Investigating the Relative Performance of Auctions Using Approximated Cost Functions As Bids *

Brad Ross

December 8, 2016

1 Introduction

Often in the real world, situations arise in which some central authority must allocate a set of different types of resources that vary in quantity together to interested parties. For example, airports must assign different runway takeoff and landing time slots to various airlines, government communications agencies must divide rights to radio spectrum in different regions between potential broadcasters, and conservation agencies must divvy up limited rights to various fishery resources amongst competing commercial and recreational fishers, among many other applicable scenarios [1], auctions in which bidders compete for bundles of different types of items in varying quantities, are a promising type of mechanism for producing optimal solutions to these complex allocation problems and, as such, have recently become an area of intense study in economics, computer science, operations research, and other related areas.

Research into combinatorial auctions has remained active in large part because of their seemingly intractable complexity, which takes two primary forms. First is the complexity inherent in the process of evaluating the space of possible bids to discover the optimal allocation of goods to bidders, called the **Winner Determination Problem** [1]. One of the principal features of combinatorial auctions is that items must be purchased in bundles containing various quantities of the different goods being offered, meaning the number of possible allocations–and, by extension,

*I thank Professor Paul Milgrom of Stanford University for inspiring me to pursue writing this paper as my final project for his Stanford undergraduate course on market design, ECON136, and for giving me generous conceptual support throughout the process of conducting the investigation.

possible bids–increases exponentially in the number of items and their possible quantities. As a result, the Winner Determination Problem is extremely difficult to solve in the general case. In fact, the factorial-order growth in the number of possible optimal bids puts the Winner Determination Problem in a class of problems computer scientists call NP-Complete, which, in simplified terms, means no algorithms have been discovered for computing their solutions whose running times grow just polynomially (rather than exponentially or factorially, for example) in the size of their inputs [1]. Perhaps the most important unresolved areas of inquiry in computer science is whether or not polynomial-time solutions to these NP-Complete problems even exist at all. Thus, solving the Winner Determination Problem efficiently is inextricably tied to one of the most vexing barriers to algorithmic efficiency facing computer scientists today.

The other principal source of complexity is the difficulty in expressing bids drawn from the vast bid space described previously. To be able to participate in a combinatorial auction, bidders must submit bids to the auctioneer detailing how much they are willing to pay for various packages of goods. Of course, it would be quite impractical for bidders to specify how they would value every one of the possible bundles of items. However, complications in trying to reduce the number of required bids arise due to the fact that the different types of goods being sold might not be perfectly substitutable. As an extreme example, supposed two goods A and B are being sold in the auction. A bidder in this auction might not value getting two of good A as opposed to one of A and one of good B equally; she might only value an extra unit of A if she could also receive an extra unit of B, or she might require a few units of A to make obtaining any quantity of B valuable at all. Indeed, the possible relationships between goods have the potential to be quite intricate, making it difficult to design a uniform way to express them. In attempts to try to remedy this issue, researchers have developed a myriad different **bidding languages** to try to maximize the expressive power of bids while minimizing the quantity and complexity of the bids required to solve the Winner Determination Problem [1]. Unfortunately, it appears that no single language has been adopted widely in the literature as a satisfactory way of communicating preferences that is both versatile in its ability to capture complex relationships between goods and simple for bidders to use to report their bids.

While general solutions to the problems facing combinatorial auction designers have continued to elude researchers, progress has been made toward developing mechanisms to tackle specific classes of combinatorial auctions and approximate general solutions somewhat more efficiently.

Examples of such efforts are described in detail in the book *Combinatorial Auctions* edited by Crampton, Shoham, and Steinberg [1]. Inspired by these efforts to tame the complexity of both the Winner Determination Problem and bid expression for more specific combinatorial auction situations, I explore the possibility of using a drastically simplified bidding language to improve the computational and communicational efficiency of determining optimal allocations of goods in single-good, multi-unit reverse auctions reminiscent of common procurement scenarios. In particular, I study the effect on the optimality of allocations generated by a Vickrey-Clark-Groves-esque sealed-bid auction if bidders–called "suppliers" in this paper–express their possible bids as affine approximations of cost functions that exhibit increasing returns to scale. While this auction is not exactly combinatorial in nature since it only allocates a single type of good, examining the admittedly simplistic model developed in this paper hopefully sheds light on the costs and benefits of using approximated value functions to express bids, paving the way for future inquiry into the feasibility of such simplifications under more general combinatorial auction conditions in which multiple goods are being purchased. Although auctions involving multiple goods are not my main subject of inquiry, at several points during the paper I attempt to describe in general terms how the multi-unit, single-good model described could be extended to handle multiple types of goods.

