_1(h)$).

Proof of Claim 1 Observe that the algorithm A that is obtained from A^1 by consecutively opening all the cells of a bidder i with the lowest valuation as soon as $\bar{v}_i(h) \leq \underline{v}_{-i}(h)$ is Pareto preferred to A^1 . Consider an arbitrary instance v and apply algorithm A . Suppose there is a history h in which A queries a cell $(r, 2)$ before cell $(r, 1)$ while $\bar{v}_2(h) > \underline{v}_1(h)$. Consider a minimal h where such is the case. Because A is column monotonic, all cells $(k, 1)$ and $(k, 2)$ are opened for $k < r$ and no other cells are opened at h . Since $\bar{v}_2(h) > \underline{v}_1(h)$, either $v_{k1} = v_{k2}$ for all $k < r$, or $\bar{v}_1(h) \leq \underline{v}_2(h)$. The latter case can be excluded since then A would open all the remaining cells of player 1 by construction of A . To deal with the former case, we construct an algorithm A' that is identical to A except for instances w for which $w_{k1} = w_{k2} = v_{k1} = v_{k2}$, $1 \leq k \leq r - 1$. In this case the algorithm A' reverses the roles of bidders 1 and 2 when compared to A . By a symmetry argument it is equivalent to A . Repeating this argument leads in a finite number of iterations to the desired algorithm A^2 . \square

Claim 2 Let A be a column monotonic weak Vickrey algorithm with the extra property that at history h the cell $(r, 2)$ may only be queried before the cell $(r, 1)$ if $\bar{v}_2(h) \leq \underline{v}_1(h)$. Then

$$G_A(k) \leq G_{B^*}(k), \quad k \in \{0\} \cup \mathbb{N}.$$

Proof of Claim 2 Without loss of generality, we further assume that A opens all cells of player i consecutively as soon as $\bar{v}_i(h) \leq \underline{v}_{-i}(h)$. Let $\tilde{r}_A : V \rightarrow \{0\} \cup \mathbb{N}$ be the function that assigns to each instance $v \in V$ the number of rows that algorithm A opens in v until it opens a row r such that $v_{r1} \neq v_{r2}$. If such a row is never opened, $\tilde{r}_A(v)$ equals the number of rows that algorithm A opens before it terminates. Let $\tilde{G}_A : \{0\} \cup \mathbb{N} \rightarrow \{0\} \cup \mathbb{N}$ be the function such that $\tilde{G}_A(k)$ equals the number of instances in V for which $\tilde{r}_A(v) \leq k$. Since obviously $\tilde{r}_A(k) \leq r_A(k)$, it holds that $\tilde{G}_A(k) \geq G_A(k)$. \square

Consider the case $k = 0$. It is evident that A can only terminate without opening any row if $v_1 = 0$, which is the case for 2^R instances v , so $\tilde{G}_A(0) \leq 2^R$.

Now note that, for $1 \leq k \leq R$, $\tilde{G}_A(k) - \tilde{G}_A(k - 1)$ is the number of instances for which algorithm A terminates after opening exactly k rows. We partition the set of

such instances for which A terminates after opening exactly k rows into the sets of instances $N_A(k)$ and $S_A(k)$, where for the instances in $N_A(k)$ additionally it has not been decided whether there is a unique highest valuation when A terminates, and for the instances in $S_A(k)$ it is known there is a unique winner when A terminates. The number of elements of $N_A(k)$ is denoted by $n_A(k)$, and $s_A(k)$ denotes the number of elements of $S_A(k)$. Note that $n_A(0) = \tilde{G}_A(0) \leq 2^R$ and $s_A(0) = 0$.

