Auctions and Corruption: An Analysis of Bid Rigging by a Corrupt Auctioneer[☆]

Yvan Lengwiler[a], Elmar Wolfstetter[b]

[a]*University of Basel, Dept. of Economics (WWZ), Peter Merian-Weg 6, P.O. Box, CH-4002 Basel, Switzerland, email yvan.lengwiler@unibas.ch*
[b]*Humboldt University at Berlin, Institute of Economic Theory I, Spandauer Strasse 1, D-10178 Berlin, Germany, email wolfstetter@googlemail.com*

Abstract

In many auctions, the auctioneer is an agent of the seller. This invites corruption. We analyze a model in which the auctioneer orchestrates bid rigging by inviting a bidder to either lower or raise his bid, whichever is more profitable. The interplay between these two types of corruption gives rise to a complex bidding problem that we tackle with numerical methods. Our results indicate that corruption does not only redistribute surplus away from the seller, but also distorts efficiency. We furthermore explain why both, the auctioneer and bidders, have a vested interest in maintaining corruption.

Keywords: auctions, corruption, procurement, bid rigging.
JEL: D44

1. Introduction

Corruption is generally defined as the "misuse of a position of trust for dishonest gain." In an auction context such a misuse occurs if the players in an auction game collude and twist the rules to their mutual benefit. This may take the form of collusion among bidders who form a bidding ring or a dishonest agent auctioneer who engages in bid rigging in concert with his favored bidder(s).

[☆]We wish to thank Peter Cramton, Dirk Engelmann, Nicolas Sahuguet, and an anonymous referee for their comments, and seminar participants at the Universities of Maryland, Melbourne, Zürich, the Stockholm School of Economics, and Korea University. Financial support was received from the *Deutsche Forschungsgemeinschaft,* SFB Transregio 15, "Governance and Efficiency of Economic Systems."

The present paper analyzes corruption in auctions that involves a dishonest auctioneer. This corruption may be a simple bilateral affair where the dishonest auctioneer rigs bids in favor of one bidder in exchange for a bribe. Or it may involve several bidders who jointly strike a deal with the auctioneer.

Obviously, this kind of corruption can only occur if the seller delegates the sale to an agent auctioneer. Such delegation is widely prevalent either because the seller lacks the expertise to run the auction himself or because the seller is a complex organization. Thereby, it does not matter whether the agent auctioneer is a specialized auction house, an employee, or a government official. What matters alone is the fact that the auctioneer acts independently on behalf of the seller.

Corruption involving a dishonest auctioneer can also not work easily in an open-bid auction simply because it lacks secrecy. However, open auctions are typically hybrids between open and sealed bid auctions since sealed bids are usually permitted and are indeed widely used. A corrupt auctioneer can then use "magic numbers" (empty envelopes) to rig bids even if some bidders participate in the open auction (see Ingraham, 2001).

An early case of corruption in auctions is Goethe's dealing with his publisher Vieweg concerning the publication of his epic poem "Hermann and Dorothea" in the year 1797. Eager to know the true value of his manuscript, Goethe designed a clever scheme. He handed over a sealed note containing his reservation price to his legal Counsel Böttiger. At the same time he asked the publisher to make a bid and send it to Böttiger, promising publication rights if and only if the bid is at or above Goethe's reserve price, in which case he would have to pay Goethe's reserve price.

Obviously, in the absence of corruption, the publisher should have bid his true valuation. On this ground, Moldovanu and Tietzel (1998) credit Goethe for anticipating the Vickrey auction. However, Goethe's legal Counsel was not reliable; indeed, he opened Goethe's envelope, and, maliciously informed the publisher about its content, before he made his bid.[1] Not surprisingly, Vieweg's bid was exactly equal to Goethe's reserve price, and thus Goethe's clever scheme fell prey to corruption.

Today, corruption is a frequently observed and well documented event especially in construction and procurement auctions. The *Chartered Institute of Build-*

[1]The letter from Böttiger to Vieweg has been preserved: "The sealed envelope is here in my office, and you have to tell me now, dear Vieweg, how much you are prepared to offer. I feel whatever a spectator, who is your friend, can feel. Just allow me to [...] say this: you cannot offer less than 1000 Taler." (p. 651 Jensen, 1984, our translation).

ing, a UK professional association of the construction industry, reports that 41% of the respondents of a representative survey have been offered bribes on at least one occasion.[2]

A specific example of a rigged auction is the bidding for the construction of a new metropolitan airport in the Berlin area, which was reopened after investigators found out that *Hochtief AG*, the winner of the auction, was enabled to change its bid after it had illegally acquired the application documents of the rival bidder *IVG*.[3] As another example, in 1996 the authorities of Singapur ruled to exclude *Siemens AG* from all public procurements for a period of five years after they determined that Siemens had bribed Choy Hon Tim, the chief executive of Singapur's public utility corporation *PUB*, in exchange for supplying Siemens with information about rival bids for a major power station construction project.[4]

A flagrant case of bid rigging that perfectly fits our analysis was uncovered a few years ago in Slovenia.[5] In November 1999 the Slovenian government called for bids to construct a tunnel of about one kilometer through the "Trojane hill," which is part of a highway that connects eastern and western Slovenia. *DARS* (the "Motorway Company of the Republic of Slovenia," a state enterprise) was responsible for the tender. It hired *DDC*, a major engineering firm, as an auctioneer. The bids were to contain a single "best price offer" and documentation of the proposed construction. The bids had to be submitted in writing and electronically, and were held in a secure room. A video was taken of the public opening of the sealed bids. The two lowest bidders were *Grassetto*, an Italian enterprise, and *SCT*, a Slovene construction firm. The video showed a yellow diskette containing *SCT*'s electronic bid. At that time, *Grassetto*'s bid was SIT 2845 million Slovenian tolars (around USD 12.5 million), and *SCT*'s bid was SIT 2945 million tolars.

Later, however, the auction committee (run by *DDC*) declared *SCT* the winner as their bid had somehow dropped to SIT 2764 million in the meantime both in the paper and in the electronic version. The diskette that was contained in the documentation and on which this new offer was stored had also changed color: it was no longer yellow, but had turned pink. Despite the various security measures, *SCT*'s bid had obviously been changed after the original bids had been opened.

---

[2]Chartered Institute of Building (CIOB), "Corruption in the UK Construction Industry — Survey 2006" (`http://www.ciob.org.uk/filegrab/CIOBCorruption.pdf?ref=283`).

[3]See *Wall Street Journal*, Aug. 19, 1999.

[4]See *Berliner Zeitung*, Feb. 2, 1996.

[5]See the Slovenian newspaper *Mladina*, "Trojanska disketa," 5. November 2001. A translation can be downloaded from the authors' websites.

SCT was later excluded from the tender. The investigation has been terminated in the meantime, however, because the prosecutor was unable to identify the person who was responsible for exchanging the bids.

Motivated by these examples, this paper analyzes a particular model of bid rigging by a corrupt auctioneer. Its essential features are as follows: (1) It is common knowledge among bidders that the auctioneer is corrupt. (2) The auctioneer minimizes illegal contacts and negotiates only with one bidder. (3) Bid rigging means that the auctioneer either allows the highest bidder to lower his bid to the level of the second highest bid, or allows the second highest bidder to match the highest bid, in exchange for a fixed share of the surplus. (4) The seller does not install rules that are capable of deterring corruption. The reason may be a lack of proper governance, as it is often the case when the seller is not a residual claimant.

2. Relation to the literature

There is a large literature on collusion in auctions that focuses on collusion among bidders, and in which the incentives of the auctioneer are properly aligned with the interest of the seller.[6] There is also a smaller literature that focuses on corruption as defined above, which is an issue only if there is delegation.

That literature views corruption either as a manipulation of the quality assessment in complex bids or as bid rigging. The former was introduced in a seminal paper by Laffont and Tirole (1991), who assume that the auctioneer has some leeway in assessing complex multidimensional bids, and is predisposed to favor a particular bidder. That framework was later adopted by several authors. For example, Celantani and Ganuza (2002) employ it to assess the impact of increased competition on equilibrium corruption. They find, surprisingly, that corruption may increase if the number of competing bidders is increased. More recently, Burguet and Che (2004) extend that framework. They consider a scoring auction, make the assignment of the auctioneer's favorite agent endogenous, and assume that bribery competition occurs at the same time as contract bidding. Their main result is that corruption may entail inefficiency, and that "… the inefficiency cost of bribery is in the same order of magnitude as the agent's [*i.e. auctioneer's*] manipulation capacity" (Burguet and Che, 2004, *p*. 61, emphasis added).

[6]Basic references in this literature are Graham and Marshall (1987); Mailath and Zemsky (1991); McAfee and McMillan (1992). Recently, this literature has focused on case studies (Porter and Zona, 1999; Athey et al., 2008) and on collusion in repeated auctions (Athey et al., 2004; Aoyagi, 2007).

A second branch of that literature considers a particular form of bid rigging, in which the auctioneer grants a "right of first refusal" to a favored bidder. This right gives the favored bidder the option to match the highest bid and win the auction.[7] In a first-price auction, the favored bidder thus effectively plays a second price auction, whereas the other bidders pay their bid if they win. Typically, that literature treats the favored bidder status as predetermined. In that case efficiency is destroyed because the favored bidder may not have the highest valuation and yet exercise his right. Burguet and Perry (2007) and Arozamena and Weinschelbaum (2009) analyze this model.

The attractive feature of that literature is that it can explain how corruption destroys efficiency. Yet, that feature is lost as soon as one makes the selection of the favored bidder endogenous. For instance, in an earlier version of their paper, Burguet and Perry (2007) also consider a variation of their model in which bidders compete for the favored bidder status before the auction by submitting bribes to the auctioneer. This restores efficiency because the strongest bidder offers the highest bribe. Similarly, Koc and Neilson (2008) consider a model in which the right to play a second-price auction is sold for a lump sum bribe before the auction. In that game, only high valuation bidders buy that right, which immediately implies efficiency.

An implausible feature of that approach is that the corrupt auctioneer approaches *all* bidders in order to select the favored one. This entails that the auctioneer exposes himself to an exceedingly high risk of detection and punishment. Every auctioneer who cares about the risk of detection will only propose corruption to the smallest possible number of bidders.

Another way to reduce that risk may be to restrict the size of the bribe, as Compte et al. (2005) assume. In their analysis bribes are offered by all bidders jointly with the submitted bids. These bribes are assumed to be restricted to the size of small gifts. In exchange for this gift, the auctioneer allows one bidder to revise his bid. In equilibrium, all bidders submit the same maximum bribe together with a zero bid, which suggests the interpretation of the role of the corrupt auctioneer as an enforcement device of collusion in the style of the zero bid pooling equilibrium by McAfee and McMillan (1992).

The restriction of corruption to small bribes may be appealing in some applications, for example because small bribes do not expose the involved parties to great risk. Empirically, however, bribes are often very large. In fact, corruption

[7] On the right of first refusal, see Bikhchandani et al. (2005) and Grosskopf and Roth (2009).

often occurs only if bribes are sufficiently large in order to compensate for the risk of detection.

This takes us to the third branch of the literature to which the present paper belongs. Its key feature is that bid rigging is arranged by the auctioneer *after* he has observed all the bids. This allows him to approach only a minimum number of bidders, and select the one bidder whose collaboration delivers the highest profit.

Lengwiler and Wolfstetter (2000) and Menezes and Monteiro (2006) consider a first-price auction where the auctioneer allows the highest bidder to lower his bid in exchange for a bribe that is proportional to the gain from corruption.[8] Essentially, they show that this game is equivalent to a standard auction without corruption in which the price to be paid by the highest bidder is a given convex combination of the two highest bids. That game has a unique monotone symmetric equilibrium which implies efficiency, as one can learn already from Güth and van Damme (1986), Riley (1989), and Güth (1995) who solved this auction game (Güth called it the "λ"-auction).