2 Auction Models

2.1 Agents and the Original Mechanism

First, I introduce in more detail the auction models this paper will study. Let N be a set of n suppliers competing to sell varying quantities of a good to a buyer in a reverse auction. The buyer would like to purchase a quantity q of the good from these suppliers and do so in such a way as to minimize the cost to the suppliers, which depending on the pricing scheme used in the auction, could also minimize the cost to the buyer. Each supplier $i \in N$ (for $1 \leq i \leq n$) sells a nonnegative, real-valued quantity $x_i \in \mathbb{R}$ of the good to the buyer for a price p_i. The cost to supplier i of producing and selling a quantity x_i of the good is expressed as a function $f_i : x_i \in \mathbb{R} \to \mathbb{R}$. There are three primary restrictions I place on f_i, some by necessity and others for convenience. First, supplier i can only produce quantities of the good up to some maximum capacity $C_i \geq 0$, since most, if not all firms cannot produce arbitrarily large quantities of any good without being limited by some sort of constraint on their production capacities. Second, f_i must be non-convex and nondecreasing ($\forall x_{i1}, x_{i2} \in \mathbb{R}$, $\alpha \in [0,1]$, $f_i(\alpha x_{i1} + (1-\alpha)x_{i2}) \geq \alpha f_i(x_{i1}) + (1-\alpha) f_i(x_{i2})$ and $\forall x_{i1}, x_{i2}$ where $x_{i1} < x_{i2}$, $f_i(x_{i1}) \leq f_i(x_{i2})$) on the interval $x_i \in [0, C_i]$, reflecting a cost structure with increasing returns to scale and the impossibility of any marginal unit of the good sold having negative marginal cost. Third, in this paper, I restrict my attention to f_i in the family of polynomial functions of the form $f_i(x_i) = (-1)^{d_i-1}(x_i - C_i)^{d_i} + 2C_i^{d_i}$, where $d_i \in \mathbb{N}$ represents the degree of f_i. While this set of possible cost functions is by no means general, the set's specific functional form still presents ample opportunity for study and guarantees several convenient properties. First, note that functions of this form must always be non-convex on the interval $[0, C_i]$. Computing the intervals of non-convexity of a polynomial in the general case is not straightforward, and it is not even guaranteed that a given polynomial will have any non-convex intervals in the first quadrant at all, let alone one including 0. Thus, restricting f_i to these functions easily guarantees an interval of non-convexity in the desired location. Second, observe that so long as $C_i \geq 0$, f_i must have a nonnegative intercept representing supplier i's fixed cost of production. Proofs of these two facts are relatively straightforward, and thus are not presented here. Third, and perhaps most importantly, only polynomial functions can be used as objective functions in the optimization software I use to run simulations of the mechanisms presented below, so it would be impossible to study any others for the purposes of this paper. Finally, it is important to recognize that, if $x_i = 0$, then supplier i is not required to produce any of the good, and thus its fixed cost should not be incurred. As a result, the cost function f_i is best expressed in the following piecewise form:

$$f_i(x_i) = \begin{cases} (-1)^{d_i-1}(x_i - C_i)^{d_i} + 2C_i^{d_i} & x_i > 0 \\ 0 & x_i \leq 0 \end{cases}$$

The first auction mechanism I will study, called (perhaps unoriginally) the **original mechanism**, is one similar to a reverse Vickrey-Clark-Groves mechanism in which the goal of the buyer is to determine the vector of quantities of the good to purchase from each supplier $\boldsymbol{x} = \{x_i\}_{i=1}^n$ that minimizes the cost to suppliers of producing a total quantity of the good q, subject to suppliers' production capacities. We can express the buyer's objective more formally as follows:

$$\begin{aligned} \boldsymbol{x_o^*} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \quad & \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i = q \\ & 0 \leq x_i \leq C_i,\ i = 1, \ldots, n \end{aligned}$$

Because this paper focuses on the impact of approximated bids on the determination of optimal allocations, I will leave questions of optimal pricing aside. However, it is important to recognize they must be considered if one wishes to fully characterize the nature of this mechanism.

2.2 The Approximated Mechanism

Procurement scenarios that can be formulated in terms of the model above appear frequently in the real world. However, suppliers' cost functions are often complex enough that fully specifying them to the buyer would be quite cumbersome. While in the model above, cost functions can be easily reported as simply a degree d_i and a capacity C_i, real firms' cost functions take much more intricate forms. Another problem with a supplier revealing its true cost function to the buyer might be that doing so would jeopardize its competitive advantage in some fashion, making it disadvantageous to report costs truthfully. These complications are only exacerbated when the buyer attempts to source multiple different types of goods from suppliers at once, as the set of possible bundles of goods grows combinatorially as their number increases. Further, even if it were possible for suppliers to fully specify their cost functions, optimization of non-convex functions subject to even just linear constraints is computationally intractable in the general case, especially as the number of possible suppliers increases (more on this later). For these reasons and many others, it may not be feasible or desirable for suppliers to fully specify their cost functions to the buyer.

