On the fastest vickrey algorithm

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

On The Fastest Vickrey Algorithm

METE**R**

On The Fastest Vickrey Algorithm

ELENA GRIGORIEVA [*]   P. JEAN-JACQUES HERINGS [†]   RUDOLF MÜLLER [‡]
DRIES VERMEULEN [§]

April 13, 2007

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 corresponding to worst case analysis, and a third criterion related to first-order stochastic dominance. We show that Pareto optimal 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 the bisection auction to be optimal according to the third criterion. The bisection auction is therefore optimal in a very strong sense.

KEYWORDS: Single item auctions, Vickrey-Clarke-Groves implementation, algorithms, performance analysis.

JEL CODES: C72, D44.

1 Introduction

Recently there has been quite some interest in designing sealed-bid auctions with limited communication (Blumrosen and Nisan (2002), Blumrosen, Nisan, and Segal (2003)) and iterative auctions with few rounds (David, Rogers, Schiff, Kraus, and Jennings (2005), Fujishima, McAdams, and Shoham (1999)). 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 Grigorieva, Herings, Müller, and Vermeulen (2007) 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. In contrast to these auctions it is however 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. The superiority of the bisection auction has been made precise in Grigorieva, Herings, Müller, and Vermeulen (2006), where we give precise 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 (1977) and Holmström (1979) 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 find this information. We introduce a number of examples of Vickrey algorithms, including the sealed bid and modified sealed bid 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 $n \times R$ matrix of zeroes and ones. And a Vickrey algorithm is deemed to be fast when it needs to query only a limited number of entries (boxes) 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 sealed bid algorithm, we show that Pareto optimal 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. Knuth (1997)). For the problem at hand, worst case analysis hardly makes any 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 is preferred to algorithm A^2 if for every k the number of realizations of valuations on which algorithm A^1 finds the Vickrey outcome by performing at most k queries is not less than the same number in algorithm A^2. For the case of two bidders we show that the bisection algorithm is preferred to any other algorithm that determines the Vickrey outcome. We discuss the complications that arise in extending this result to more than two bidders. Our counting of comparisons 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 running time of sorting algorithms (see, e.g., Chapter 9 of Cormen, Leisserson and Rivest (1990)).

The paper is organized as follows. In Section 2 we give a formal definition of a Vickrey algorithm and we give a number of examples of such algorithms. In Section 3 we introduce some performance criteria and study how particular algorithms perform. We show that Pareto optimal algorithm do not exist, and argue that worst-case analysis is not very helpful. Section 4 introduces the performance criterion that we advocate and show that for the case of two bidders the bisection algorithm is the optimal Vickrey algorithm. Section 5 concludes.

2 Auctions as Algorithms

Suppose there is an auctioneer who wants to sell a single indivisible item to an individual in a set $N = \{1, \ldots, n\}$ of bidders, where $n \geq 2$. Each bidder i has a valuation for the item that is given in a binary encoding of length $R \geq 1$. Thus, the valuation of bidder i is a vector $v_i = (v_{ri})$, $r = 1, \ldots, R$, where each element v_{ri} is a binary digit, i.e. 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 valuation of all bidders is therefore a matrix $v = (v_i)_{i \in N}$, where column i corresponds to the valuation of bidder i. In the following we will also refer to a cell in this matrix as a box. 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.

3

If the auctioneer wants to allocate the object in an incentive compatible and efficient way, he should use a Vickrey-Clarke-Groves mechanism. To implement this mechanism, the auctioneer needs to find a bidder with the highest valuation, and makes him pay a price equal to the second-highest valuation. 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 second-highest column [1]. 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 contained in a box of the matrix v. The well-known 2nd price, sealed bid auction [2] would ask for the values contained in all the boxes of v. 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 $h = (b^1, \ldots, b^k)$, where $b^j \in \{0,1\}$, $j = 1, \ldots, k$. The length of the 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 $C = \{1, \ldots, R\} \times N$ denotes the set of matrix coordinates, in this paper also referred to as entries or boxes.