Their analysis is, however, incomplete, because even if the corrupt auctioneer deals only with one bidder, it may be more profitable for him to let the second highest bidder match the highest bid (we will refer to this as "type II corruption"), as in a right of first refusal arrangement, rather than allowing the highest bidder to lower his bid and match the second highest bid (to which we will refer as "type I corruption"). A rational auctioneer flexibly chooses the alternative that creates the largest surplus, which depends upon the spread between the two highest bids.

Specifically, if that spread is "large," the auctioneer's optimal choice is to propose to the highest bidder to lower his bid (type I), whereas if that spread is "small" and bid shading is significant, he should propose to the second highest bidder to raise his bid (type II). The key feature of our model is that we allow for this endogenous choice of corruption type. Because type II corruption implies a change of the allocation, welfare losses are now possible.

The interplay between the two types of corruption gives rise to a complex bidding problem. We tackle this problem with the help of numerical methods.

3. The model

The owner of a single good has delegated its sale to an agent auctioneer who runs a first-price sealed-bid auction with $n \geq 2$ risk neutral bidders. The agent

[8]Menezes and Monteiro also consider a lump sum bribe, and Lengwiler and Wolfstetter also consider corruption in the second price auction.

auctioneer is corrupt and is able to rig bids after they have been submitted, and this fact is common knowledge among all bidders. Bidders have private values which are denoted by $v_1, \ldots, v_n$. They are independent draws from the continuously differentiable c.d.f. F with support $[0, 1]$ and p.d.f. $f(v) := F'(v)$. F and n are common knowledge among all bidders and the auctioneer.

From the perspective of one bidder, the valuations of all rival bidders is a random sample of size $n - 1$. We denote the highest and second highest of these $n - 1$ valuations by the order statistics Y_1 and Y_2. The probability distribution function of Y_1 is $G(x) := \Pr\{Y_1 \leq x\} = F(x)^{n-1}$, and the joint density function of the order statistics Y_2 and Y_1 is $f_{Y_2 Y_1}(z, y) = (n-1)(n-2)F(z)^{n-3} f(z) f(y)$, for $z \leq y$ (and 0 otherwise), see David (1970, p. 10).

The bidding/corruption game is modeled as a non-cooperative, Bayesian game that involves a combination of simultaneous and sequential moves, as follows.

Stage 1: Bidders simultaneously submit their bids, $b_1, \ldots, b_n$, to the auctioneer.

Stage 2: After bids have been submitted, the auctioneer opens all bids and proposes bid rigging either to the highest or to the second highest bidder. If the bidder rejects a proposal, the auctioneer proceeds without further manipulation. Otherwise, the auctioneer first reveals all the submitted bids to his partner in corruption. If the proposal is made to the highest bidder, the auctioneer allows him to reduce his bid to the next highest bid (*type I corruption*); if the proposal is made to the second highest bidder, the auctioneer allows him to match the highest bid (*type II corruption*).

The surplus that the corrupt coalition achieves is shared in fixed proportions among its members: the auctioneer receives a share of $1 - \alpha$, and his partner in corruption receives the remainder share α. α is common knowledge among all bidders and the auctioneer. The auctioneer chooses the type of corruption that maximizes his payoff.

Let b_1 and b_2 denote the two highest bids that are submitted. The total surplus (for the auctioneer and the corrupt bidder) from type I corruption equals $b_1 - b_2$; the total surplus from type II corruption involves an inference from the bid b_2 to the underlying valuation. It is equal to the difference between the conditional expected value of the valuation of the second highest bidder, $E[V_2 \mid b_2 = \beta(V_2)]$ and the highest bid, b_1, where β denotes the bid function. This introduces a signalling aspect into the bidding problem. We will consider only separating equilibria, i.e. equilibrium bid functions that are strict monotone increasing. There, the auctioneer can draw an exact inference from an observed equilibrium bid b from the image set of β to the underlying valuation, using the inference rule

$v = \beta^{-1}(b)$. If he observes an off-equilibrium bid $b > \beta(1)$, we assume that the auctioneer infers $v = 1$, and if he observes any other off-equilibrium bid he infers $v = 0$. Therefore, the auctioneer proposes type II corruption if and only if $b_1 - b_2 < x_2 - b_1$, where $x_2 := \beta^{-1}(b_2)$; and, that proposal is accepted if and only if $v_2 - b_1 - (1 - \alpha)(x_2 - b_1) > 0$, which is assured, unless the second highest bidder has bid considerably higher that the equilibrium bid (see Lemma 1 below).

Note that the auctioneer cannot completely bypass the highest bid. He must either invite the highest bidder to lower his bid to the level of the second highest bid (type I corruption) or the second highest bidder to match the highest bid (type II corruption). The restriction to these two kinds of bid rigging is due to the fact that the price paid by the winner, and sometimes even all bids, must be published after the auction. This publication requirement serves the purpose to restrict corruption, although it cannot prevent it altogether.[9] Moreover, the auctioneer cannot allow the second highest bidder to get away with paying less than the highest submitted bid to the seller nor the highest bidder to pay less than the second highest bid without involving more than one bidder into the corrupt coalition, which we rule out as an option on the ground that the auctioneer wants to minimize the number of illegal contacts.

Also note that the assumed *ex post* bid rigging is superior to *ex ante* manipulation because it allows the auctioneer to minimize illegal contact and yet deal with the most profitable partner in corruption.

4. The corrupt auctioneer's decision problem

In the following we assume, as a working hypothesis, that a symmetric equilibrium exists that exhibits a strict monotone increasing bid function β and bid shading, $\beta(v) < v$.[10] Such an equilibrium is a "separating equilibrium" since it allows the auctioneer to infer the valuation that underlies an observed bid. Of course, we verify that these working hypotheses confirm in the numerical solutions reviewed in Section 8.

Consider a bidder who has submitted the bid $\beta(x)$ where x may or may not differ from his true valuation v. Denote the highest and the second highest valuations

[9] Most procurement rules require publication of the winning bid (but not of losing bids). See, for example, the Procurement Guidelines by the World Bank and the Asian Development Bank (The World Bank, 2004a,b; The Asian Development Bank, 2002). We mention that our analysis applies regardless of whether only the winning bid is published or all bids are published.

[10] This procedure allows us to find a monotone equilibrium; it does *not* assume monotonicity.

of all other bidders by y_1 and y_2, respectively, and their associated equilibrium bids by $\beta(y_1)$ and $\beta(y_2)$. In order to be able to state that bidder's payoff function, we elaborate the conditions under which that bidder is proposed either type I or type II corruption, or is not proposed corruption and loses the auction.

First, suppose this bidder has made the highest bid, i.e. $\beta(x) > \beta(y_1)$. In that case he is proposed type I corruption if the gain from proposing type I corruption to him is at least as great as that from proposing type II corruption to the bidder who submitted the second highest bid $\beta(y_1)$, $\beta(x) - \beta(y_1) \geq y_1 - \beta(x)$.

Denote the largest y_1 for which this condition is met by $\phi(x)$,

$$\phi(x) = \sup\left\{y_1 \in [0,1] \;\middle|\; y_1 \leq 2\beta(x) - \beta(y_1)\right\}. \tag{1}$$

Then, the bidder who bids $\beta(x)$ is proposed type I corruption if and only if

$$x > y_1 \quad \text{and} \quad y_1 \leq \phi(x). \tag{2}$$

We call this the "proposal-condition for type I corruption". Note that ϕ is an increasing function.

Next, suppose $\beta(x)$ is the second highest bid, $\beta(y_2) < \beta(x) < \beta(y_1)$, which implies $y_2 < x < y_1$. The second highest bidder is proposed type II corruption if $x - \beta(y_1) > \beta(y_1) - \beta(x)$. Denote the "largest" y_1 that satisfies this condition by $\psi(x)$,

$$\begin{aligned}\psi(x) &:= \sup\left\{y_1 \in [0,1] \;\middle|\; x - \beta(y_1) > \beta(y_1) - \beta(x)\right\} \\ &= \begin{cases}\beta^{-1}\left(\frac{x+\beta(x)}{2}\right) & \text{if } (x+\beta(x))/2 < \beta(1), \\ 1 & \text{if } (x+\beta(x))/2 \geq \beta(1).\end{cases}\end{aligned} \tag{3}$$

A bidder who bids $\beta(x)$ is proposed type II corruption if and only if

$$y_2 < x < y_1 \quad \text{and} \quad y_1 < \psi(x). \tag{4}$$

We call this the "proposal-condition for type II corruption". Note that ψ is an increasing function.

Figure 1 illustrates the proposal-conditions (2) and (4).

5. Bidders' decision to accept or reject corruption

A bidder who is proposed type I corruption always accepts since being allowed to lower that bid is always profitable. We will now show that if type II corruption is

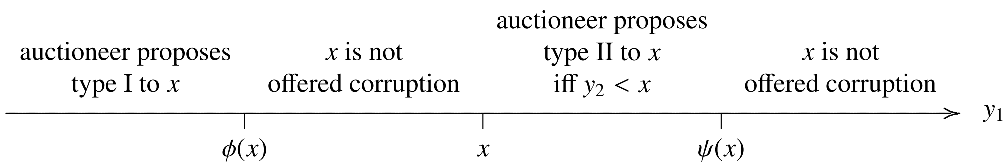

Figure 1: Regions where type I or type II corruption is proposed to bidder who bids $\beta(x)$.

proposed, that proposal is also accepted, as long as the bidder's stated valuation x is close to his true valuation v. We also explain what happens for larger deviations of x from v.

Consider a bidder who has made a bid $\beta(x)$ where x may differ from his valuation v. If he is proposed type II corruption (which occurs if condition (4) is satisfied), the auctioneer demands a transfer of $(1-\alpha)(x-\beta(y_1))$. Therefore, that bidder accepts if and only if

$$v-\beta(y_1)-(1-\alpha)(x-\beta(y_1))>0, \quad \text{resp.}$$
$$y_1<\tilde{\psi}(v,x):=\beta^{-1}\left(\frac{v-(1-\alpha)x}{\alpha}\right). \tag{5}$$

We call this the acceptance-condition for type II corruption. Note that $\tilde{\psi}$ is a decreasing function of x.

Lemma 1. *Suppose x is in a sufficiently small neighborhood of v, then the proposal of type II corruption implies acceptance, i.e.* (4) $\Longrightarrow$ (5).

Proof: Step 1: Suppose $x=v$. Then the acceptance condition, (5), simplifies to $\alpha(v-\beta(y_1))>0$, which is immediately implied by the proposal condition, (4).

Step 2: Let $x:=v+\epsilon$ and define the function r as the product of the functions that characterize the proposal and the acceptance condition, respectively,

$$r(\epsilon):=((v+\epsilon-\beta(y_1))-(\beta(y_1)-\beta(v+\epsilon)))\,(v-\beta(y_1)-(1-\alpha)(v+\epsilon-\beta(y_1)))\,.$$

If the proposal condition is satisfied for $\epsilon=0$, then by step 1, we know that the acceptance condition is also satisfied, i.e. $r(0)>0$. Furthermore, since r is a continuous function, it is also positive in a neighborhood of 0. Hence, if the proposal condition is satisfied for an ϵ in that neighborhood (which is equivalent to the first factor of r being positive), then the acceptance condition (which is captured by the second factor of r being positive) must also be satisfied. □

Figure 2 depicts the combinations of y_1 and x for which the bidder who bids $\beta(x)$ wins the auction. This requires first of all that $x > y_2$. Moreover, below ϕ, in the shaded area I, type I corruption is proposed and, of course, accepted. Above the 45°-line and below ψ, type II corruption is proposed. It is accepted, however, only below $\tilde{\psi}$. Thus, type II corruption takes place in the shaded area II, . This region is bounded from above by the function

$$\psi^*(v, x) := \min\{\psi(x), \max\{\tilde{\psi}(v, x), x\}\}. \tag{6}$$

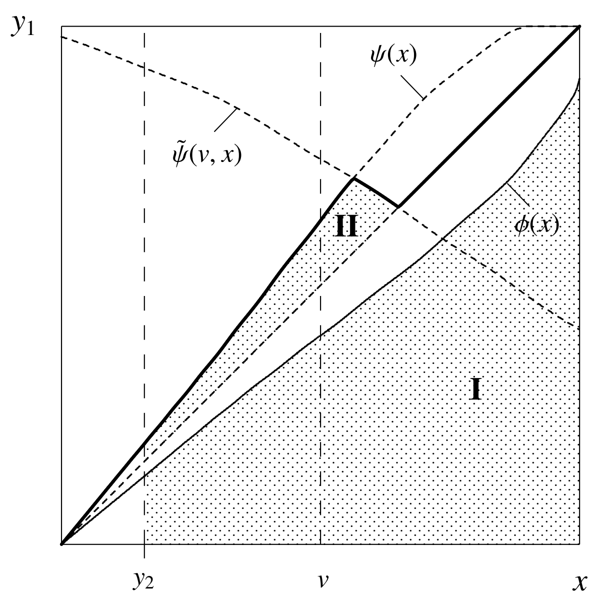

Figure 2: Sets of (y_1, x) where the bidder (v, x) is proposed corruption and either accepts or rejects. The bold line depicts $\psi^*(v, x)$ as defined in (6) as a function of x.