While eliminating questions of strategy in these scenarios will not be touched upon in this paper, one possible way to simplify cost function specification is to have supplier i attempt to approximate its cost function with another function g_i. Perhaps the simplest function a supplier i can use to do so is an affine function of the form $g_i(x_i) = a_i x_i + k_i$ where a_i represents what amounts to a roughly constant marginal cost of production for each unit of the good and k_i specifies the supplier's fixed cost of production. Suppose (naively) that every supplier i finds it within its interest to report the closest possible affine cost function g_i to its true cost function f_i. Then a natural choice of approximated cost function is to specify a_i and k_i such that the total distance between the two functions on $[0, C_i]$ is minimized, which can be determined as follows:

$$a_i^*, k_i^* = \underset{a_i, k_i}{\text{argmin}} \int_0^{C_i} (f_i(x) - (a_i x + k_i))^2 dx$$

This approximation is a continuous reformulation of the popular least-squares method for finding the best function to fit a given dataset. I label the cost function of supplier i approximated in this way as g_i^{LS}. Again, if $x_i = 0$, then supplier i is not required to produce any of the good, and thus its fixed cost should not be incurred. Thus, the cost function g_i^{LS} is best expressed in the following piecewise form:

$$g_i^{LS}(x_i) = \begin{cases} a_i^* x_i + k_i^* & x_i > 0 \\ 0 & x_i \leq 0 \end{cases}$$

Assuming that all firms choose to approximate their cost functions in this way, the second type of mechanism I study, called the **approximated mechanism**, is similar to the original mechanism, except suppliers' approximated cost functions g_i^{LS} are used to determine the optimal vector $\boldsymbol{x}$ instead of using suppliers' true cost functions. Thus, the optimization problem to be solved in order to determine the efficient set of quantities can be expressed as follows:

$$\begin{aligned} \boldsymbol{x_a^*} = \underset{\boldsymbol{x}}{\text{argmin}} \quad & \sum_{i=1}^{n} g_i^{LS}(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i = q \\ & 0 \leq x_i \leq C_i, \ i = 1, \dots, n \end{aligned}$$

Note that the actual cost incurred by any supplier i as a result of its production of the quantity x_i of the good is still computed using its true cost function f_i; its approximated cost function g_i^{LS} is only used to determine x_i.

2.3 A Generalization to Multiple Goods

I now sketch briefly a possible generalization of this model to accommodate auctions in which multiple goods are to be sold together. Let M be the set of m different types of goods being sold. The buyer in this scenario wants to purchase a quantity q_j of good j for $1 \leq j \leq m$ from these suppliers. Each supplier $i \in N$ sells the buyer a quantity x_{ij} of good j, where $1 \leq i \leq n$ and

$1 \leq j \leq m$. The cost to supplier i of producing some vector of quantities of goods $\boldsymbol{x_i} = \{x_ij\}_{j=1}^m$ can be expressed as a function $f_i(\boldsymbol{x_i}) : \mathbb{R}^m \to \mathbb{R}$. Supplier i is constrained in its production by some level set of a capacity function $c_i(\boldsymbol{x_i}) = C_i$ for some $c_i : \mathbb{R}^m \to \mathbb{R}$, which demarcates the boundary beyond which it is not feasible for supplier i to produce; thus, supplier is constrained in its production by $\boldsymbol{x_i} \geq \mathbf{0}$ (with the inequality being defined component-wise) and $c_i(\boldsymbol{x_i}) \leq C_i$. Further, f_i must concave, i.e. for all vectors $\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}} \in \mathbb{R}^m$ and $\alpha \in [0,1]$, $f_i(\alpha \boldsymbol{x_{i1}} + (1-\alpha)\boldsymbol{x_{i2}}) \geq f_i(\alpha \boldsymbol{x_{i1}}) + (1-\alpha) f_i(\boldsymbol{x_{i2}})$. f_i must also be nondecreasing, meaning for all vectors $\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}} \in \mathbb{R}^m$ where $\boldsymbol{x_{i1}} < \boldsymbol{x_{i2}}$ (defined component-wise), $f_i(\boldsymbol{x_{i1}}) \leq f_i(\boldsymbol{x_{i2}})$. Instead of approximating supplier i's cost function using just a slope and an intercept, the multi-good approximated mechanism must choose the affine function in m dimensions that minimizes the distance between it and the supplier's true cost function in the compact space defined by the constraints on production above. In addition, the level set of the cost function c_i must be approximated by some other affine function, likely estimated using a similar technique to the approximation of the supplier's true cost function. Unfortunately, this approximation technique eliminates the possibility of expressing more complex relationships between goods since affine functions are limited to expressing linear relationships in value between bundles of goods (i.e. goods that are substitutes); however, for scenarios in which rich relationships aren't as prevalent, these approximations may be sufficient.

While this paper does not dwell further on this generalization of the models presented in the previous sections to multiple goods, it is quite feasible to adapt the simulations presented in the next section to test the performance of the approximated generalized model as well.