DEFINITION 2.1 Consider a triplet $A = (H, \sigma, \varphi)$, where H is a collection of histories, and $\sigma \colon H \to \{0,1\}$ is the *stopping function*. The set of non-terminal histories is $H^0 = \{h \in H \mid \sigma(h) = 0\}$. The function $\varphi \colon H^0 \to C$ 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 box $\varphi(h^0)$ of v, which generates the history $h^1 = (v_{\varphi(h^0)})$, where $v_{\varphi(h^0)}$ is the value found in box $\varphi(h^0)$. If $\sigma(h^1) = 1$, the algorithm stops. Otherwise, the algorithm performs another query and opens box $\varphi(h^1)$, 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 box $\varphi(h^k)$, which generates the history $h^{k+1} = (h^k, v_{\varphi(h^k)})$, and so forth.

---

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

[2] This 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".

EXAMPLE 2.2 The algorithm corresponding to the 2nd price sealed bid auction, where first bidder one is queried for its valuation by asking for increasing values of r the value of box $(r, 1)$, next the second bidder is queried, and so on, and so forth, would be defined a follows. The set of histories H consists of all sequences with less than or equal to Rn binary digits. For $h \in H$, $\sigma(h) = 0$ if $\ell(h) \leq Rn - 1$ and $\sigma(h) = 1$ if $\ell(h) = Rn$. For $h \in H^0$, $\varphi(h) = (r, i)$, where i is the largest integer less than or equal to $(\ell(h)/R) + 1$, and $r = \ell(h) + 1 - (i - 1)R$.

DEFINITION 2.3 The *sealed bid algorithm* $A^s = (H^s, \sigma^s, \varphi^s)$ is the algorithm defined in Example 2.2.

The notation of a history only includes the announced binary digits, but not the box that contained them. This is without loss of generality, since the algorithm itself can be used to determine the queried box. 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.

With any algorithm we can associate as follows a rooted binary tree with arcs directed from the root to the endnodes. Nodes of this tree correspond to histories in H, at which the query $\varphi(h)$ is posed, arcs correspond to answers to the queries $\varphi(h)$. The root of the tree is $h^0 = \emptyset$. A sequence of nodes on a path from the root to any node of the tree is a realization of algorithm A. Any node h for which the stopping criterion $\sigma(h)$ is equal to 0 is followed by two arcs. Any node h for which $\sigma(h)$ is equal to 1 is an endnode. The length of an endnode e is equal to the number of nodes, not counting the root, on the path from the root to h. Alternatively, $\ell(e)$ equals the number of arcs on the path from the root to e. This length indicates how many queries are performed before algorithm A ends for an instance v for which the algorithm goes along this path. The set of instances in V for which algorithm A goes along this path and ends in node e has cardinality $2^{Rn-\ell(e)}$, since the number of unopened boxes when algorithm A stops is equal to $Rn - \ell(e)$.

An algorithm is a *Vickrey algorithm* if it does not stop before the identity of all bidders with the highest valuation and the exact value of the second-highest valuation are found. 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 boxes of bidder i, and the maximal valuation of bidder i, $\bar{v}^i(h)$, by assigning a 1 to all unopened boxes 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

1. there is a bidder $i' \in N$ such that $\underline{v}^{i'}(h) = \bar{v}^{i'}(h)$,

2. there is at least one bidder $i \neq i'$ such that $\underline{v}^i(h) \geq \bar{v}^{i'}(h)$, and

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

Notice that the requirement $\underline{v}^{i'}(h) = \bar{v}^{i'}(h)$ is equivalent to the requirement that all boxes of bidder i have been opened.

DEFINITION 2.4 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.

We 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.

EXAMPLE 2.5 The sealed bid algorithm of Example 2.2 qualifies as a Vickrey algorithm. The sealed bid algorithm of Example 2.2 would continue to make queries even after all the information needed has been retrieved. A superior alternative is therefore the *modified sealed bid 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. We denote the modified sealed bid algorithm by A^{ms}.

DEFINITION 2.6 The *modified sealed bid algorithm* $A^{ms} = (H^{ms}, \sigma^{ms}, \varphi^{ms})$ is the algorithm defined in Example 2.5.

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 sealed bid algorithm is not proper, whereas the modified sealed bid 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. Therefore an acyclic algorithm always ends within Rn performed queries.

DEFINITION 2.7 An algorithm $A = (H, \sigma, \varphi)$ is *cyclic* if there are histories $g, h \in H$ with $g < h$ such that $\varphi(g) = \varphi(h)$. An algorithm $A = (H, \sigma, \varphi)$ is *acyclic* if for all histories $g, h \in H$ with $g < h$ it holds that $\varphi(g) \neq \varphi(h)$.