6. First-order conditions

We now characterize the equilibrium bid function for the case of $n \geq 3$, and separately for the somewhat special case $n = 2$.

The payoff of a bidder with valuation v who bids as if his valuation were x, while all others bid the symmetric, strict monotone increasing equilibrium strategy

β, is, for $n \geq 3$,

$$U(v, x) := \int_0^{\phi(x)} (v - \beta(y) - (1 - \alpha)(\beta(x) - \beta(y)))\, dG(y) + \int_x^{\psi^*(v,x)} \int_0^x (v - \beta(y) - (1 - \alpha)(x - \beta(y)))\, f_{Y_2 Y_1}(z, y) dz dy. \quad (7)$$

The first integral is the region where the bidder wins with type I corruption. In this region, his strongest competitor is sufficiently weak ($y < \phi(x)$) so that the auctioneer prefers to offer type I corruption to the highest bidder. The second part is the region where the bidder wins through type II corruption. Two conditions must be met for this to happen. The bid must be the second highest that has been submitted (this is captured by the inner integral with z, the third highest valuation, running from 0 to x), and the distance to the highest bid must not be too large (this is the outer integral, with the highest valuation y running from x to $\psi^*(v, x)$).

Note that, using the joint density of the order statistics, Y_1, Y_2, $f_{Y_2 Y_1}(z, y)$, one can simplify the second line of (7) using the fact that

$$\int_0^x f_{Y_2 Y_1}(z, y) dz = (n - 1) f(y) F(x)^{n-2} = G'(x) \frac{f(y)}{f(x)}. \quad (8)$$

When there are three or more bidders, a bidder is proposed type II corruption if three conditions are met: first, his bid must be lower than the highest rival bid, but second, not too much lower, and third, it must be higher than the second highest rival bid. The third requirement is meaningless if there are only two bidders. Therefore, if $n = 2$ the payoff function simplifies to

$$U(v, x) = \int_0^{\phi(x)} (v - \beta(y) - (1 - \alpha)(\beta(x) - \beta(y))) dF(y) + \int_x^{\psi^*(v,x)} (v - \beta(y) - (1 - \alpha)(x - \beta(y))) dF(y). \quad (9)$$

Inspection of (9) together with (8) and $n = 2$ reveals that (9) is just a special case of (7).

The strategy β is a symmetric equilibrium if $v = \arg\max_x U(v, x)$ for all v. Using the first-order condition of this requirement (keeping in mind that $\psi^*(v, x) =$

$\psi(x)$ for x in a neighborhood of v by Lemma 1), one obtains

$$\begin{aligned}
0 = {} & \phi'(v)(v - (1-\alpha)\beta(v) - \alpha\beta(\phi(v)))G'(\phi(v)) \\
& - (1-\alpha)\left(\beta'(v)G(\phi(v)) + \frac{G'(v)}{f(v)}(F(\psi(v)) - F(v))\right) \\
& + \alpha(v - \beta(v))\frac{G'(v)}{f(v)}\left(\frac{\psi'(v)f(\psi(v))}{2} - f(v)\right) \\
& + \alpha(n-1)(n-2)F(v)^{n-3}f(v)\int_v^{\psi(v)}(v - \beta(y))dF(y).
\end{aligned} \tag{10}$$

7. Existence

Equation (10) is a delay differential equation. The theory of delay differential equations assures the existence of a solution (Kuang, 1993, Theorem 2.1), but it does not guarantee existence of a monotone solution.[11] So it is not clear whether our game has an equilibrium or not.

An analytical solution is available for $\alpha = 1$. In this case, truthful bidding is an equilibrium. To see why, note that $\phi(v) = v$ and $\psi(v) = v$ if $\beta(v) = v$, by (1) and (3), respectively. Use this fact, set $\alpha = 1$, and check that $\beta(v) = v$ solves (10).[12]

In general, however, (10) cannot be solved analytically. In the following we establish a necessary condition that restricts the set of parameters for which a monotone solution exists.

Suppose β is a solution of (10) for the uniform distribution, and let $s := \beta'(0)$. Consider the first-order Taylor approximation of β at 0. Then $\beta(v) \approx sv$, $\phi(v) \approx$

[11] β is not necessarily differentiable at some points. For instance, it may have a kink at the smallest v where $\psi(v) = 1$, so both, left-hand and right-hand derivatives, exist, but do not coincide. However, this does not cause any problem, because the derivative in the delay differential equation is defined as the right-hand derivative.

[12] The statement is also valid if we take the limit of α: $\forall v \lim_{\alpha\to 1}\beta(v) = v$. Observe that $\phi(v) \to v$ and $\psi(v) \to v$ as $\beta(v) \to v$. Using these facts, one checks that $\beta(v) = v$ solves (10) as $\alpha \to 1$. Similarly, $\forall v \lim_{n\to\infty}\beta(v) = v$. The argument is again very similar. Because $\phi(v) \to v$ and $\psi(v) \to v$ as $\beta(v) \to v$, and because $\lim_{n\to\infty} G(v) = 0$ for all $v < 1$, $\beta(v) = v$ solves (10) as $n \to \infty$.

$2s/(1+s)$, and $\psi(v) \approx (1+s)/(2s)$ for $v \approx 0$. Therefore, s solves the polynomial

$$\begin{aligned} h(s,\alpha,n) := \frac{2s}{1+s}&\left(1-(1-\alpha)s-\alpha s\frac{2s}{1+s}\right)(n-1)\left(\frac{2s}{1+s}\right)^{n-2} \\ &-(1-\alpha)s\left(\frac{2s}{1+s}\right)^{n-1}+\alpha(1-s)(n-1)\left(\frac{1+s}{4s}-1\right) \\ &-(1-\alpha)(n-1)\left(\frac{1+s}{2s}-1\right) \\ &+\alpha(n-1)(n-2)\left(\frac{1+s}{2s}-1-\frac{s}{2}\left(\left(\frac{1+s}{2s}\right)^2-1\right)\right). \end{aligned} \tag{11}$$

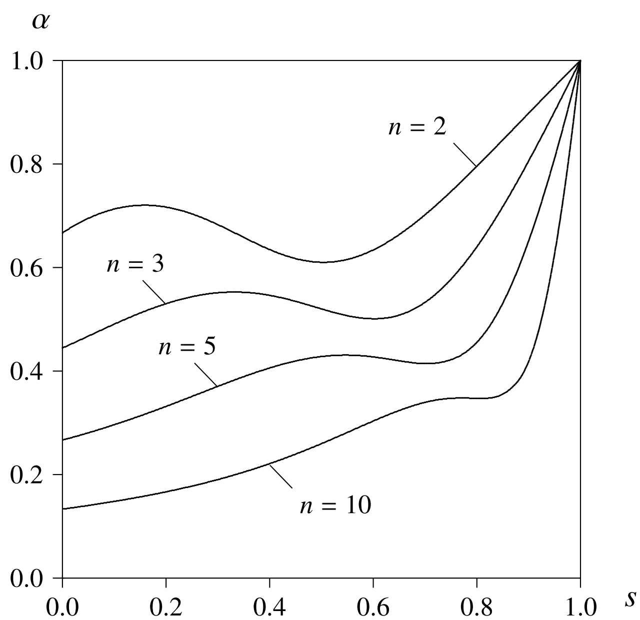

Figure 3: Roots of the polynomial (11) for various n.

Figure 3 plots combinations of s and α for which $h(s,\alpha,n) = 0$. Notice that, for given n, there is no real root in the unit interval if α is too small. Define $\alpha^*(n) := \inf\{\alpha : \exists s \in [0,1]\ h(s,\alpha,n) = 0\}$. An equilibrium cannot exist if $\alpha < \alpha^*(n)$.

8. Numerical analysis

Because (10) cannot be solved analytically in general, we analyze it with the help of numerical methods. Throughout we assume uniformly distributed valuations. We compute the equilibrium for various combinations of n and α. All the details can be found in the technical appendix.

We also check numerically whether the solution, which by construction satisfies only the local best reply conditions, constitutes also a global best reply. In standard auction problems this is typically achieved by showing that the payoff function $U(v, x)$ is pseudo-concave in x. Unfortunately, pseudo-concavity does not hold in the present problem. However, by computing the payoff function $U(v, x)$ for all valuations v and all deviations x, we verify numerically that the computed bid functions are indeed global best replies. The details are again spelled out in the technical appendix.

The numerical analysis gives rise to five results, for which we will also provide intuition, as far as possible.

The first result relates to the question of existence of an equilibrium. Of course, numerical methods can never prove existence, because they give us an approximate solution at best. And yet, the following result suggests that an equilibrium does indeed exist as long as α is large enough.

Result 1. *Numerical solutions are obtained with high accuracy if the share of the surplus that goes to the winning bidder (α) is sufficiently large. Below a certain threshold value of α, the precision deteriorates discontinuously. This suggests that an equilibrium exists only for sufficiently large α.*

In Table G.4 we report the numerical approximation error for various combinations of n and α. For $n = 2$, this error increases discontinuously when we change α from 0.6101 to 0.6100. This suggests that the necessary condition $\alpha \geq \alpha^*(2) = 0.6101$ is also sufficient if $n = 2$. This is not true for larger n. For instance, $\alpha^*(5) = 0.2667$, yet we find numerical solutions of (10) with "small error" only for $\alpha \geq 0.4515$. More precisely, the approximation error changes by several orders of magnitude (from 10^{-18} to 10^{-5}) when we change α from 0.4515 to 0.4514. We take this as evidence that an approximate equilibrium exists for the former α, but fails for the latter.

Figure 4 depicts numerical solutions of the equilibrium bid functions.

Result 2. *If α is reduced below 1, welfare decreases, the expected payoffs of the auctioneer increases, and that of the seller decreases.*

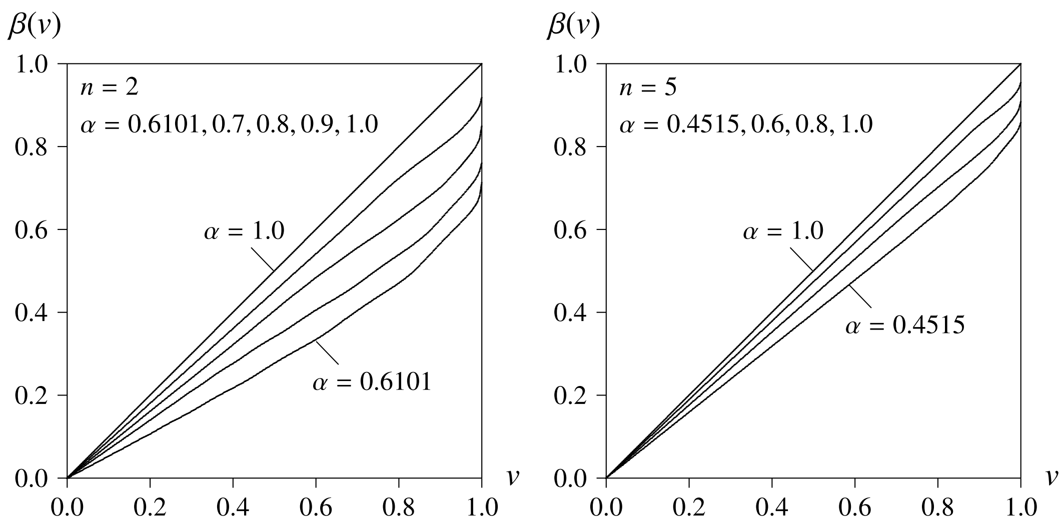

Figure 4: Approximate equilibrium bid functions for the uniform distribution, for $n = 2$ (left panel) and $n = 5$ (right panel) and various values for α.