Since A is a column monotonic weak Vickrey algorithm, and always continues with the first column when no winner is known yet, an instance in $N_A(k)$ necessarily satisfies $v_{r1} = v_{r2}$ for $1 \leq r \leq k$ and $v_{r1} = 0$ for $r > k$. Moreover, an instance in $N_A(k)$ does not belong to $N_A(j)$ for $1 \leq j \leq k-1$. This gives us an upper bound on $n_A(k)$ as follows. Note that there are 2^R instances with $v_{r1} = v_{r2}$ for $1 \leq r \leq k$ and $v_{r1} = 0$ for $r > k$. Furthermore, out of the instances that satisfy $v_{r1} = v_{r2}$ for $1 \leq r \leq k$ and $v_{r1} = 0$ for $r > k$, $2^{j-k}n_A(j)$ instances are also an element of $N_A(j)$.⁹ Hence,

$$n_A(k) \leq 2^R - \sum_{j=0}^{k-1} 2^{j-k} n_A(j). \quad (5.1)$$

Next, note that an instance v belongs to $S_A(k)$ if and only if v does not belong to $\bigcup_{j=0}^{k-1} (N_A(j) \cup S_A(j))$, and moreover $v_{k1} \neq v_{k2}$. Such is the case for exactly half of the instances in $V \setminus \bigcup_{j=0}^{k-1} (N_A(j) \cup S_A(j))$. We find that

$$s_A(k) = 2^{-1}(2^{2R} - \tilde{G}_A(k-1)).$$

We now show by induction that the following formula holds.

$$\tilde{G}_A(k) = \sum_{j=0}^k 2^{j-k} n_A(j) + \sum_{j=1}^k 2^{2R-j}.$$

It has already been observed that $\tilde{G}_A(0) = n_A(0)$. Suppose the formula holds for all $k \leq k'$. We show it to be true for $k = k' + 1$. We have

$$\begin{aligned} \tilde{G}_A(k' + 1) &= \tilde{G}_A(k') + n_A(k' + 1) + s_A(k' + 1) \\ &= \tilde{G}_A(k') + n_A(k' + 1) + 2^{-1}(2^{2R} - \tilde{G}_A(k')) \\ &= 2^{-1}\tilde{G}_A(k') + n_A(k' + 1) + 2^{2R-1} \\ &= 2^{-1} \sum_{j=0}^{k'} 2^{j-k'} n_A(j) + n_A(k' + 1) + 2^{-1} \sum_{j=1}^{k'} 2^{2R-j} + 2^{2R-1} \\ &= \sum_{j=0}^{k'+1} 2^{j-k'-1} n_A(j) + \sum_{j=1}^{k'+1} 2^{2R-j}, \end{aligned}$$

⁹For each instance in $N_A(j)$, one can construct 2^{k-j} different instances in $N_A(k)$ by changing the values of $v_{j+1,2}, \dots, v_{k2}$. Only one of these instances satisfies $v_{r1} = v_{r2}$ for $1 \leq r \leq k$, namely when we choose $v_{j+1,2} = \dots = v_{k2} = 0$.

where the fourth equality uses the induction hypothesis. Further, using (5.1) we see that

$$\sum_{j=0}^k 2^{j-k} n_A(j) \leq 2^R - \sum_{j=0}^{k-1} 2^{j-k} n_A(j) + \sum_{j=0}^{k-1} 2^{j-k} n_A(j) = 2^R.$$

It follows that

$$G_A(k) \leq \tilde{G}_A(k) \leq 2^R + \sum_{j=1}^k 2^{2R-j} = G_{B^*}(k)$$

for $0 \leq k \leq R-1$, thereby completing the proof. \square

6 More Bidders and Winner Determination

The conjecture that a suitably defined variation of the bisection algorithm is a stochastically dominant Vickrey algorithm is deceptively simple and plausible. So far we have only been able to show this entirely reasonable conjecture for the case of two bidders. For three or more bidders the problem quickly becomes more complicated than one might initially think. In this section we try to give a few examples that highlight the reasons why.