3 A Mechanism Evaluation Framework

Having defined the two auctions I study in this paper, the original mechanism and the approximated mechanism, I now proceed to investigate whether the approximated mechanism yields outcomes nearly as efficient as those determined by the original mechanism. In more formal terms, if $\boldsymbol{x_o^*}$ is the minimum cost allocation yielded by the original mechanism and $\boldsymbol{x_a^*}$ is the minimum cost allocation yielded by the approximated mechanism, I determine whether or not the true cost of $\boldsymbol{x_a^*}$, $\sum_{i=1}^n f_i(x_{ai}^*)$, is sufficiently close to the true cost of $\boldsymbol{x_o^*}$, $\sum_{i=1}^n f_i(x_{oi}^*)$. If so, then it is possible to achieve efficient outcomes with much less communicational and computational overhead than using the original mechanism directly. To probe the relative performance of these two mechanisms, I simulate both mechanisms in numerous randomly generated contexts and measure the difference between the allocations outputted by the two mechanisms. Further, I try to characterize how several parameters varied between simulated auction scenarios contribute to any discrepancies in the costs of allocations I observe in the data.

3.1 Solving the Winner Determination Problem

Of course, in order to simulate the two mechanisms, it is important to be able to compute the optimal allocations the two mechanisms would provide. To translate the problem of determining $\boldsymbol{x}$ into a form more conducive to computation, the optimization problems for the original and approximated mechanisms can be expressed instead as Mixed-Integer Programs, or MIPs. MIPs are similar to traditional Linear and Non-Linear Programs except variables can either be continuous or restricted to integer values. While solving MIPs is an NP-Hard problem, research into algorithmic techniques like branch-and-bound procedures have made solving MIPs much more feasible in specific cases, especially in scenarios like the simulations I will describe below in which the number of variables is not overwhelmingly large, integer variables are restricted to binary values, objective functions are limited to polynomials, and constraints are limited to linear functions.

3.1.1 The Original Mechanism as an MIP

First, I formulate the original mechanism as a Mixed-Integer Nonlinear Program, or MINLP. To start, the MINLP must have n variables corresponding to each supplier i's quantity produced, x_i. Per the model specification above, each of these variables is constrained by two inequalities: $x_i \geq 0$ and $x_i \leq C_i$. Next, it is necessary to somehow specify which suppliers are supplying at all, in which case their fixed costs should be incurred. To do so, n binary variables $b_1, \ldots, b_n$ can also be defined, with $b_i = 1$ meaning supplier i is supplying a nonzero quantity of the good, and $b_i = 0$ indicating supplier i is not supplying any of the good. To ensure that $b_i = 1$ always holds as long as x_i is nonzero, I add the constraint $x_i \leq 2C_i b_i$; note that if x_i is nonzero and feasible, i.e. $x_i \leq C_i$, then the inequality holds so long as $x_i > 0$ and $b_i = 1$ or $x_i = 0$ and $b_i = 0$, as required. To capture the conditional payment of supplier i's fixed cost, I express $f_i(x_i)$ as the following function:

$$f_i(x_i, b_i) = (-1)^{d_i - 1}(x_i - C_i)^{d_i} + C_i^{d_i} + C_i^{d_i} b_i$$

Note that if $x_i = 0$ and $b_i = 0$, $f_i(x_i, b_i) = (-1)^{d_i-1}(-C_i)^{d_i} + C_i^{d_i} = -C_i^{d_i} + C_i^{d_i} = 0$, as desired. While there is no constraint dictating that if $x_i = 0$, $b_i = 0$ must be true as well, the cost to supplier i can always be decreased by setting $b_i = 0$ if $x_i = 0$. Thus, if $x_i = 0$, any optimization technique should pick $b_i = 0$ over $b_i = 1$ to minimize the cost of the optimal allocation. Note that although all of the constraints in this MINLP are linear, the objective function may not be, which is why the problem is classified as a MINLP. Having redefined the constraints and cost functions as such, the optimization problem to be solved as part of the original mechanism the can be rewritten as the following MINLP with $2n$ variables and $3n + 1$ constraints:

$$
\begin{aligned}
\boldsymbol{x_o^*} = \underset{\boldsymbol{x},\boldsymbol{b}}{\text{argmin}} \quad & \sum_{i=1}^{n} f_i(x_i, b_i) \\
\text{subject to} \quad & \sum_{i=1}^{n} x_i = q \\
& 0 \leq x_i \leq C_i, \ i = 1, \ldots, n \\
& x_i \leq 2C_i b_i, \ i = 1, \ldots, n \\
& b_i \in \{0, 1\}, \ i = 1, \ldots, n
\end{aligned}
$$

All that is required to generalize this MIP to solve the Winner Determination Problem for reverse auctions involving multiple goods is to replace the n linear constraints $x_i \leq C_i$ with n (potentially nonlinear) constraints $c_i(\boldsymbol{x_i}) \leq C_i$, and to replace the n linear constraints $x_i \leq 2C_i b_i$ with n constraints $c_i(\boldsymbol{x_i}) \leq 2C_i b_i$, where $c_i(\boldsymbol{x_i}) = C_i$ is the level set of the capacity function described in section 2.3. If there are m different goods in this scenario, the MIP to solve involves $nm + n = n(m + 1)$ variables and $nm + 2n = n(m + 2)$ constraints.