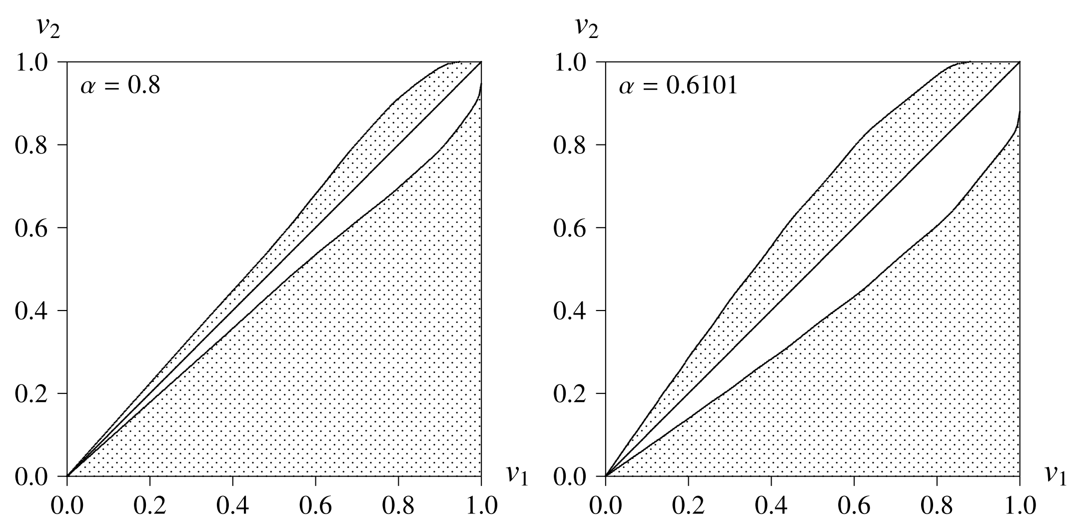

Figure 5: Equilibrium allocation for $n = 2$ and two different values of α. Bidder 1 wins in the shaded regions, bidder 2 in the white regions.

Table 1: Numerical accuracy: sum of squared errors, $\mathrm{SSE} = \sum_{v \in \mathbb{G}} D(v)^2$, for various α and n.

$n = 2$	α	SSE
	0.9	4.40×10^{-22}
	0.8	2.23×10^{-17}
	0.7	9.51×10^{-15}
	0.6101	6.17×10^{-12}
. . .	. . .	. . .
	0.6100	9.76×10^{-8}
	0.6	3.04×10^{-8}
	0.5	5.51×10^{-2}

$n = 3$	α	SSE
	0.9	3.82×10^{-28}
	0.8	4.39×10^{-25}
	0.7	9.73×10^{-24}
	0.6	2.15×10^{-24}
	0.5010	3.98×10^{-16}
. . .	. . .	. . .
	0.5009	5.73×10^{-9}
	0.5	1.00×10^{-1}

$n = 5$	α	SSE
	0.9	4.18×10^{-21}
	0.8	6.32×10^{-21}
	0.7	5.85×10^{-20}
	0.6	8.84×10^{-17}
	0.5	7.64×10^{-19}
	0.4515	2.55×10^{-18}
. . .	. . .	. . .
	0.4514	1.24×10^{-5}
	0.4	2.11×10^{-1}

$n = 10$	α	SSE
	0.9	2.23×10^{-19}
	0.8	9.75×10^{-18}
	0.7	6.07×10^{-18}
	0.6	4.15×10^{-17}
	0.5	1.25×10^{-16}
	0.4013	2.95×10^{-15}
. . .	. . .	. . .
	0.4012	8.74×10^{-10}
	0.4	2.17×10^{-5}

Figure 5 depicts the equilibrium allocations in the state space for $\alpha = 0.8$ and for $\alpha = 0.6101$ (the smallest α for which we have found a solution). Bidder 1 wins the object in the shaded area; in the white area, bidder 2 is the winner. Efficiency requires that bidder 1 wins if and only if he has the higher valuation, $v_1 > v_2$, and *vice versa*. Therefore, for efficiency, the entire area below the 45°-line should be shaded, and the entire area above it should be white. Clearly, the equilibrium allocation is not efficient. In both parameter cases, there is a white wedge in the area below the 45°-line that should be shaded (there, bidder 2 wins although he has the lower valuation), and a shaded wedge in the area above the 45°-line that should be white (there, bidder 1 wins although he has the lower valuation). These "wedges" indicate the presence of type II corruption which changes the allocation by letting the second highest bidder win the auction.

Comparing the two figures for $\alpha = 0.6101$ and $\alpha = 0.8$ illustrates the second fact that the two inefficiency "wedges" increase in size as α is lowered. The intuition for this is as follows: type II corruption requires bid shading. If α is close to one, the equilibrium bid function is close to truthful bidding, therefore there is almost no room for type II corruption. As α is lowered, bids are shaded more, and this, in turn, makes more room for type II corruption, by increasing the spread of valuations, $v_1 - v_2$, for which the auctioneer benefits the most from allowing the second highest bidder to win.

We now compute the welfare loss (π_{loss}) and the the expected equilibrium payoffs of the seller (π_{seller}) and the auctioneer (π_{auc}):

$$\begin{aligned}\pi_{\text{auc}} &:= (1-\alpha)\int_0^1 \bigg(\int_0^{\phi(x)} (\beta(x)-\beta(y)) f_{X_2X_1}(y,x)dy \\ &\qquad + \int_{\phi(x)}^{x} (y-\beta(x)) f_{X_2X_1}(y,x)dy \bigg) dx, \end{aligned} \tag{12}$$

$$\pi_{\text{seller}} := \int_0^1 \left(\int_0^{\phi(x)} \beta(y) f_{X_2X_1}(y,x)dy + \int_{\phi(x)}^{x} \beta(x) f_{X_2X_1}(y,x)dy \right) dx, \tag{13}$$

$$\pi_{\text{loss}} := \int_0^1 \int_{\phi(x)}^{x} (x-y) f_{X_2X_1}(y,x)dydx, \tag{14}$$

where $f_{X_2X_1}(y,x) := n(n-1)F(y)^{n-2}f(y)f(x)$, $y \le x$, denotes the joint density of the highest and second highest of a sample of n valuations.

Table 2 summarizes how welfare and payoffs change compared to the equilibrium in the absence of corruption. Not surprisingly, as α is reduced, the auctioneer's expected profit increases, expected welfare diminishes, and the seller's expected profit decreases as well.

Result 3. *If α is reduced below* 1*, not only the auctioneer's but also bidders' expected payoffs increase. Paradoxically, both parties benefit from a 'stronger auctioneer' because they both benefit* more *when the share of the surplus that goes to the auctioneer is increased. This indicates that corruption is hard to fight as both involved parties benefit from it.*

The surprising fact that bidders' payoffs increase when the share that goes to the auctioneer is increased, can be explained from Myerson's revenue equivalence theorem together with the observation that the allocation rule is more distorted the smaller α is.

A bidder with valuation v wins the first-price auction with probability $H_\alpha(v) := G(\psi(v)) - G(v) + G(\phi(v))$. This differs from the efficient allocation rule $G(v)$, which is the cause of inefficiency.

In all our computations, H_α is a monotone increasing function. Therefore, by Myerson (1981, Lemma 2), bidders' expected payoff is

$$U(v,v) = \int_0^v H_\alpha(y)dy = \int_0^v (G(\psi(v)) - G(v) + G(\phi(v)))dy, \tag{15}$$

Table 2: Changes of expected welfare and payoffs due to corruption compared to no corruption.

$n = 2$	α	welfare	auctioneer	seller
	0.9	−0.001 (−0.2%)	+0.030	−0.032 (−9.6%)
	0.8	−0.005 (−0.7%)	+0.054	−0.063 (−18.9%)
	0.7	−0.011 (−1.6%)	+0.074	−0.097 (−29.0%)
	0.6101	−0.019 (−2.9%)	+0.095	−0.134 (−40.1%)
$n = 3$	α	welfare	auctioneer	seller
	0.9	−0.001 (−0.1%)	+0.024	−0.025 (−5.0%)
	0.8	−0.002 (−0.3%)	+0.045	−0.050 (−10.0%)
	0.7	−0.006 (−0.8%)	+0.065	−0.077 (−15.4%)
	0.6	−0.011 (−1.5%)	+0.086	−0.109 (−21.8%)
	0.5010	−0.019 (−2.5%)	+0.112	−0.150 (−30.0%)
$n = 5$	α	welfare	auctioneer	seller
	0.9	−0.000 (−0.0%)	+0.016	−0.017 (−2.5%)
	0.8	−0.001 (−0.2%)	+0.032	−0.035 (−5.2%)
	0.7	−0.003 (−0.4%)	+0.047	−0.054 (−8.1%)
	0.6	−0.006 (−0.8%)	+0.063	−0.076 (−11.4%)
	0.5	−0.011 (−1.3%)	+0.082	−0.104 (−15.6%)
	0.4515	−0.013 (−1.6%)	+0.094	−0.120 (−18.0%)
$n = 10$	α	welfare	auctioneer	seller
	0.9	−0.000 (−0.0%)	+0.009	−0.010 (−1.2%)
	0.8	−0.001 (−0.1%)	+0.018	−0.021 (−2.6%)
	0.7	−0.002 (−0.2%)	+0.026	−0.032 (−3.9%)
	0.6	−0.003 (−0.4%)	+0.036	−0.045 (−5.4%)
	0.5	−0.005 (−0.6%)	+0.047	−0.060 (−7.3%)
	0.4013	−0.008 (−0.9%)	+0.062	−0.080 (−9.7%)

since by construction $U(0,0) = 0$. Moreover, we find that for $\alpha < 1$, G is a mean preserving spread of H_α, and for $\alpha' < \alpha$, H_α is a mean preserving spread of $H_{\alpha'}$. This is illustrated in Figure 6, where we plot G and H for two values of α and $n = 2$. A mean preserving spread implies second order stochastic dominance. Thus, for $\alpha' < \alpha < 1$,

$$\int_0^v G(y)dy \leq \int_0^v H_\alpha(y)dy \leq \int_0^v H_{\alpha'}(y)dy$$

for all v, with strict inequality for all $v \notin \{0, 1\}$. This proves that bidders' equilibrium expected payoff, $U(v, v)$, is decreasing in α.

Of course, we know from result 1 that α cannot be reduced arbitrarily, because existence of a separating equilibrium breaks down if α is too small.