Example 6.1 Our first example concerns the definition of a plausible candidate for a stochastically optimal Vickrey algorithm in the case $n \geq 3$. Since the bisection algorithm is not proper when the number of bidders exceeds two, it is not a candidate for a stochastically optimal Vickrey algorithm for $n \geq 3$. Unfortunately it is also not difficult to show that the modified bisection algorithm, which is a proper algorithm for any number of bidders, is not stochastically optimal. Consider the instance

$$v = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

The bisection auction would open seven cells, namely (1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 2), and (3, 3). Of these, the modified bisection algorithm would not open cell (3, 3), since after opening cell (3, 2) the second-highest valuation is known. Still better would be an adjustment of the bisection algorithm, where once a unique winning bidder is found, a systematic search is made among the remaining active bidders whether one of them has only ones in the remaining cells. Applied to the example in this paragraph, this would require opening first the cells (1, 1), (1, 2), and (1, 3), and next continuing with opening cells (2, 2) and (3, 2) of bidder 2. This adjusted bisection algorithm would terminate after five steps only. An appropriately defined adjusted bisection algorithm, in particular detailing the systematic search for ones after a unique winner has been found, is a candidate for a stochastically optimal Vickrey algorithm.

Another indication for the complexity of the case with n bidders follows from the consideration of the winner determination problem. The problem is to determine the

set of bidders with the highest valuation for an instance $v \in V$. This problem is related to sorting in computer science. An algorithm that solves this problem is called a WD algorithm. A plausible candidate for a stochastically dominant WD algorithm is the following algorithm.

Step (0). Set $W_0 = N$. Set $k = 1$. Go to Step (k).

Step (k). If $k = R + 1$, STOP. If $k \leq R$ and $|W_{k-1}| = 1$, STOP. Otherwise, open all cells (k, i) in row k with $i \in W_{k-1}$. Let W_k be the set of bidders in W_{k-1} with a 1 on the k th row if there is at least one such bidder, otherwise set $W_k = W_{k-1}$. Set $k = k + 1$. Go to Step (k).

We call this algorithm the WD bisection algorithm, and denote it by B^{WD} . Again we can show that, for $n = 2$, B^{WD} is a stochastically dominant WD algorithm. We can even show the following stronger statement for $n = 2$. Consider the problem where, given an instance $v \in V$, one has to find a row of v that is a single, or show that the instance v does not have a single. An algorithm that solves this problem is called an S algorithm. Notice that every WD algorithm is automatically an S algorithm.

Theorem 6.2 *Assume $n = 2$. Then the algorithm B^{WD} is a stochastically dominant S algorithm. Consequently, it is also a stochastically dominant WD algorithm.*

Proof Observe that the proofs of Lemma 4.7 and Theorem 4.8 also apply in this case. \square

However, for the case $n = 3$, we cannot use the same proof since the definitions of a single and a double are specific to the case $n = 2$. Even for $n = 3$ and $R = 2$, we can only prove a somewhat weaker claim.

Example 6.3 Assume $n = 3$ and $R = 2$. Then there is no WD algorithm A that strictly stochastically dominates the algorithm B^{WD} . This can be seen as follows. Let A be a WD algorithm that strictly stochastically dominates B^{WD} . There are in total $2^6 = 32$ instances. Of these, B^{WD} solves 24 instances in 3 steps. These are the instances where the first row is $(1, 0, 0)$, $(0, 1, 0)$, or $(0, 0, 1)$. In order to match this performance, A has to open the first row, unless this row starts with $(1, 1, *)$, in which case it opens at least the first two cells $(1, 1)$ and $(1, 2)$. In case A opens the first row, from then on it matches exactly the performance of A . So, the only possible gain can be obtained by opening cell $(2, 3)$ before opening cell $(1, 3)$ in case cells $(1, 1)$ and $(1, 2)$ were opened and contained a one.

So, we concentrate on those instances whose first row is of the form $(1, 1, *)$. Clearly cells $(2, 1)$ and $(2, 2)$ have to be opened anyway, no matter what happens. This implies that A does not need to open cell $(1, 3)$ in at most 8 cases. However, for the instances that start with $(1, 1, *)$, B^{WD} does not open cell $(2, 3)$ in exactly 8 cases, namely the ones whose first row is $(1, 1, 0)$. Hence, algorithm A cannot strictly stochastically dominate algorithm B^{WD} .