3.1.2 The Approximated Mechanism as an MILP

Because the approximated mechanism's objective function is expressed as the sum of affine approximated cost functions, the optimization problem to be solved falls under a special class of MIPs, called Mixed-Integer Linear Programs (or MILPs) whose solutions are much less computationally expensive to compute than general MIPs. As can be easily demonstrated, the variables and constraints involved in the MILP for the approximated mechanism are exactly the same as those required for the MIP formulation of the original mechanism. The only difference between the two mechanisms is that supplier i's cost function $g_i^{LS}(x_i)$ is instead re-expressed as the following:

$$g_i^{LS}(x_i) = a_i^* x_i + k_i^* b_i$$

Then, similarly to the MIP equivalent to the original mechanism, the MILP equivalent to the optimization problem that determines the allocation generated by the approximated mechanism can be described as follows:

$$
\begin{aligned}
\boldsymbol{x}_{\boldsymbol{a}}^{*} = \underset{\boldsymbol{x},\boldsymbol{b}}{\operatorname{argmin}} \quad & \sum_{i=1}^{n} g_i^{LS}(x_i, b_i) \\
\text{subject to} \quad & \sum_{i=1}^{n} x_i = q \\
& 0 \leq x_i \leq C_i,\ i = 1, \ldots, n \\
& x_i \leq 2C_i b_i,\ i = 1, \ldots, n \\
& b_i \in \{0, 1\},\ i = 1, \ldots, n
\end{aligned}
$$

3.2 Computational Constraints on Simulations

I now briefly address some of the computational constraints facing the comparison of the original and approximated mechanisms by simulation. Most commercial optimization libraries like Gurobi and CPLEX are only built to solve a few types of MIPs (at least according to their documentation), specifically MIPs with linear and/or quadratic objective functions and linear and/or quadratic constraints. Thus, to run my simulations I rely on an open-source MIP-solving library called SCIP, developed by researchers at The Zuse Institute Berlin [2]. SCIP is able to solve MIPs with general polynomial objective functions and constraints, which is why I am restricted to being able to test only polynomial cost functions [2]. However, SCIP runs almost nine times slower than state-of-the-art commercial solvers, limiting my capacity to run large numbers of simulations. In addition, because solving MIPs with even polynomial objective functions is computationally taxing, simulating more than several hundred thousand auctions with more than 10 suppliers each (and thus more than 30 variables) takes more than a day on a standard laptop. Thus, as I explain below, I am limited in my capacity to increase auction sizes and widen the scope of my inquiry beyond interactions between three changing simulation parameters. In the future, it would be desirable to attempt simulating even more complex auction scenarios with more varying parameters and over a longer time period on a more powerful computer.

3.3 Simulation Specifications

Having defined how to solve the optimization problems required to determine the optimal allocations given by the original and approximated mechanisms, I now characterize the simulations I run to evaluate the two mechanisms. The atomic component of my simulations is the randomly generated cost function. Each random cost function f_i is determined by two random variables: the degree $d_i \sim Binomial(\frac{\mu_d}{p_d}, p_d)$ and the capacity $C_i \sim \mathcal{N}(\mu_C, \sigma_C^2)$. Here, μ_d is the desired mean degree of randomly generated cost functions, p_d is the probability parameter of the binomial distribution of these cost function degrees, μ_C is the desired mean capacity of a supplier, and σ_C is the desired standard deviation of the suppliers' capacities. f_i is then computed using the formula specified in section 3.1.1.

To compute the approximated cost function of f_i, g_i^{LS}, I create a matrix consisting of a sufficiently large number of $(x_i, f_i(x_i))$ pairs with x_i distributed uniformly along the interval $[0, C_i]$, and then compute a best-fit line g_i^{LS} with coefficients a_i^* and k_i^* using the least-squares method for solving overdetermined systems of equations.

To simulate a random auction j, I generate n_j suppliers for some specified number of suppliers n_j with random cost functions using the procedure specified above and compute their total capacity, $C_j = \sum_{i=1}^{n} C_i$. I then set the quota q_j of auction j to be $\gamma_j C_j$ some constant quota factor γ_j, where $0 < \gamma_j \leq 1$; as such, the quota to be purchased q_j can never be larger than the total capacity C_j, guaranteeing that the entire quota is procurable from the participating suppliers. I then determine the optimal allocations given by the original and approximated mechanisms for the given quota and supplier cost functions by solving the MIPs specified in sections 3.1.1 and 3.1.2. Finally, I compute the percent error r_j between the actual true minimum cost $c_o^* = \sum_{i=1}^{n} f_i(x_{oi}^*)$ discovered by the original mechanism and the approximate true minimum cost $c_a^* = \sum_{i=1}^{n} f_i(x_{ai}^*)$ discovered by the approximated mechanism, $r_j = \mid \frac{c_a^* - c_o^*}{c_o^*} \mid$.