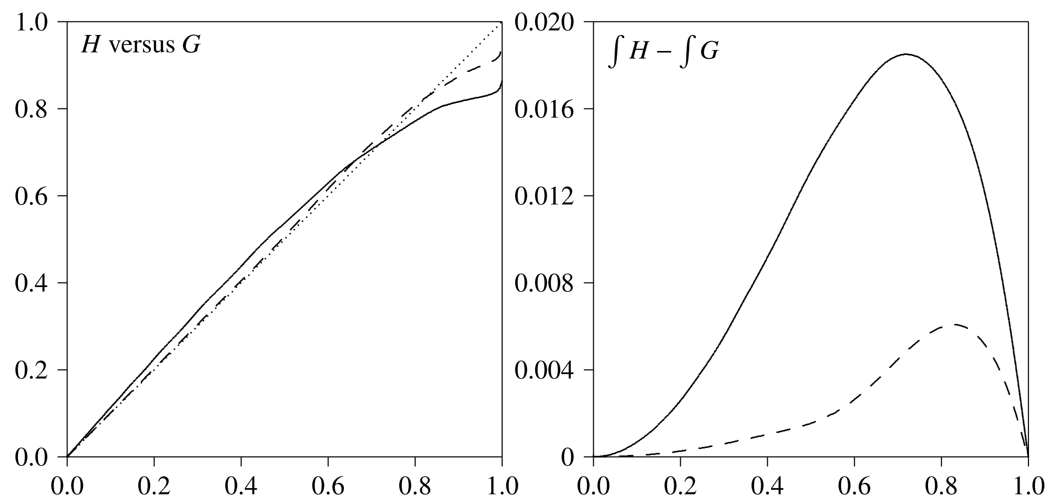

Figure 6: *Left panel*: winning probability of an agent as function of his valuation if there are two bidders ($n = 2$), in the efficient allocation (G, dotted line) and in the auction with corruption (H); α is 0.6101 (solid line) or 0.8 (dashed line), respectively. *Right panel*: expected gain of bidders from corruption as function of their valuation.

Result 4. *If the number of bidders is increased, welfare improves, the expected payoff of the seller increases, and the expected payoffs of the auctioneer and bidders decrease.*

The fact that welfare and the seller's payoff improve and that the auctioneer looses from more competition can be seen from Table 2. The numerical computations also confirm that bidders' payoff decreases with competition. While the result concerning welfare, the seller's payoff, and bidders' payoff may be as expected, we have no compelling intuition why the auctioneer should loose from more intense competition among bidders. This may be an artefact of the uniform distribution.

Result 5. *The seller's expected payoff is substantially smaller in a model that allows for both types of corruption compared to a model that allows only for type I corruption.*

Table 3 compares the effect of corruption in our model to the alternative model that only considers type I corruption (as in Lengwiler and Wolfstetter (2000) and Menezes and Monteiro (2006)). Defining $K(x) := G(x)^{\frac{1}{1-\alpha}}$, one can show that

Table 3: Expected payoff of the seller and expected reduction of this payoff due to corruption. The table compares the effect in a simpler model that considers only type I corruption with the effect in our model that allows for both types of corruption.

α	**type I corruption only**	**type I & type II corruption**
$n = 2$	seller's expected payoff without corruption is 0.333	
0.9	0.303 (−9.1%)	0.301 (−9.6%)
0.8	0.278 (−16.7%)	0.270 (−18.9%)
0.7	0.256 (−23.1%)	0.236 (−29.0%)
0.6101	0.240 (−28.1%)	0.199 (−40.1%)
$n = 3$	seller's expected payoff without corruption is 0.500	
0.9	0.476 (−4.8%)	0.475 (−5.0%)
0.8	0.455 (−9.1%)	0.450 (−10.0%)
0.7	0.435 (−13.0%)	0.423 (−15.4%)
0.6	0.417 (−16.7%)	0.391 (−21.8%)
0.5010	0.400 (−20.0%)	0.350 (−30.0%)
$n = 5$	seller's expected payoff without corruption is 0.667	
0.9	0.650 (−2.4%)	0.650 (−2.5%)
0.8	0.635 (−4.8%)	0.632 (−5.2%)
0.7	0.620 (−7.0%)	0.613 (−8.1%)
0.6	0.606 (−9.1%)	0.591 (−11.4%)
0.5	0.593 (−11.1%)	0.563 (−15.6%)
0.4515	0.586 (−12.1%)	0.547 (−18.0%)
$n = 10$	seller's expected payoff without corruption is 0.818	
0.9	0.809 (−1.1%)	0.808 (−1.2%)
0.8	0.800 (−2.2%)	0.797 (−2.6%)
0.7	0.792 (−3.2%)	0.786 (−3.9%)
0.6	0.783 (−4.3%)	0.773 (−5.4%)
0.5	0.775 (−5.3%)	0.758 (−7.3%)
0.4013	0.767 (−6.2%)	0.738 (−9.7%)

$\beta(v) = v - \int_0^v \frac{K(y)}{K(v)} dy$ solves this auction game. Compared to the model that considers only type I corruption, we see that the interplay between type I and type II corruption has a significant impact. First of all, it gives rise to inefficiency, which cannot occur when only type I corruption is possible, because this only affects the pricing rule but not the allocation rule. Second, the seller's loss due to corruption is significantly higher when one allows for both types of corruption. This is true in particular for cases with only few bidders. For instance, suppose that $n = 3$, then the seller's expected payoff without corruption is 0.5. As an example and guide for reading the table, suppose further that only type I corruption is possible and $\alpha = 0.6$. Then the seller's expected payoff is 0.417, so the seller loses 16.7% of the payoff that he would have achieved without corruption. If type I and type II corruption are possible, however, the seller's expected payoff is only 0.308, and the seller loses 21.8% of his payoff without corruption.

9. Conclusion

We have examined how bidding behavior, efficiency, and the allocation of payoffs is affected by corruption in a first-price auction with risk-neutral bidders and independent private values. Corruption here means the ability of the auctioneer (who is an agent of the seller) to make a side deal with one bidder at the expense of the seller. The analysis is considerably complicated by the fact that the auctioneer has a choice with which bidder to initiate corruption. He can either invite the highest bidder to lower his bid (we call this type I corruption) or he can invite the second highest bidder to raise his bid (type II corruption).

We find: (1) In a symmetric, monotone equilibrium the object is not always awarded to the bidder who has the highest valuation. Inefficiency occurs whenever the auctioneer allows the second highest bidder to match the highest bid (type II corruption). (2) The associated welfare loss is a decreasing function of the number of bidders, and an increasing function of the share claimed by the corrupt auctioneer. (3) The expected gains from corruption are also decreasing in the number of bidders. Surprisingly, both parties gain more when the share claimed by the auctioneer is increased (up to the limit when a pure strategy equilibrium fails to exist). This suggests that even bidders prefer being matched with a strong corrupt auctioneer who claims a large share. (4) Only the auctioneer and the winning bidder have hard evidence of the corrupt activity, yet both benefit from corruption. This contributes to explain why fighting corruption is intrinsically difficult.

Our analysis rests on the assumption that the corrupt auctioneer contemplates only one illegal contact. We justify this with the fact that bid rigging schemes that involve several bidders are subject to increased risk of detection, because every transfer leaves some hard evidence that may be traced. Nevertheless, a more general model would allow the auctioneer to optimize not only the type of corruption, but also the size of the corrupt coalition.

Another limitation concerns the existence of a monotone symmetric equilibrium bid function. Intuitively, there are two reasons that may prevent existence of such an equilibrium: first, bidders are not unambiguously interested in making the highest bid, and second, because the corrupt auctioneer uses bids to infer the unknown valuation of the second highest bidder, bidders have an interest in sending a distorted signal. The issue is similar to the inference problem in auctions with resale, where existence of a monotone symmetric equilibrium is not generally assured (see Haile (2003) and the literature reviewed there). In fact, we have proved that a monotone equilibrium cannot exist in our model if the share of the auctioneer is too large (α is too small). Yet, for small enough shares of the auctioneer (large enough α), we are able to produce highly accurate numerical approximations. Depending on the parameters, the maximum absolute deviation of the delay-differential equation (10) from zero is always smaller than 4×10^{-6}, and typically in the neighborhood of 10^{-10} (see the technical appendix for details). Although invoking numerical solution methods does not prove existence, the low error suggests that our numerical solution indeed approximates the true solution. Alternatively, one may argue that we have at least found an approximate equilibrium in which bidders who play according to our numerical solution would make only extremely small mistakes.

The main policy implication of our analysis is that fighting corruption is paramount because, due to presence of type II corruption, it does not only redistribute rents but also distorts efficiency. Our analysis also suggests that accumulating hard evidence about corruption may be difficult in a sealed bid auction because all parties that own hard evidence profit from the corrupt activity. One way to make it easier to collect information about corruption is to employ an open bidding format. However, open auctions facilitate collusion among bidders, which may make this cure worse than the disease.

One way to deter the kind of corruption that we have analyzed here is to give the auctioneer an incentive contract that pays him a share of the profit. This makes corruption less attractive or may even remove it completely. In fact, we observe such incentive schemes in private markets where auction houses compete for sellers.[13]

Moreover, one can use technology to make bid rigging impossible. Computer scientists have developed encryption and time stamping technologies that contribute to prevent bid rigging (see Parkes et al., 2007). These methods should be applied in auctions and procurements, both in the private and public sector and international organizations.

However, when the seller is a government or a bureaucratic entity, rather than

[13]For example, for live auctions at Sotheby's salesrooms the fees are a non-linear function of the hammer price. The rate is 25% on the first \$50.000, 20% on the exceeding amount up to \$1.000.000, and 12% on the remaining amount, see `http://www.sothebys.com/help/faq/faq_duringauction.html`.

a residual claimant, these solutions are often ignored due to a lack of proper governance.

Literature

Aoyagi, M., 2007. Efficient collusion in repeated auctions. Journal of Economic Theory 134, 61–92.

Arozamena, L., Weinschelbaum, F., 2009. The effect of corruption on bidding behavior in first-price auctions. European Economic Review 53, 645–657.

Athey, S., Bagwell, K., Sanchirico, C., 2004. Collusion and price rigidity. Review of Economic Studies 71, 317–349.

Athey, S., Levin, J., Seira, E., 2008. Comparing Open and Sealed Bid Auctions: Evidence from Timber Auctions. Working Paper. NBER Working Paper, # 14590.

Bikhchandani, S., Lippman, S.A., Ryan, R., 2005. On the right-of-first-refusal. Advances in Theoretical Economics 5, 1–42.

Burguet, R., Che, Y.K., 2004. Competitive procurement with corruption. RAND Journal of Economics 35, 50–68.

Burguet, R., Perry, M.K., 2007. Bribery and favoritism by auctioneers in sealed-bid auctions. The B.E. Journal of Theoretical Economics 7.

Celantani, M., Ganuza, J.J., 2002. Corruption and competition in procurement. European Economic Review 46, 1273–1303.

Compte, O., Lambert-Mogiliansky, A., Verdier, T., 2005. Corruption and competition in public market auctions. RAND Journal of Economics 36, 1–15.

David, H.A., 1970. Order Statistics. Wiley Series in Probability and Mathematical Statistics, Wiley.

Graham, D.A., Marshall, R.C., 1987. Collusive bidder behavior at single–object second–price and English auctions. Journal of Political Economy 95, 1217–1239.

Grosskopf, B., Roth, A.E., 2009. If you are offered the Right of First Refusal, should you accept? an investigation of contract design. Games and Economic Behavior 65, 176–204.

Güth, W., 1995. Preisregeln für Auktionen und Ausschreibungen: Eine ordnungspolitische Analyse. Zeitschrift f. Wirtschafts– und Sozialwissenschaften 115, 1–26.

Güth, W., van Damme, E., 1986. A comparison of pricing rules for auctions and fair division games. Social Choice and Welfare 3, 177–189.

Haile, P., 2003. Auctions with private uncertainty and resale opportunities. Journal of Economic Theory 108, 72–110.

Heath, M.T., 2002. Scientific Computing. An Introductory Survey. McGraw-Hill. 2nd edition.

Ingraham, A.T., 2001. Bidder-Auctioneer Cheating in Sealed Bid Auctions. Ph.D. Thesis, University of Maryland.

Jensen, I., 1984. Quellen und Zeugnisse zur Druckgeschichte von Goethes Werken. Akademie-Verlag.

Koc, S., Neilson, W.S., 2008. Interim bribery in auctions. Economics Letters 99, 238–241.

Kuang, Y., 1993. Delay Differential Equations with Applications in Population Dynamics. Mathematics in Science and Engineering, Academic Press.

Laffont, J.J., Tirole, J., 1991. Auction design and favoritism. International Journal of Industrial Organization 9, 9–42.