The relatively weak claim stated below has the status of a conjecture.

Conjecture 6.4 Assume $n \geq 3$. There does not exist a WD algorithm that strictly stochastically dominates the WD algorithm B^{WD} .

Notice that stochastic dominance is not a total ordering. Therefore, even when we can prove Conjecture 6.4, that does not imply that B^{WD} does itself stochastically dominate all other algorithms. Thus, a full proof of the stochastic dominance of B^{WD} for $n \geq 3$ remains elusive.

Example 6.3 identifies an important source of complexity. The algorithms for which the proof becomes involved are those that open cell $(2, 3)$ before cell $(1, 3)$. Even though such non-monotonic behavior does not seem to be advantageous, so far we did not find an argument that it cannot be stochastically dominant.

Thus, the main problem seems to be the relative complexity of the comparison of different instances when at least one of the algorithms is not column monotonic. To emphasize the added complexity arising from algorithms that are not column monotonic, we finally present the following theorem, whose proof is straightforward.

Theorem 6.5 Let $n \geq 3$. The WD algorithm B^{WD} is a Pareto best column monotonic WD algorithm.

Proof Let A be a column monotonic WD algorithm. Let $v \in V$ be an instance, and assume that B^{WD} opens cell (r, i) for this instance.

By definition of B^{WD} , there is a bidder $j \neq i$ such that $v_{kj} = v_{ki}$ for all $k \in \{1, \dots, r-1\}$. Moreover, we have $\sum_{k=1}^{r-1} v_{ki} 2^{R-k} \geq \sum_{k=1}^{r-1} v_{kj} 2^{R-k}$ for all $j \in N \setminus \{i\}$. Now suppose that algorithm A does not open cell (r, i) for this instance. Then, because A is column monotonic, A does also not open any of the cells (r', i) for $r' > r$. Then however, A is not able to solve the winner determination problem between bidder i and any bidder j for which $\sum_{k=1}^{r-1} v_{ki} 2^{R-k} = \sum_{k=1}^{r-1} v_{kj} 2^{R-k}$. So either A never terminates or A opens cell (r, i) . In both cases, $\ell_{B^{\text{WD}}}(v) \leq \ell_A(v)$, which completes the proof. \square

7 Conclusion

In this paper we evaluated the performance of auctions from an algorithmic point of view. In particular, we consider single item auctions with private information and analyze the algorithmic performance of Vickrey-Clarke-Groves mechanisms. These algorithms have superior strategic properties, and in particular give bidders incentives to reveal their true valuation. We introduce a number of such mechanisms, called Vickrey algorithms, including the direct revelation and modified direct revelation algorithms, as well as the bisection and modified bisection algorithms. Vickrey algorithms that perform well are attractive, as they lead to a fast allocation of the object and allow for a limited revelation of the valuations of the bidders.

We consider three possible performance criteria for Vickrey algorithms. The most demanding criterion is the one of being Pareto best, and requires that an algorithm terminates in at most as many steps as any other algorithm, irrespective of the particular instance of the problem under consideration. We show that Pareto best Vickrey

algorithms do not exist. Next we turn to worst-case analysis. Since all acyclic Vickrey algorithms have identical worst-case performance, this criterion is not very helpful to distinguish between algorithms.

We therefore advocate an optimality concept that is equivalent to first-order stochastic dominance. Using this concept, an algorithm stochastically dominates another one if, for any k , the number of instances it can solve in at most k steps is greater than or equal to the number of instances the other algorithm can solve in at most k steps. We show the bisection algorithm to be stochastically dominant for the case with two bidders.

We also study the class of weak Vickrey algorithms, where instead of identifying the identity of all bidders with the highest valuation, it suffices to identify at least one such bidder. To solve this task we introduce the One-Search algorithm and show that it is a stochastically dominant column monotonic weak Vickrey algorithm for the case with two bidders. For the weak Vickrey problem, the One-Search algorithm strictly stochastically dominates the bisection algorithm.