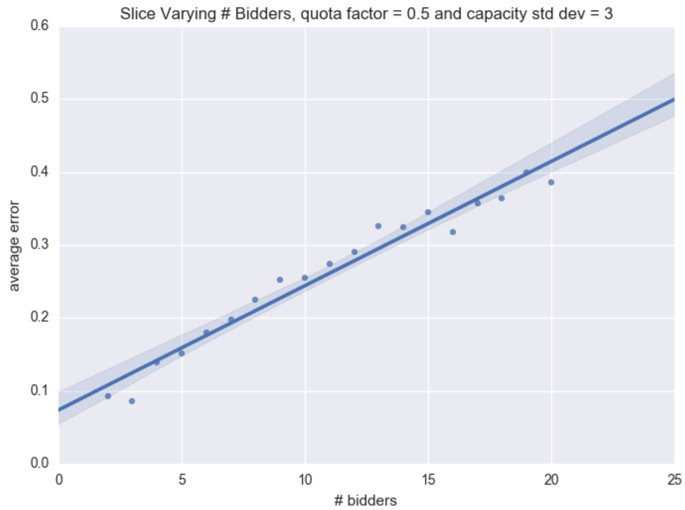
Due to both temporal and computational constraints, in this paper I focus my analysis on how three of the parameters defined above impact the relative performance of the two mechanisms: the number of suppliers n_j, the percent of maximum capacity used as the buyer's desired quota γ_j, and the variance of suppliers' capacities, σ_C. Intuitively, increasing the number of suppliers should increase the complexity of the auction, which could have an adverse effect on relative performance. The percentage of the maximum capacity of the market being purchased could also have an effect on the relative performance of the two mechanisms. Finally, varying the degree of heterogeneity in the market by changing the variance in suppliers' capacities is likely to have some impact on relative performance; an auction with many similar suppliers is likely to perform differently than an auction with several larger suppliers and many smaller ones. To analyze the relative performance of the two mechanisms in these varying scenarios while maintaining reasonable computation time, I fix the mean degree of all randomly generated cost functions μ_d at 3, the probability parameter of the degree's binomial distribution p_d at $\frac{1}{3}$, and the mean capacity μ_C at 10. I then uniformly vary the number of suppliers n_j by integer increments between 2 and 10, the quota factor γ_j by increments of 0.016 between 0.1 and 0.9, and the standard deviation of randomly generated capacities σ_C by increments of 0.225 between 0.5 and 5, and run 30 randomly generated auctions using each possible combination of the values of these three parameters. A table below summarizes this parameterization:

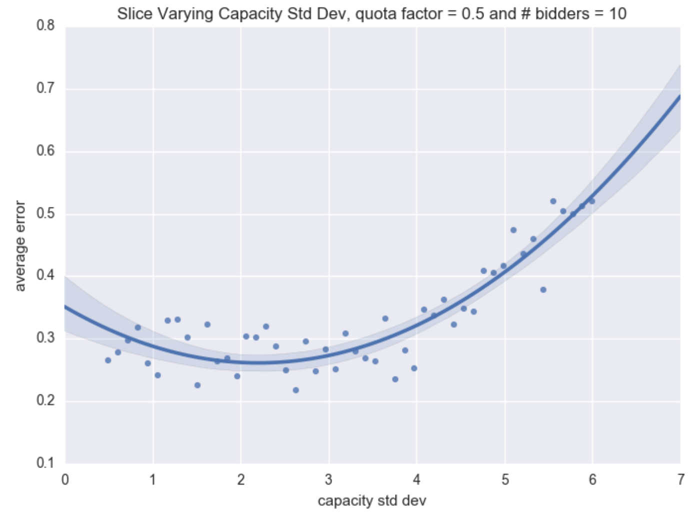
Variable	*Description*	*Values*
μ_d	the mean degree of all randomly generated cost functions	fixed at 3
p_d	the probability parameter of the degree's binomial distribution	fixed at $\frac{1}{3}$
μ_C	the mean capacity of all randomly generated cost functions	fixed at 10
n_j	the number of bidders in auction j	$\{2, \ldots, 10\}$
γ_j	the percent of suppliers' total capacity to purchase	$\{0.1, 0.116, \ldots, 0.9\}$
σ_C	the standard deviation of all randomly generated capacities	$\{0.5, 0.725, \ldots, 5\}$