Both the sealed bid algorithm and the modified sealed bid algorithm are acyclic.

Another Vickrey algorithm is the algorithm corresponding to the bisection auction proposed in Grigorieva et al. (2007). 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 boxes of the matrix v in R steps. In step r the algorithm B opens boxes in the rth row of v. Inside a step the boxes 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 boxes 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 players 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 box (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 boxes 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 (Grigorieva et al. (2007)).

DEFINITION 2.8 The *bisection algorithm* $B = (H^B, \sigma^B, \varphi^B)$ is the algorithm defined in previous paragraphs.

EXAMPLE 2.9 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 boxes $(1,1), (1,2), (2,1), (2,2)$, 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 sealed bid algorithm and the modified sealed bid algorithm.

We state the following result without proof.

THEOREM 2.10 The bisection algorithm B is acyclic and, for $n = 2$, proper.

The following example shows that for $n = 3$, the bisection algorithm need not be proper.

EXAMPLE 2.11 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 boxes $(1,1), (1,2), (1,3), (2,2)$, 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. For $n \geq 3$, we may therefore consider the modified bisection 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.

DEFINITION 2.12 The *modified bisection algorithm* $B^m = (H^{mB}, \sigma^{mB}, \varphi^{mB})$ is the algorithm defined in Example 2.11.

According to Theorem 2.10, the bisection algorithm is proper for $n = 2$. The bisection algorithm and the modified bisection algorithm do therefore coincide for $n = 2$.

The number of different algorithms is staggering. Suppose we restrict ourselves to acyclic algorithms. There are Rn possibilities for the first box to be opened. An acyclic algorithm proceeds conditional on the

information found, chooses one box out of $Rn - 1$ boxes to be opened next, and so on. If we ignore the stopping criterion, and consider only algorithms that proceed until the last box is opened, then we obtain $(Rn)!$ different acyclic algorithms.

3 Performance Criteria for Algorithms

Given an algorithm A, $\ell_A : V \to \mathbb{N} \cup \{\infty\}$ 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 $v \in V$, $\ell_{A^1}(v) \le \ell_{A^2}(v)$. 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 $v \in V$, $\ell_{A^1}(v) \le \ell_{A^2}(v)$. 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. The algorithms A^1 and A^2 are *Pareto indifferent* if A^1 is Pareto preferred to A^2 and A^2 is Pareto preferred to A^1 (in other words, when $\ell_{A^1}(v) = \ell_{A^2}(v)$ for all $v \in V$).

DEFINITION 3.2 A Vickrey algorithm is *Pareto optimal* if it is Pareto preferred to any other Vickrey algorithm.

Not all Vickrey algorithms are candidates for being Pareto optimal. It is rather obvious that neither Vickrey algorithms that are not proper nor cyclic Vickrey algorithms are Pareto optimal.

THEOREM 3.3 *For any Vickrey algorithm $A = (H, \sigma, \varphi)$ that is not proper, there is a Vickrey algorithm $A' = (H', \sigma', \varphi')$ that is strictly Pareto preferred to A^1.*

PROOF. We construct an algorithm A' with the desired properties. Let H' be the subset of histories $h \in H$ that does not have subhistories $g < h$ such that after history g the identity of all bidders with the highest valuation and the exact value of the second-highest valuation are found. When h is a maximal history in H', then define $\sigma'(h) = 1$, otherwise $\sigma'(h) = 0$. For all $h \in H'$ with $\sigma'(h) = 0$ define $\varphi'(h) = \varphi(h)$. Then A' coincides with A until the desired information has been found, and stops thereafter. It is strictly Pareto preferred to A. Q.E.D.

THEOREM 3.4: *For any cyclic Vickrey algorithm A there is a Vickrey algorithm A' that is strictly Pareto preferred to A.*

PROOF. Take a cyclic Vickrey algorithm $A = (H, \sigma, \varphi)$. We construct an algorithm $A'(H', \sigma', \varphi')$ such that for all $v \in V$, $\ell_{A'}(v) \le \ell_A(v)$ with strict inequality for some v. Since A is cyclic, there is an instance $v \in V$ generating histories $a, b \in H$ such that $a < b$ and $\varphi(a) = \varphi(b)$.