Lengwiler, Y., Wolfstetter, E., 2000. Auctions and Corruption. Working Paper 401. CESifo (revised 2002 and 2004).

Mailath, G., Zemsky, P., 1991. Collusion in second-price auctions with heterogeneous bidders. Games and Economic Behavior 3, 467–486.

McAfee, R., McMillan, J., 1992. Bidding rings. American Economic Review 82, 579–599.

Menezes, F., Monteiro, P., 2006. Corruption and auctions. Journal of Mathematical Economics 42, 97–108.

Moldovanu, B., Tietzel, M., 1998. Goethe's second-price auction. Journal of Political Economy 106, 854–859.

Myerson, R.B., 1981. Optimal auction design. Mathematics of Operations Research 6, 58–73.

Parkes, M., Rabin, O., Shieber, S., Thorpe, C., 2007. Practical secrecy-preserving, verifiably correct and trustworthy auctions. Electronic Commerce Research and Applications 7, 294–312.

Porter, R., Zona, J., 1999. Ohio school milk markets: An analysis of bidding. RAND Journal of Economics 30, 263–288.

Riley, J.G., 1989. Expected revenue from open and sealed bid auctions. Journal of Economic Perspectives 3, 41–50.

The Asian Development Bank, 2002. Handbook for Users of Consulting Services: Procedures and Practices. Asian Development Bank, Manila.

The World Bank, 2004a. Guidelines: Procurement of Goods and Services by World Bank Borrowers. The International Bank for Reconstruction and Development. The World Bank, Washington, D.C.

The World Bank, 2004b. Guidelines: Selection and Employment of Consultants by World Bank Borrowers. The International Bank for Reconstruction and Development. The World Bank, Washington, D.C.

Technical Appendix

Appendix A. Numerical solution: the problem

As explained in the paper, we want to find the solution to

$$\begin{aligned} D(v) := {} & \phi'(v)(v - (1-\alpha)\beta(v) - \alpha\beta(\phi(v)))(n-1)\phi(v)^{n-2} \\ & - (1-\alpha)\Big(\beta'(v)\phi(v)^{n-1} + (n-1)v^{n-2}(\psi(v) - v)\Big) \\ & + \alpha(v - \beta(v))(n-1)v^{n-2}\left(\frac{\psi'(v)}{2} - 1\right) \\ & + \alpha(n-1)(n-2)v^{n-3}\left(\Big(\psi(v) - v\Big)v - \int_v^{\psi(v)} \beta(y)dy\right) = 0. \end{aligned} \tag{A.1}$$

The evaluation of (A.1) is not trivial because we first need to determine $\psi(v)$ and $\phi(v)$, which depend on β. ψ involves the inverse of β, so we will have to make sure that our candidate β is always invertible. ϕ demands a little more. It is the solution of a fixed point problem,

$$\phi(v) + \beta(\phi(v)) - 2\beta(v) = 0. \tag{A.2}$$

Because the left-hand side of this equation is monotonic in $\phi(v)$, and is negative if $\phi(v) = 0$, and positive if $\phi(v) = v$ (as long as there is bid shading, $\beta(v) < v$), we can use the bi-section method and determine $\phi(v)$ to an arbitrary precision. The required precision is determined in the code by `phiEps` and set by default to machine precision. For the remaining, we draw heavily from the splendid book by Heath (2002).

Appendix B. Root finding methods

To make problem (A.1) suitable for numerical analysis we search for approximate solutions within the class of strictly increasing, continuous, piecewise linear functions that satisfy the boundary condition $\beta(0) = 0$. Let $\mathbb{G} := \{0, 1/g, \dots, (g-1)/g, 1\}$, for some $g \in \mathbb{N}$, be a uniform grid on the unit interval. A piecewise linear function is defined by the numbers $\{\beta(v) : v \in \mathbb{G}\}$. This transforms the infinite-dimensional problem (A.1) into a g-dimensional problem. You can select the fineness of the grid on valuation space by changing the constant g in the source code.

All iterative procedures start from some initial "guess" (more on this later). Let β_0 be the initial bid function, and $\beta_1, \beta_2, \ldots$ denote the bid functions along an iteration. A root finding algorithm is a rule that prescribes how to get from β_i to β_{i+1}. The idea is to do this in such a way that the sequence of β_i approaches the true solution in the process. We have implemented several standard methods for such problems. You can select the method by setting the switch `rootmethod` to a value between `0` and `3`.

Appendix B.1. The Gauss-Newton method

A standard method for moving from β_i to β_{i+1} is the Gauss-Newton method. It involves the Jacobian of D_i with respect β_i, which we denote with J_i. To approximately compute the components of the Jacobian, for each $v_k := k/g \in \mathbb{G}$, we compute D_i at two points, namely β_i^+ and β_i^-, given by $\beta_i^+(v_k) := \beta_i(v_k) + \delta$, $\beta_i^-(v_k) := \beta_i(v_k) - \delta$, and $\beta_i^+(v) = \beta_i^-(v) = \beta_i(v)$ for all $v \neq v_k$.[14] The Jacobian is then approximately given by

$$J \approx \begin{bmatrix} \frac{D_1(\beta_1^+)-D_1(\beta_1^-)}{2\delta} & \cdots & \frac{D_1(\beta_g^+)-D_1(\beta_g^-)}{2\delta} \\ \vdots & \ddots & \vdots \\ \frac{D_g(\beta_1^+)-D_g(\beta_1^-)}{2\delta} & \cdots & \frac{D_g(\beta_g^+)-D_g(\beta_g^-)}{2\delta} \end{bmatrix}.$$

The iteration proceeds according to the following rule

$$\beta_i \mapsto \beta_{i+1} := \beta_i - \tau_i J_i^{-1} D_i. \tag{B.1}$$

τ_i is a positive number called the step size. As explained later, we optimize over the step size τ_i.

Appendix B.2. Steepest descent method

Another standard method is the method of steepest descent. Compute the sum of squared errors,

$$\mathrm{SSE}_i := \frac{1}{2} \sum_{v \in \mathbb{G}} D_i(v)^2,$$

(the division by 2 is a normalization that will make sense later). We aim at minimizing SSE. To that avail we compute the gradient of SSE with respect to

[14] δ is called `diffDelta` in the program, and is equal to `diffDelta = 1E-10` by default; you are free to change this parameter.

$\{\beta_i(v) : v \in \mathbb{G}\}$. As before, we approximate the gradient by computing a discrete difference of the components of the bid function ($\pm\delta$): we compute $\mathrm{SSE}_i^+(\beta_i, v)$ as the SSE that would result if $\beta_i(v) \mapsto \beta_i(v) + \delta$ for a given $v \in \mathbb{G}$, and $\mathrm{SSE}_i^-(\beta_i, v)$ as the SSE that would result if $\beta_i(v) \mapsto \beta_i(v) - \delta$. The gradient is then approximately equal to

$$\nabla\mathrm{SSE}(\beta_i) := \left(\frac{\mathrm{SSE}_i^+(\beta_i, v) - \mathrm{SSE}_i^-(\beta_i, v)}{2\delta}\right)_{v\in\mathbb{G}}.$$

With this information, we can iterate according to the following rule

$$\beta_i \mapsto \beta_{i+1} := \beta_i - \tau_i \nabla\mathrm{SSE}(\beta_i). \tag{B.2}$$

We move here into the direction in which SSE decreases the fastest locally, hence the name of the method. τ_i is again the step size, which we optimize.

Appendix B.3. A hybrid method

The Gauss-Newton method finds the minimum of a quadratic function in one iteration. For non-quadratic problems, the initial point has to be sufficiently close to the solution. If not, the method might not converge at all. So this method is fast if we are close to the solution, but quite demanding in terms of choice of starting point.

Compared to the Gauss-Newton method, the steepest descent method is slow. Like the Gauss-Newton method, it exhibits global convergence for quadratic problems (though not in one iteration step). The advantage of steepest descent over the Gauss-Newton method is that steepest descent is much less demanding with respect to the starting point. Even if we are far from the solution, the method will point into more or less the right direction[15] and initially converge at a decent speed. Only when we get close to the solution does convergence speed deteriorate.

These observations suggest a hybrid method, that combines the advantages of the steepest descent method in the early phase of the process, and later switches to the Gauss-Newton method. We switch from steepest descent to Gauss-Newton when there has been no significant improvement (`switchEps = 1E-12`) over sufficiently many consecutive iterations (`switchKeep = 5`).

Appendix B.4. The Levenberg-Marquardt method

This method is similar in spirit to the hybrid method we have just discussed, but instead of completely switching from one method to the other, it moves more

[15] Of course, we might get stuck on a local minimum if SSE is not unimodal.

gradually between the two. The iteration rule is given by

$$\beta_i \mapsto \beta_{i+1} := \beta_i - \tau_i\,((1-\lambda_i)J_i^T J_i + \lambda_i I)^{-1} J_i^T D_i, \qquad \lambda_i \in (0,1). \tag{B.3}$$

Note that (B.3) is a convex combination between the Gauss-Newton rule (B.1), when $\lambda_i = 0$, and the steepest descent rule (B.2), when $\lambda_i = 1$.[16]

The strategy of the method is to dynamically change λ_i in a smart fashion. Define $\mu_i := \lambda_i(1-\lambda_i)^{-1}$. We start from some initial μ_0 (`muInit = 1E-3`). If the SSE improves from one iteration to the next, we decrease μ by dividing it by some factor $m > 0$ (in the code, m is called `muFactor = 10`), thus moving closer to the Gauss-Newton method; if the SSE deteriorates we increase μ_i by multiplying it with m, thus moving into the direction of the steepest descent method.

We slightly vary this strategy because we have found that it works better. We decide about increasing or decreasing μ_i not by comparing the SSE from one iteration to the next. Instead, we decrease μ_i only if the current SSE is better than the best SSE that has been encountered so far during the whole process; we increase it otherwise.

Appendix C. Optimizing the step size

Optimization of the step size is quite important, particularly as long as we are far from the solution. Moreover, our root finding methods might suggest a step that would make the next bid function β_{i+1} locally decreasing or that might violate weak bid shading. Non-monotonicity in particular would cause trouble in further calculations by making the bid function not invertible. We proceed as follows: in each iteration step, we first determine the maxmimum size of τ that still guarantees weak monotonicity and bid shading; call this step size τ^*. Then we optimize by searching for a step τ in the interval $(0, \tau^*)$ which minimizes SSE.[17]

A good algorithm for finding the minimum along a line if the function is unimodal is the golden section search. You can select this option by setting `taumethod = 1` along with the required precision (`tauEps`).

Because we cannot be sure that SSE is unimodal in our search direction, we have also implemented a more expensive procedure (`taumethod = 0`), which we

[16]Note that $\nabla\text{SSE} = J^T D$. Here is where the division by 2 in the definition of SSE comes in.

[17]If this requirement does not put a constraint on the step size, we allow τ^* to be at most equal to `GRANDMAXTAU = 2.0`, so the bid function is changed by at most twice as much as the step size suggested by the chosen method.

call “brute force”: we compute SSE for step sizes $\tau \in \{\tau^* s S^{-1} : s = 1, \dots S - 1\}$, and we select the best one. S is called `tausteps` in the program. Note that we restrict $0 < \tau^* S^{-1} \le \tau \le \tau^*(S-1)S^{-1} < \tau^*$. $\tau = 0$ would immediately lead to an infinite loop; the algorithm would be stuck. $\tau = \tau^*$ is very likely to lead to a locally flat bid function. This makes inverting the bid function impossible, and also raises the possibility that τ^* would become zero in the next iteration step. The algorithm would again be stuck. Yet, the strategy of disallowing extreme step sizes is only partially successful. It happens on rare occasions that τ^* becomes smaller than machine precision. The algorithm stops in such a case.