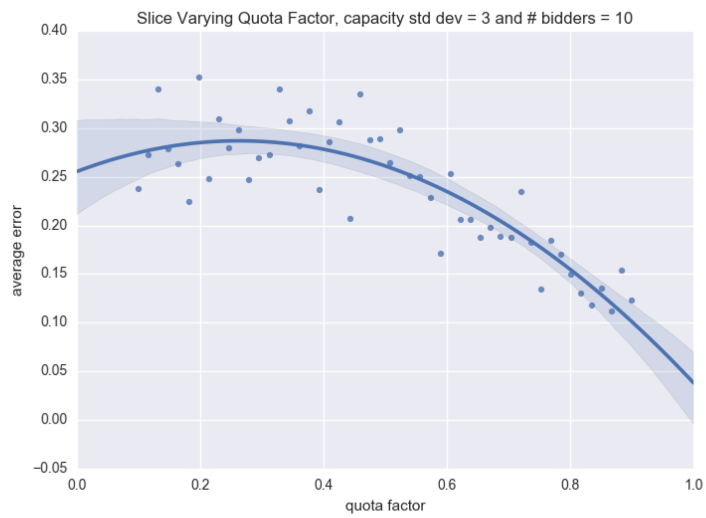
4 Simulation Results

Running the above simulations produces a dataset of $270,000$ random auctions produced with varying values of n_j, γ_j, and σ_C. Due to the variance between r_j values for a fixed set of parameter values $(n_j, \gamma_j, \sigma_C)$ created by the randomization inherent in the simulation process, I conduct all of my analysis on the mean value of r_j over the 30 auctions simulated with each unique combination of $(n_j, \gamma_j, \sigma_C)$, $\hat{\mu_r}$. While 30 auctions is a somewhat small amount of replication, for this number of duplicate simulations, the Law of Large Numbers guarantees that $\hat{\mu_r}$ is reasonably close to the true mean μ_r. Of course, with more computing power, it might be worth replicating these results with more simulations per set of parameter values to get even more accurate estimates.

To provide some visualization of the data, three scatter plots of $\hat{\mu_r}$ for various one-dimensional slices of the data along with guiding linear and quadratic trend lines are provided below:







Unfortunately, as is clear even in the slices of the data presented above, $\hat{\mu_j}$ is almost never small; in the graphs above, it never drops below 10%. In fact, $\hat{\mu_j}$ never falls below 20% except for when the number of suppliers is small or the quota factor is more than 80% of maximum capacity. To investigate the relationship between these parameters and the error more formally, I fit the following regression of $\hat{\mu_r}$ on n_j, γ_j, and σ_{Cj}:

$$\hat{\mu_{r}}_j = \beta_n n_j + \beta_{\gamma 1}\gamma_j + \beta_{\gamma 2}\gamma_j^2 + \beta_{\sigma 1}\sigma_{Cj} + \beta_{\sigma 2}\sigma_{Cj}^2 + \beta_0 + \epsilon_j \tag{4.1}$$

In addition, I fit another regression model using the same specification as above but with standardized exogenous and endogenous input values to compare the relative impact of different variables on the average error. A table summarizing the data and partial residual plots for each of the three variables are provided in Appendix sections 6.2 and 6.3, and the results of the regressions are summarized in the table below:

	Original Model		Standardized Model	
Variable	Coefficient	Std. Error	Coefficient	Std. Error
Intercept	-0.0776***	0.004	0.2035***	0.012
n_j	0.0258***	0.000	0.6800***	0.006
γ_j	0.4974***	0.012	-0.2033***	0.006
γ_j^2	-0.5821***	0.012	-0.3293***	0.007
σ_C	-0.0194***	0.002	0.2369***	0.006
σ_C^2	0.0066***	0.000	0.1257***	0.007
R^2	0.659		0.659	
F-statistic	3477		3477	
Note: *** indicates $p < 0.001$				

To determine which variables have the most impact on $\hat{\mu_r}$, I use the standardized regression model to compare the relative impact of the same small variation Δ in the values of two standardized variables. To do so, observe that for any two standardized variables α_1 and α_2, the changes in $\hat{\mu_r}$ caused by perturbing α_1 and α_2 by Δ (for sufficiently small Δ) are $\frac{\partial \hat{\mu_r}}{\partial \alpha_1}\Delta$ and $\frac{\partial \hat{\mu_r}}{\partial \alpha_2}\Delta$, respectively. Then, the ratio of these changes, $\mid \frac{\partial \hat{\mu_r}}{\partial \alpha_1}\Delta \cdot (\frac{\partial \hat{\mu_r}}{\partial \alpha_2}\Delta)^{-1} \mid = \mid \frac{\partial \hat{\mu_r}}{\partial \alpha_1}(\frac{\partial \hat{\mu_r}}{\partial \alpha_2})^{-1} \mid$, will be greater than 1 if the impact of α_1 is larger, and less than 1 if the impact of α_2 is larger. This direct comparison in rates of change for the same peturbation Δ is possible because α_1 and α_2 are standardized to the same units, meaning the rates of change are both expressed as standard deviations of $\hat{\mu_r}$ per standard deviation change in either of the predictors, α_1 or α_2. Note that because $\hat{\mu_r}$ is quadratic in γ_j and σ_C, the ratios of the derivatives depend on the values of γ_j and σ_C at which the peturbation by Δ occurs. For brevity, I will only compare the impacts of $\hat{\mu_r}$ and γ_j and $\hat{\mu_r}$ and σ_C, since comparing the relative impacts of γ_j and σ_C would involve nontrivial analysis of a multivariate function with absolute values. Comparing these pair of variables using this technique, I observe that n_j has a larger impact on $\hat{\mu_r}$ than γ_j when $\mid \frac{\partial \hat{\mu_r}}{\partial \gamma_j}(\frac{\partial \hat{\mu_r}}{\partial n_j})^{-1} \mid = \mid \frac{-0.2033 + 2 \cdot -0.3293\gamma_j}{0.6800} \mid < 1$, which occurs for all standardized γ_j $Z_{\gamma_j} \in [-1.34, 0.724]$, or $\gamma_j \in [0.184, .671]$. Using similar logic, I determine that n_j has a larger impact on $\hat{\mu_r}$ than σ_C for all standardized σ_C $Z_{\sigma_C} \in [-3.65, 1.76]$, or $\sigma_C \in [-2.22, 5.15]$ which, because σ_C must be greater than 0, is equivalent to the interval $[0, 5.15]$. The statistics used to generate these intervals, (i.e. means and standard deviations of γ_j and σ_C), can be found in the table in Appendix section 6.2. Thus, except for scenarios with tremendous variance in the capacities of different suppliers, the number of suppliers will have a larger impact on the average error, while for auctions in which the quantity being purchased is either a somewhat small or somewhat large percentage of maximum capacity, the number of bidders will have a larger impact on the average error.