Suppose that there is such an instance $v \in V$ for which $\ell_A(v)$ is finite, so there are histories $a, b, c \in H$ such that $a < b < c$, $\varphi(a) = \varphi(b)$, $\ell(c) \in \mathbb{N}$, and $\sigma(c) = 1$. We construct algorithm A' in the following

way. A' performs the same queries as A does, except after history b. After history b it skips query $\varphi(b)$ and continues with the same queries as A after history $(b, v_{\varphi(b)})$. By skipping query $\varphi(b)$ algorithm A' does not lose any information and is therefore able to stop whenever A stops. For all other histories A' is identical to A. It follows that for all instances v that generate history b, $\ell(c) - 1 = \ell_{A'}(v) < \ell_A(v) = \ell(c)$, whereas for all other instances $\ell_{A'}(v) = \ell_A(v)$. Hence A' is strictly Pareto preferred to A.

Suppose that for all instances $v \in V$ that generate histories a, b with $a < b$ and $\varphi(a) = \varphi(b)$, it holds that $\ell_A(v) = \infty$. Construct algorithm A' as follows. For all histories $h \in H$ that do not contain a subhistory g with $g < h$ and $\varphi(h) = \varphi(g)$, $\varphi'(h) = \varphi(h)$. For minimal histories $h \in H$ that do contain a subhistory g with $g < h$ and $\varphi(h) = \varphi(g)$, $\varphi'(h)$ is the lexicographically minimal unopened box, and A' continues by opening unopened boxes in lexicographical order till all boxes are opened. So, for all instances $v \in V$ for which A terminates in finite steps, $\ell_A(v) = \ell_{A'}(v)$, and for all $v \in V$ for which $\ell_A(v) = \infty$, $\ell_{A'}(v)$ is finite. Hence A' is strictly Pareto preferred to A. \hfill Q.E.D.

Thus both cyclic and non-proper Vickrey algorithm are not Pareto optimal. Unfortunately, the Pareto criterion has only limited further use beyond these straightforward observations. There do exist rather silly algorithms that accidently perform very well for particular instances. An example is the reversed modified sealed bid algorithm, that starts by opening box (R, n), next box $(R - 1, n)$, and continues in this way until all boxes 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 2.9 with $n = 2$, $R = 3$, and

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

For this instance, the length of the sealed bid algorithm and the modified sealed bid algorithm is six, the length of the bisection algorithm and the modified bisection algorithm is five, and the length of the reversed modified sealed bid algorithm is four. Indeed, this algorithm opens box $(3, 1)$ 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.

The following result shows that for $R \geq 2$ Pareto optimal Vickrey algorithms do not exist [3].

THEOREM 3.5 *Assume $n \geq 2$ and $R \geq 2$. Then there is no Pareto optimal Vickrey algorithm.*

We prove Theorem 3.5 by means of Lemmas 3.6 and 3.7 that consider the case $n = 2, R \geq 2$, and the case $n \geq 3, R \geq 2$, respectively.

LEMMA 3.6 *Assume $n = 2$ and $R \geq 2$. Then there is no Pareto optimal Vickrey algorithm.*

---

[3]When $R = 1$ all acyclic Vickrey algorithms are of course Pareto indifferent and take n steps.

PROOF. Based on Theorems 3.3 and 3.4, we can restrict our attention to proper acyclic algorithms when searching for a Pareto optimal Vickrey algorithm. Consider the instance

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

Let A be an algorithm that opens one of the boxes $(r, 1)$, $r \leq R-1$, in the first round. We get $l_A(v^1) > R+1$, because A will have to open box $(R, 1)$ to determine the winner and all boxes $(r, 1)$, $r \leq R$, to determine the second highest valuation. Thus, because there exists a Vickrey algorithm A^1 with $l_{A^1}(v^1) = R + 1$, A is not Pareto optimal. Indeed, A^1 opens all boxes in the second column together with box $(R, 1)$. Next consider the instance

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

By the same reasoning as before, any algorithm A that opens one of the boxes $(r, 2)$, $r \leq R-1$, in the first round cannot be Pareto optimal. Therefore, any Pareto optimal algorithm A has to start with opening either box $(R, 1)$ or $(R, 2)$. W.l.o.g., assume it starts with box $(R, 2)$. Now consider the instance

$$v^3 = \begin{bmatrix} 0 & 1 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

By the same reasoning as before, $l_A(v^3) > R+1$. But there exists a Vickrey algorithm with $l_A(v^3) = R+1$, a contradiction. It follows that Pareto optimal Vickrey algorithms do not exist.   Q.E.D.