Another related problem is the winner determination problem. The adaptation of the bisection algorithm that is suitable to solve this problem is the WD bisection algorithm. For the case with two bidders, the WD bisection algorithm is a stochastically dominant winner determination algorithm. For the case with n bidders it is a Pareto best column monotonic winner determination algorithm.

Acknowledgements We thank Yu-Hsiang Huang for correcting an error in an earlier draft of this paper. We are also grateful for the comments of two anonymous referees that helped us to improve the exposition of the paper.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Arin, J., Kuipers, J., Vermeulen, D.: An axiomatic approach to egalitarianism in TU-games. *Int. J. Game Theory* **37**, 565–580 (2008)
2. Blumrosen, L., Nisan, N.: Auctions with severely bounded communication. In: *Proceedings of 43th Annual Symposium on Foundations of Computer Science (FOCS 02)*, Vancouver, Canada, pp. 406–416 (2002)
3. Blumrosen, L., Nisan, N., Segal, I.: Multi-bidder and multi-round auctions with severely bounded communication. In: *Proceedings of the 11th Annual European Symposium on Algorithms (ESA 03)*. Budapest, Hungary, 2003
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: *Introduction to Algorithms*. MIT Press, Cambridge (1990)
5. David, E., Rogers, A., Schiff, J., Kraus, S., Jennings, N.R.: Optimal design of English auctions with discrete bid levels. In: *Proceedings of ACM Conference on Electronic Commerce (EC'05)*, Vancouver, Canada, pp. 98–107 (2005)
6. Dutta, B., Ray, D.: A concept of egalitarianism under participation constraints. *Econometrica* **57**, 615–630 (1989)
7. Fishburn, P.C.: Continua of stochastic dominance relations for bounded probability distributions. *J. Math. Econ.* **3**, 295–311 (1976)
8. Fishburn, P.C.: Non-cooperative stochastic dominance games. *Int. J. Game Theory* **7**, 51–61 (1978)
9. Fujishima, Y., McAdams, D., Shoham, Y.: Speeding up ascending-bid auctions. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 554–563 (1999)

10. Green, J., Laffont, J.-J.: Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica* **45**, 427–438 (1977)
11. Grigorieva, E., Herings, P.J.J., Müller, R., Vermeulen, D.: The communication complexity of private value single item auctions. *Oper. Res. Lett.* **34**, 491–498 (2006)
12. Grigorieva, E., Herings, P.J.J., Müller, R., Vermeulen, D.: The private value single item bisection auction. *Econ. Theory* **30**, 107–118 (2007)
13. Heydenreich, B., Müller, R., Uetz, M., Vohra, R.: Characterization of revenue equivalence. *Econometrica* **77**, 307–316 (2009)
14. Hiller, B., Vredeveld, T.: Probabilistic analysis of online bin coloring algorithms via stochastic comparison. METEOR Research Memorandum 08/06, Maastricht University, Maastricht (2008)
15. Holmström, B.: Groves schemes on restricted domains. *Econometrica* **47**, 1137–1144 (1979)
16. Knuth, D.E.: The Art of Computer Programming, vol. 1: Fundamental Algorithms. Addison-Wesley, Reading (1997)
17. Kushilevitz, E., Nisan, N.: Communication Complexity. Cambridge University Press, Cambridge (1997)
18. Levy, H.: Stochastic Dominance. Kluwer Academic, Norwell (1998)
19. Noyan, N., Ruszczyński, A.: Valid inequalities and restrictions for stochastic programming problems with first order stochastic dominance constraints. *Math. Program.* **114**, 249–275 (2008)
20. Patrascua, M., Tarnit, C.E.: On dynamic bit-probe complexity. *Theor. Comput. Sci.* **380**, 127–142 (2007)
21. Schulteis, T., Perea, A., Peters, H., Vermeulen, D.: Stochastic dominance equilibria in two-person non-cooperative games. *Int. J. Game Theory* **34**, 457–473 (2006)