If we are close to the solution, it can be that the smallest step is still too large so that the algorithm would overshoot. If the SSE is larger even with the smallest allowed step, we re-optimize the step size by evaluating the SSE at steps $\tau \in \{\tau^* s S^{-k} : s = 1, \dots S - 1\}$, with $k = 2$. We keep increasing k until the iteration leads to an improvement of the SSE. However, in no case do we allow τ to become smaller than $\tau^*\mathrm{SSE}/(g+1)$. This limit allows smaller steps once the error is small, but avoids getting stuck on very small steps as long as the error is still large.

Appendix D. Finding an initial guess

Appendix D.1. Starting from a linear bid function

Because we cannot hope to have global convergence, the choice of the initial bid function β_0 has to be done with care, and we should try to start from a point which is as close as possible to the solution. A simple idea is to start from the solution of a simpler problem which we can solve explicitly. We start from the solution of the restricted first-price auction in which only type I corruption is possible. One can show that

$$\beta(v) = \begin{cases} v - \int_0^v \frac{K(y)}{K(v)} dy = E_{Y_1 \sim K}[Y_1 \mid Y_1 < v], & \text{if } \alpha < 1, \\ v, & \text{if } \alpha = 1, \end{cases} \tag{D.1}$$

$$K(y) := F(y)^{\frac{n-1}{1-\alpha}}. \tag{D.2}$$

is the solution to this simpler problem. The rationale for starting from this point is the hope that adding the possibility of type II corruption does not alter the solution in a too extreme fashion. With the uniform distribution of valuations, the bid function of the restricted game is linear and given by

$$\beta_0(v) := \frac{n-1}{n-\alpha} v. \tag{D.3}$$

More generally, you can choose to start from any linear bid function by setting `initmethod = 0`. The slope of this initial bid function is set in the variable `slope`, which is by default set to the slope defined in (D.3).

Appendix D.2. Starting from a non-linear bid function

One could also start from a non-linear bid function. To do that, you need to write an alternative to the procedure `InitLinear`, maybe along these lines:

```
private static void InitPower()
{
    Console.WriteLine(">>> Filling in power bid function " +
                      "as an initial guess...");
    for (int j=1; j<=g; j++)
        betavec[j, 0] = slope * Math.Pow(val(j),exponent);
    compute();
}
```

You will have to declare the constant `exponent` in the beginning of the code somewhere (preferably in the neighborhood of the declaration of `slope`), and also change a part of the `Main()` method as follows,

```
switch (initmethod)
{
    case 0:
       InitLinear();
       break;
    case 1:
       RunGridSearch();
       OutputResult();
       break;
    case 2:
       InitPower();
       break;
}
```

Setting `initmethod = 2` would then select the power bid function as the starting point.

Appendix D.3. Grid search

An alternative to a fixed initial guess is to run a grid search (select `initmethod = 1`). In a grid search, we determine the RMSE of a large number of bid functions and select the one with the smallest RMSE as the starting point. More precisely, for the first dimension $i = 1$ (that is, $v_1 = 1/g$), we select Q equally spaced bids b_1 strictly between 0 and v_1 (Q is called `GridSearchSteps` in the code). For each of these bids, we then select Q bids for the second dimension, again equally spaced, and strictly between b_1 and v_2, in order to observe strictly monotonicity and bid shading. We do this through all g dimensions. This gives rise to Q^g bid functions. It is obvious that with a reasonably fine grid on the valuation space, say $g = 50$, this kind of grid search is not feasible because already with $Q = 2$ we would have to evaluate more than 10^{15} bid functions. So grid search is feasible only if we work with a coarse grid on the valuations, at least initially.

Appendix D.4. A variation: progressively finer grid method

Since it is important to start from an initial bid functions which is close to the final solution, and it is easier to find the root for a problem with less dimensions rather than more, one can follow a strategy of starting with a relatively coarse grid on the valuation space, and making the grid progressively finer. To consider an extreme example, suppose we start with $g = 1$. This amounts to searching for a solution within the class of linear functions. Apply a root finding method as described before. Then subdivide the grid on the valuation space, say by 4. So now $g = 4$ and we linearly interpolate the bid function for the new points in the valuation grid. We then again apply one of the root finding methods, and again subdivide the valuation space. We keep on doing this until we reach a reasonable fineness of the grid on the valuation space. This method has potentially two advantages. Firstly, it allows us to start with a very coarse grid, making the grid search a feasible choice for the initial guess. Secondly, after each subdivision of the valuation space, we start from a bid function which should be rather close to the solution for the new valuation grid.

In the code, the progressively finer grid method is controlled by two parameters: `NbSubdivisions` is the number of rounds in which the valuation grid is made progressively finer, and `SubdivisionFactor` is the factor by which it is made finer in each round. Thus, if we start with a grid of $g = 3$, which we subdivide twice with a factor of four, we end up with $3 \times 4^2 = 48$ points in the valuation space. Setting `NbSubdivisions = 0` turns off the progressively finer grid method.

Appendix E. The results

For most cases we do not use the progressively finer grid method (`NbSubdivisions = 0`). Instead we run a rather fine grid on the valuation space to begin with (`g = 200`). This rules out the grid search for the starting point (`initmethod = 0`), and we use a linear initial function following (D.3).

We find that the Levenberg-Marquardt method turns out to be the best choice for our problem. An iteration step is quite slow, but it makes up for this drawback with fast convergence. If there is an equilibrium, it finds it within a small number of iterations. The steepest descent method works also, although convergence is very slow and it does not seem to generate the same amount of precision.

For the optimization of the step size, we find that golden section search works well, so we use it throughout. The brute force method often works too, but is somewhat inferior.[18]

Finally, we stop the iteration if no improvement has occurred for sufficiently many (`keepiter = 20`) iterations, or after a maximum of `maxiter = 100` iterations. We then report the iteration step with the smallest SSE that we have detected so far.

Figure G.7 depicts the approximate equilibrium bid functions; Table G.4 reports the precision we achieve with these computations. For each n we have investigated, there is a borderline α at which the SSE increases dramatically when we increase α by just 0.0001 beyond this limit. These borderline αs are identified in the table.

In addition, we have found that convergence is tricky in some cases with relatively large values of α as well. The examples with $n = 5$ and $n = 10$, respectively, and $\alpha = 0.9$ did not converge well. We have found, however, that this is no indication of a lack of existence for large α, but is due to the imprecision imposed by too coarse a grid. It seems that the equilibrium bid function is not well enough captured in these cases by a piecewise linear function with only 200 vertices because it exhibits too large a curvature. In these two cases we have therefore used the progressively finer grid method and have doubled the grid in one step from 200 to 400 points (`NbSubdivisions = 1`, `SubdivisionFactor = 2`). This was sufficient to achieve low approximation errors.

[18] The parameters are `tauEps = 1E-2` for the golden section search and `tausteps = 10` for the brute force method. Brute force step size optimization often seems to work more harmoniously with the steepest descent root finding method, but since we do not use steepest descent, we have also no need for brute force step size optimization.

Appendix F. Testing for global optimality

Because the differential equation (A.1) is only a local condition for optimality we must check whether the solution of (A.1) indeed constitutes a global best reply against the bid function β. In other words, one needs to check if $v \in \arg\max U(v, x)$. Using the uniform distribution, the expected payoff function that is developed in the paper simplifies to

$$U(v, x) = \phi(x)^{n-1}(v - (1 - \alpha)\beta(x)) - \alpha(n - 1) \int_0^{\phi(x)} y^{n-2}\beta(y)dy$$
$$+ (\psi^*(v, x) - x)(n - 1)x^{n-2}(v - (1 - \alpha)x) - \alpha(n - 1)x^{n-2} \int_x^{\psi^*(v,x)} \beta(y)dy. \quad \text{(F.1)}$$

We compute $U(v, x)$ for each valuation in the grid $v \in \mathbb{G}$, and for all possible deviations $x \in \mathbb{G}$, using the numerical solutions we have found for β, ϕ, and ψ^*.

The numerical solutions are only approximate. Because we need to integrate numerically to compute (F.1), small errors may accumulate. Moreover, the payoff function is naturally flat close to the maximum. For these reasons, it is easily possible that occasionally $x^* := \arg\max_x U(v, x)$ deviates from v. As long as these deviations, $|x^* - v|$, are small and, more importantly, the implied maximum is close to the payoff one receives from bidding according to the payoff function β, i.e. $U(v, x^*) - U(v, v)$ is small, one should not suspect the numerical solution to be grossly wrong in the sense of mistaking a local optimum for a global one.

Figure G.8 shows a typical example of a plot of v against x^*. Ideally, the graph would be on the 45°-line. One can see some small deviations in particular at high values of v. However, the payoff function has very little curvature in this region, so that small deviations of the maximizer x^* from the true valuation v are to be expected. Altogether, these deviations are minor, and we find no evidence that large deviations from the computed bid functions can improve bidders' payoffs.

A typical example of a payoff function is depicted in the left panel of Figure G.9. There, one can see clearly that the equilibrium bid is the global maximizer for $v = 0.5$.

The worst example we found is the case of $n = 10$, $\alpha = 0.4013$, and $v = 0.485$, which is depicted in the right panel of Figure G.9. There, we can see that the payoff function has very little curvature close to the maximum. For this reason, a small numerical approximation error in the computation of the bid function β or in the computation of the payoff function U (which involves numeric integration) can lead to the conclusion that $x^* \neq v$. Yet, even in this case, the difference between the computed maxima, $U(v, x^*) - U(v, v)$, is small, so that this fact does not constitute evidence that v is not a global maximizer.

Appendix G. The program

The program is implemented in $C^\sharp$. Free IDEs for this language can be downloaded from `http://www.microsoft.com/express/Downloads/` or `http://www.icsharpcode.net/OpenSource/SD/`. A ZIP archive containing the program and computed results is available for download from our websites. The archive contains a file called `Main.cs`, which is the $C^\sharp$ source code, as well as plain text files which are copies of the output produced by the program. The naming conventions of these files are as follows: the file with name `x-0y.txt` contains the computation for $n = \texttt{x}$ and $\alpha = 0.\text{y}$. All parameters and choices of methods are specified at the beginning of the source code (Table G.5), so you do not need to modify the program itself when making your choices.