LEMMA 3.7 *Assume $n \geq 3$ and $R \geq 2$. Then there is no Pareto optimal Vickrey algorithm.*

PROOF. For $i = 1, \ldots, n$, let the instance v^i be such that it has zeroes everywhere, except in box (R, i), where it has a 1. Any algorithm A has to open all the boxes of bidders not equal to i to decide whether such bidders have the highest valuation or not. When also box (R, i) is opened, the identity of all bidders with the highest valuation, i, and the exact value of the second-highest valuation, 0, is known. A Pareto optimal algorithm terminates after $R(n-1) + 1$ steps. Any algorithm that opens a box (r, i), $r \leq R-1$, $1 \leq i \leq n$, in round 1 will need at least $R(n-1) + 2$ steps for instance v^i as it has to open all boxes in some column $j \neq i$ to determine the second highest valuation, all other boxes in columns other than i in order to assure that this is the second highest valuation, and box (R, i) to determine the winner. A by now familiar argument shows that a Pareto optimal algorithm starts by opening a box in row R.

W.l.o.g., assume that a Pareto optimal A algorithm starts with box $(R, 1)$. Let the instance $\bar{v}^1$ be such that it has a 1 in box $(1, 1)$, a 0 everywhere else. Algorithm A has to open all boxes in some column j

in order to determine the second highest valuation, all other boxes in columns not equal to i in order to verify that this is the second highest valuation, and box $(1,1)$ in order to determine the winner. Thus $l_A(\bar{v}^1) \geq R(n-1) + 2$. However, there exists a Vickrey algorithm $\bar{A}$ with $l_{\bar{A}}(\bar{v}^1) = R(n-1) + 1$, a contradiction. It follows that a Pareto optimal algorithm does not exist. Q.E.D.

The usual approach taken when assessing the performance of an algorithm is to perform a worst-case analysis. This leads to the following definitions.

DEFINITION 3.8 A Vickrey algorithm A^1 is *worst case preferred* to a Vickrey algorithm A^2 if $\max_{v \in V} \ell_{A^1}(v) \leq \max_{v \in V} \ell_{A^2}(v)$. The algorithm A^1 is *strictly worst case preferred* to A^2 when A^1 is worst case preferred to A^2, but A^2 is not worst case preferred to A^1. A^1 and A^2 are *worst case equivalent* when A^1 is worst case preferred to A^2 and A^2 is worst case preferred to A^1.

DEFINITION 3.9 A Vickrey algorithm A^1 is a *worst case optimal Vickrey algorithm* if A^1 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 given number of bidders n and length of valuations R. The connection with the Pareto criterion is straightforward.

THEOREM 3.10 *Let A^1 and A^2 be two Vickrey algorithms. If A^1 is Pareto preferred to A^2, then A^1 is also worst case preferred to A^2.*

Unfortunately, worst case analysis does not give many insights when applied to Vickrey algorithms, as all acyclic Vickrey algorithms are equivalent to each other according to this criterion.

THEOREM 3.11 *Let A^1 and A^2 be two acyclic Vickrey algorithms. Then A^1 is worst case equivalent to A^2.*

PROOF. Any acyclic Vickrey algorithm terminates in less than or equal to Rn steps. 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 boxes. Q.E.D.

4 Optimality of the Bisection Algorithm

Thus, the usual criteria for comparison of algorithms, Pareto optimality and worst-case analysis, are of no use for comparison of Vickrey algorithms. The criterion we advocate to compare (Vickrey) algorithms relies on the concept of the characteristic function of an algorithm.

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

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 is *preferred* to an algorithm A^2 if $F_{A^1}(k) \geq F_{A^2}(k)$ for all $k \in \mathbb{N}$. A^1 is *strictly preferred* to A^2 when A^1 is preferred to A^2, but A^2 is not preferred to A^1.

Notice that an algorithm A^1 is preferred to 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. For the trivial case where $R = 1$, a Vickrey algorithm has to open all the boxes, so $F_A(k) = 0$ for all $k \leq n - 1$. When $R \geq 2$, the most favorable instance is the one where in the first row one box 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 boxes in column i determines the exact value of the second-highest valuation. Any Vickrey algorithm has to open all the boxes 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 an *optimal Vickrey algorithm* when A is preferred to any other Vickrey algorithm.

Definition 4.2 corresponds to the definition of first-order stochastic dominance (see, e.g. Levy (1998)) if 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 is (strictly) preferred to an algorithm A^2 has a (strictly) shorter average running time than the algorithm A^2.

Consider an acyclic Vickrey algorithm A. Since the number of elements in V equals 2^{Rn}, and opening all boxes definitely yields the required information, it holds that $F_A(k) = 2^{Rn}$ for all $k \geq Rn$. When all bidders have identical valuations, a Vickrey algorithm will open all the boxes. It follows that $F_A(Rn) - F_A(Rn-1) \geq 2^R$. Using the tree representation of the algorithm, we can say that

$$F_A(k) = \sum_{j=1}^{k} |E_j| 2^{Rn-j},$$

where E_j is the set of endnodes with length j. Here we use that an acyclic algorithm that terminates after j steps, has $Rn - j$ boxes unopened, which corresponds to 2^{Rn-j} instances.

The next result claims that optimality is weaker than Pareto optimality, but stronger than worst case optimality. The proof is obvious and therefore omitted.

Theorem 4.4 *Let A^1 and A^2 be two algorithms. If A^1 is Pareto preferred over A^2, then A^1 is preferred over A^2. If A^1 is preferred over A^2, then A^1 is worst case preferred over A^2.*

Suppose we have two Vickrey algorithms A^1 and A^2, and suppose that the designer of algorithm A^1 challenges the designer of algorithm A^2 saying: I claim that my algorithm requires less steps than yours on the set V of instances. How could we decide that this is a legitimate claim? We suggest to let the designers play the following game. Each contestant starts with an instance bin that initially contains all instances and a discard bin that is initially empty. In the first round of the game A^2 may choose any instance v from his instance bin. The task of A^1 is to find an instance v' (not necessarily equal to v) in his instance bin on which his algorithm performs at least as good as algorithm A^2 does on instance v, i.e. such that $\ell_{A^1}(v') \leq \ell_{A^2}(v)$. If A^1 is not able to find such an instance, then A^1's claim is considered illegitimate. However, if A^1 succeeds to find such an instance, both contestants discard the instance they have chosen to their respective discard bins, and the game enters the second round that is played using only the instances that are still left in their instance bins. This procedure is repeated until either all remaining instances are worse for A^1 than the chosen instance by A^2, or the instance bins are empty, in which case we declare A^1 to be the winner.

It is fairly easy to find optimal strategies for both designers. Designer A^2 can basically chose any instance from his instance bin while designer A^1 should, when faced with instance v, select an instance v' that has the highest value of ℓ_{A^1} among those instances that are left in his instance bin and for which $\ell_{A^1}(v') \leq \ell_{A^2}(v)$ holds. Notice that following the optimal strategy designer A^1 wins the game if and only if for all k the number of instances for which algorithm A^1 performs at most k queries is not less than the number of instances for which algorithm A^2 performs at most k queries, so if and only if A^1 is preferred over A^2.

We will now show how to use this game to prove that in the case of 2 bidders, i.e. $N = \{1, 2\}$, the bisection algorithm is optimal. We know that the bisection algorithm coincides with the modified bisection algorithm. We show that the bisection algorithm is optimal. 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 $r_A \colon V \to \mathbb{N}$ be the function that assigns to each instance v in V the number of rows that algorithm A opens in v before it stops. Let $G_A \colon \mathbb{N} \to \mathbb{N}$ be the function where $G_A(k)$ equals the number of instances in V for which $r_A(v) \leq k$.

Since a Vickrey algorithm A should find the exact value of the second-highest valuation and thus should open all the boxes 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:

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

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 is preferred to an acyclic Vickrey algorithm A^2, it is enough to show that $G_{A^1}(k) \geq G_{A^2}(k)$ holds 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, since it cannot exclude that both bidders valuations are equal. This contradicts the definition of a Vickrey algorithm.   Q.E.D.

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 opened

$$
\begin{bmatrix}
0 & 0 \\
1 & 1 \\
0 & x
\end{bmatrix}
$$

after five queries. 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 box. That means that the algorithm, before it stops, neither opens all R rows nor a single.

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. We show that, for $k < R$,

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

Since $G_A(k) \leq G_{A'}(k)$, this proves the lemma.