Figures of the Technical Appendix

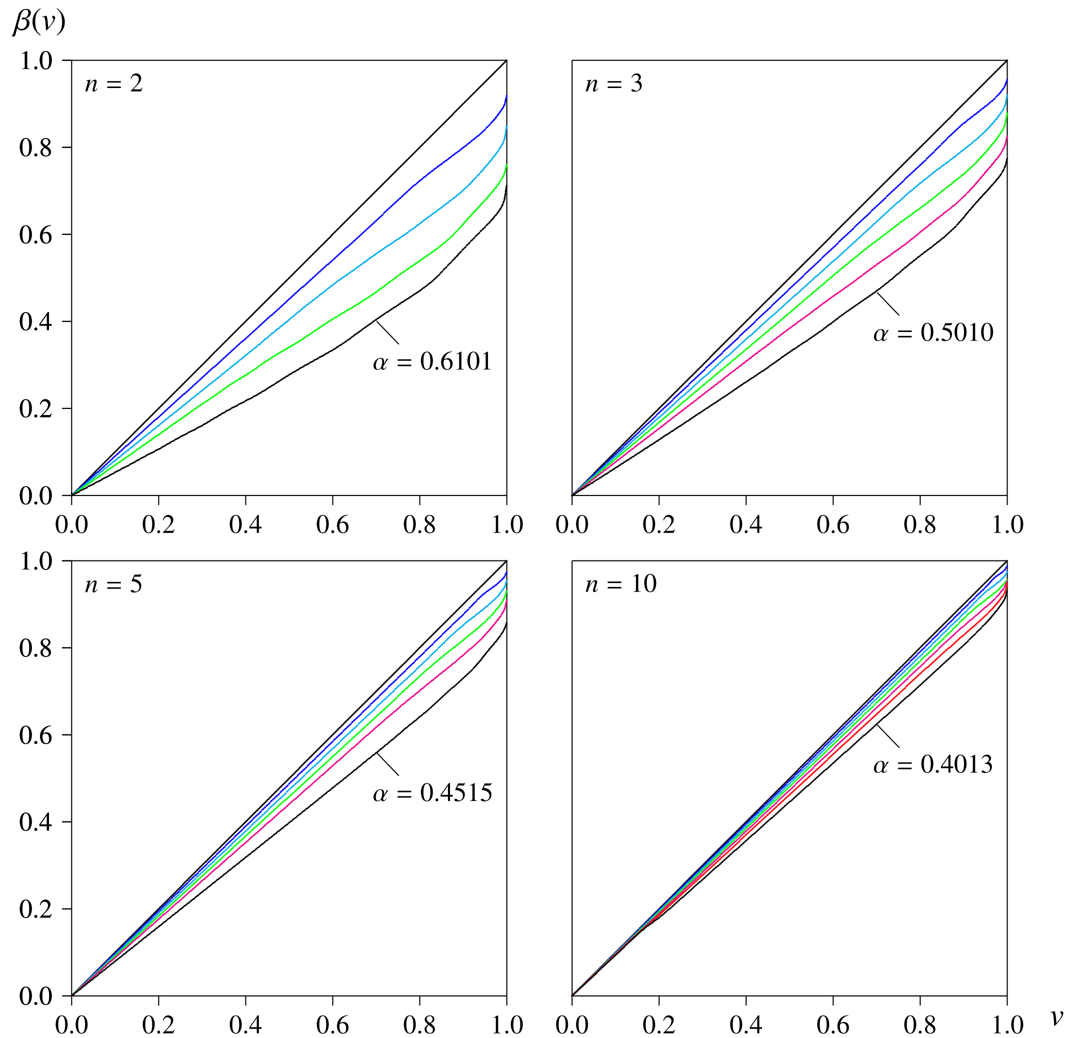

Figure G.7: Approximate equilibrium bid functions for the first-price auction with type I and type II corruption, for $n \in \{2, 3, 5, 10\}$ and $\alpha \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, and, for each n, for the smallest α for which we found a solution.

Table G.4: Precision of the computations.

$n = 2$	α	# iterations	SSE	RMSE	max absolute error	mean error
	0.9	32	4.40×10^{-22}	2.09×10^{-12}	2.04×10^{-11}	-1.43×10^{-16}
	0.8	34	2.23×10^{-17}	4.71×10^{-10}	3.94×10^{-9}	$+8.86 \times 10^{-14}$
	0.7	100	9.51×10^{-15}	9.73×10^{-9}	1.06×10^{-7}	$+6.23 \times 10^{-11}$
	0.6101	43	6.17×10^{-12}	2.48×10^{-7}	3.50×10^{-6}	-1.78×10^{-8}
	0.6100	25	9.76×10^{-8}	3.12×10^{-5}	2.96×10^{-4}	-7.41×10^{-6}
	0.6	29	3.04×10^{-8}	1.74×10^{-5}	1.84×10^{-4}	-2.06×10^{-6}
	0.5	61	5.51×10^{-2}	2.34×10^{-2}	6.33×10^{-2}	-1.95×10^{-2}

$n = 3$	α	# iterations	SSE	RMSE	max absolute error	mean error
	0.9	46	3.82×10^{-28}	1.95×10^{-15}	1.80×10^{-14}	$+6.49 \times 10^{-17}$
	0.8	31	4.39×10^{-25}	6.61×10^{-14}	3.72×10^{-13}	-1.97×10^{-16}
	0.7	50	9.73×10^{-24}	3.11×10^{-13}	2.77×10^{-12}	-2.28×10^{-16}
	0.6	37	2.15×10^{-24}	1.46×10^{-13}	1.38×10^{-12}	$+2.20 \times 10^{-15}$
	0.5010	50	3.98×10^{-16}	1.99×10^{-9}	2.82×10^{-8}	-1.37×10^{-10}
	0.5009	58	5.73×10^{-9}	7.55×10^{-6}	6.58×10^{-5}	$+3.13 \times 10^{-7}$
	0.5	25	1.00×10^{-1}	3.15×10^{-2}	2.22×10^{-1}	-6.96×10^{-3}

$n = 5$	α	# iterations	SSE	RMSE	max absolute error	mean error
*	0.9	19; 100	4.18×10^{-21}	4.57×10^{-12}	5.76×10^{-11}	-4.96×10^{-13}
	0.8	99	6.32×10^{-21}	7.93×10^{-12}	1.11×10^{-10}	$+4.07 \times 10^{-13}$
	0.7	98	5.85×10^{-20}	2.41×10^{-11}	2.59×10^{-10}	-2.41×10^{-12}
	0.6	16	8.84×10^{-17}	9.38×10^{-10}	7.90×10^{-9}	-1.15×10^{-10}
	0.5	99	7.64×10^{-19}	8.72×10^{-11}	8.17×10^{-10}	-5.11×10^{-12}
	0.4515	98	2.55×10^{-18}	1.59×10^{-10}	1.91×10^{-9}	-1.93×10^{-11}
	0.4514	18	1.24×10^{-5}	3.51×10^{-4}	4.47×10^{-3}	$+9.85 \times 10^{-6}$
	0.4	13	2.11×10^{-1}	4.59×10^{-2}	2.30×10^{-1}	-1.86×10^{-2}

$n = 10$	α	# iterations	SSE	RMSE	max absolute error	mean error
*	0.9	19; 100	2.23×10^{-19}	3.34×10^{-11}	2.04×10^{-10}	-7.84×10^{-12}
	0.8	44	9.75×10^{-18}	3.11×10^{-10}	1.74×10^{-9}	-7.96×10^{-11}
	0.7	98	6.07×10^{-18}	2.46×10^{-10}	1.47×10^{-9}	-5.77×10^{-11}
	0.6	100	4.15×10^{-17}	6.43×10^{-10}	3.78×10^{-9}	-1.55×10^{-10}
	0.5	98	1.25×10^{-16}	1.12×10^{-9}	6.38×10^{-9}	-2.97×10^{-10}
	0.4013	37	2.95×10^{-15}	5.42×10^{-9}	3.08×10^{-8}	-1.61×10^{-9}
	0.4012	43	8.74×10^{-10}	2.95×10^{-6}	2.93×10^{-5}	-4.56×10^{-8}
	0.4	100	2.17×10^{-5}	4.65×10^{-4}	2.09×10^{-3}	-2.44×10^{-4}

[*] In these two cases we use the *progressively finer grid method*: we first set $g = 200$ and then double the grid to $g = 400$. The two numbers in the iterations column refer to the two phases of the process.

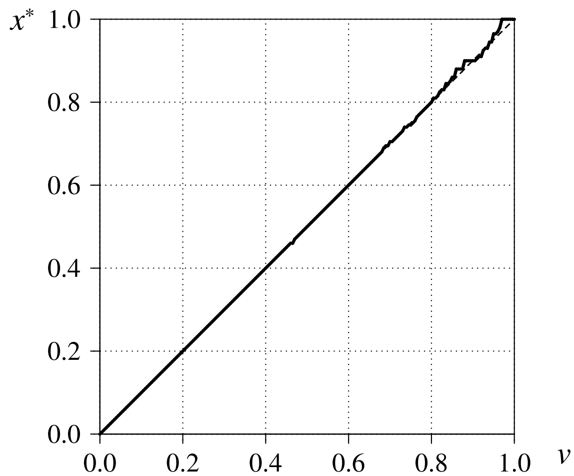

Figure G.8: Comparing v with x^*.

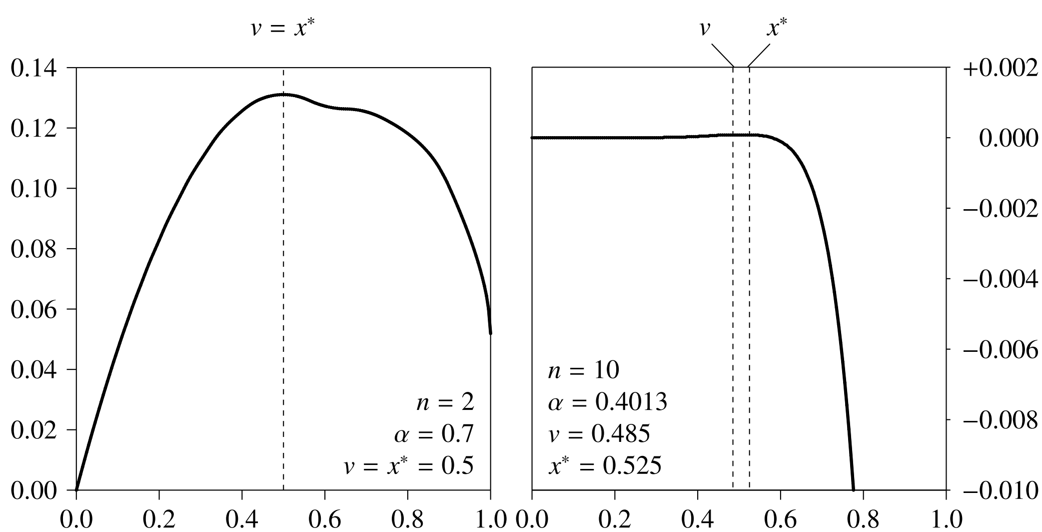

Figure G.9: Payoff function $U(v, x)$ as a function of x, for two examples.

Table G.5: Section of the source code in which all parameters are declared.

```
// ==========================================================================
// In this section, the parameters of the problem are given.
// You are free to change these parameters.

const byte n = 2;                  // # bidders (greater than or equal to 2)
const double alpha = 0.9;          // the bidder's share (between 0.0 and 1.0)

// ==========================================================================
// Next we define the parameters defining the discrete approximation, the methods
// for finding the root and the initial guess, and the stopping conditions.
// These settings should only be changed with caution.

// --- discrete valuation space ---------------------------------------------

static int g = 200;                     // initial # of points in the space of valuations
const int SubdivisionFactor = 4;        // factor by which valuation space is subdivided
const int NbSubdivisions = 0;           // number of times the space is subdivided

// "delta" for finite difference derivatives
const double diffDelta = 1E-10;         // this is the "delta" we use to compute the
                                        // derivatives
// --- inititial bid function -----------------------------------------------

const int initmethod = 0;               // choice of method for initial guess
                                        // initmethod = 0 : linear guess
                                        // initmethod = 1 : grid search

const double slope = (n-1)/(n-alpha);   // slope of linear initial guess

const int GridSearchSteps = 50;         // parameter for the grid search
                                        // GridSearchSteps = # values that are tried for
                                        // each point in the valuation grid
// --- step size optimization -----------------------------------------------

const int taumethod = 1;        // taumethod = 0: brute force
                                // taumethod = 1: golden section search

const int tausteps = 10;        // number of stepsizes we try out in each iteration step
                                // when using brute force
const double tauEps = 1E-2;     // required precision when using golden section search

const double GRANDMAXTAU = 2.0; // maximum step size under all circumstances
```


→ *continued on next page*

(Table G.5 continued)

```
// --- root finding ----------------------------------------------------------------

const int rootmethod = 3;       // choice of method for root finding
                                // rootmethod = 0 : steepest descent method
                                // rootmethod = 1 : Gauss-Newton method (inverse Jacobian)
                                // rootmethod = 2 : hybrid method
                                // rootmethod = 3 : Levenberg-Marquardt method

// switching rule for hybrid method
const double switchEps = 1E-12;
const double switchKeep = 5;

// coefficients for Levenberg-Marquardt method
const double initMu = 1E-3;             // initial mu
const double muFactor = 10;             // factor by which mu is multiplied or divided
const double maxMu = 1E+100;            // upper bound for mu
const double minMu = Double.Epsilon;    // lower bound for mu

// --- stopping rules --------------------------------------------------------------

const double eps = Double.Epsilon;  // stop iterating if SSE < eps
                                    // (Double.Epsilon is machine precision, so this
                                    // effectively turns off this stopping rule; you
                                    // can choose a larger value for eps)
const int keepiter = 20;            // stop iterating if no improvement in so many steps
const int maxiter = 100;            // stop after so many steps in any case

// ================================================================================
```