The case $R = 1$ is trivial, so we assume $R \geq 2$. In our proof we will construct a set M_d of instances for which the first opened row by A' has distinct values, and a set M_e of instances for which the first opened row 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 box b_r is opened. Define the instance v' as follows:

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

15

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

$$M_e := M_e \cup \{v\},$$
$$M_d := M_d \cup \{v'\},$$

and otherwise

$$M_e := M_e \cup \{v'\},$$
$$M_d := M_d \cup \{v\}.$$

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 4.5, 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 opened row is a double. At the end of the procedure we have sets M_d and M_e. By Lemma 4.5, $V(2) = M_d$, 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 $k < R$,

$$G_{A'}(k) = |\bigcup_{j=1}^{k} V(j)| = \sum_{j=1}^{k} 2^{2R-j}. \qquad \text{Q.E.D.}$$

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 boxes of the second highest valuation.

THEOREM 4.7 *Assume $n = 2$. The bisection algorithm B is an optimal 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 $|V_k(B)| = 2^{2R-k}$ and for $k < R$,

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

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

$$F_A(k) \leq F_B(k). \qquad \text{Q.E.D.}$$

5 Conclusion

In this paper we evaluate 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 sealed bid and modified sealed bid 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 Pareto optimality, and requires that an algorithm terminated in at most as many steps as any other algorithm, irrespective of the particular instance of the problem under consideration. We show that Pareto optimal 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 related to first-order stochastic dominance. Using this concept, an algorithm is preferred to 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. Although this is a very demanding criterion too, we show the bisection algorithm to be optimal for the case with two bidders.

An interesting and very natural question is to extend this result to an arbitrary number of bidders. Since the bisection algorithm is not proper when the number of bidders exceeds two, it is not a candidate for an optimal Vickrey algorithm. It is not hard to show that also the modified bisection algorithm, which is a proper algorithm for any number of bidders, is not optimal. Suppose $n = 3$, $R = 3$, $v^1 = (1,0,0)$, and $v^2 = v^3 = (0,1,1)$. The bisection auction would open seven boxes. Of these, the modified bisection algorithm would not open box $(3,3)$, since after opening box $(3,2)$ 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 boxes. Applied to the example in this paragraph, this would require opening first the boxes $(1,1)$, $(1,2)$, and $(1,3)$, and next continuing with opening all the boxes of player 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 an optimal Vickrey algorithm.

Acknowledgements

References

Blumrosen, L., and N. Nisan (2002), "Auctions with Severely Bounded Communication," in *Proceedings of 43th Annual Symposium on Foundations of Computer Science (FOCS 02),* Vancouver, Canada, 406–416.

Blumrosen, L, N. Nisan, and I. Segal (2003), "Multi-bidder and Multi-round Auctions with Severely Bounded Communication," in *Proceedings of 11th Annual European Symposium on Algorithms (ESA 03),* Budapest, Hungary.

Cormen, T. H., C. E. Leiserson, R. L. Rivest (1990), *Introduction to Algorithms,* MIT Press, Cambridge.

David, E., A. Rogers, J. Schiff, S. Kraus, and N.R. Jennings (2005), "Optimal Design of English Auctions with Discrete Bid Levels," in *Proceedings of ACM Conference on Electronic Commerce (EC'05),* Vancouver, Canada, 98–107.

Fujishima, Y., D. McAdams, and Y. Shoham (1999), "Speeding Up Ascending-bid Auctions," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence,* 554–563.

Green, J., and J.-J. Laffont (1977), "Characterization of Satisfactory Mechanisms for the Revelation of Preferences for Public Goods," *Econometrica, 45,* 427–438.

Grigorieva, E., P.J.J. Herings, R. Müller, and D. Vermeulen (2006), "The Communication Complexity of Private Value Single Item Auctions," *Operations Research Letters, 34,* 491–498.

Grigorieva, E., P.J.J. Herings, R. Müller, and D. Vermeulen (2007), "The Private Value Single Item Bisection Auction," *Economic Theory, 30,* 107–118.

Holmström, B. (1979), "Groves Schemes on Restricted Domains," *Econometrica, 47,* 1137–1144.

Knuth, D.E. (1997), *The Art of Computer Programming; Volume 1: Fundamental Algorithms,* Addison-Wesley, Massachusetts.

Levy, H. (1998), *Stochastic dominance,* Kluwer Academic Publishers, Norwell, Massachusetts.