In absolute terms, the results presented above are much more sobering. Indeed, $\hat{\mu_r}$ is almost never sufficiently small (e.g. less than 5%) for many realistic sets of parameters, such as auctions involving more than five suppliers whose sum of maximum capacities is not approached by the quantity being purchased. As the data show, while average error is relatively low for auctions involving small numbers of suppliers (really only two or three), small variation in capacity between suppliers, and a large quota factor, for more realistic auction scenarios with larger numbers of bidders, larger variance in the capacities of suppliers, and a potentially small quota factor, the average error incurred by the approximated mechanism is likely intractable. As a result, given the large discrepancies between the costs generated by the original mechanism and the approximated mechanism, and in particular the differences observed in more complex auction settings, it seems unlikely that using the approximated mechanism in place of the original mechanism will yield desirable outcomes.

5 Conclusion

As explained above, the model and accompanying simulation data presented in this paper demonstrate that, in its current formulation, the approximated mechanism does not perform nearly as well as the original mechanism in most scenarios. Of course, situations in which there are a small number of bidders, the quantity being purchased is close to the total production capacity of suppliers participating in the auction, and there is a moderate amount of variance in individual suppliers' production capacities could be ripe for employing the approximated mechanism in place of the original mechanism. While the communicational benefits of the simplified bids used in the approximated mechanism would be realized in circumstances like these, there would be little opportunity to leverage the potential computational advantages of using the approximated mechanism over the original mechanism.

For the remainder of this (rather brief) conclusion, I describe several areas of potential future inquiry, some of which were already touched on in this paper. The first opportunity is examining the relative performance of the approximated mechanism in true combinatorial scenarios in which several different goods are purchased together by the buyer in multi-unit bundles, as described in section 2.3. Given that the number of suppliers, and, by extension the number of variables have the largest adverse effect on approximated mechanism performance, I posit that increasing the number of variables further by adding more types of goods would hamper the effectiveness of the mechanism still more; however, the only way to effectively test this hypothesis is through additional research. A second area of investigation could be to tweak the nature of the simulations I run in this paper; perhaps I defined my parameterization naively in some way, and a better approach to randomizing auctions might be in order. Third, simulating the approximated mechanism on a more powerful computer with different optimization software and larger quantities of more sophisticated simulations might yield new insights into the relative performance of the approximated mechanism. Finally, this paper does not touch at all on potential pricing rules for and strategic analysis of the approximated mechanism. If it can be shown that the approximated mechanism does in fact yield desirably efficient allocations in scenarios different than the ones examined in this paper, then these aspects of the approximated mechanism's design must be considered. It may be that although the approximated mechanism does perform as desired in some cases, it is ripe for being taken advantage of by untruthful and/or colluding suppliers. Unfortunately, this paper cannot address all of these potential concerns in addition to the existing qualms it raises about the approximated mechanism. However, before discounting this mechanism completely, it is important to devote more serious effort to conclusively validating or invalidating its potential.

References

[1] P. Crampton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. MIT Press, Cambridge, Massachusetts, 2006.

[2] G. Gamrath, T. Fischer, T. Gally, A. M. Gleixner, G. Hendel, T. Koch, S. J. Maher, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, S. Schenker, R. Schwarz, F. Serrano, Y. Shinano, S. Vigerske, D. Weninger, M. Winkler, J. T. Witt, and J. Witzig. The scip optimization suite 3.2. Technical Report 15-60, ZIB, Takustr.7, 14195 Berlin, 2016.

6 Appendix

6.1 Paper Code

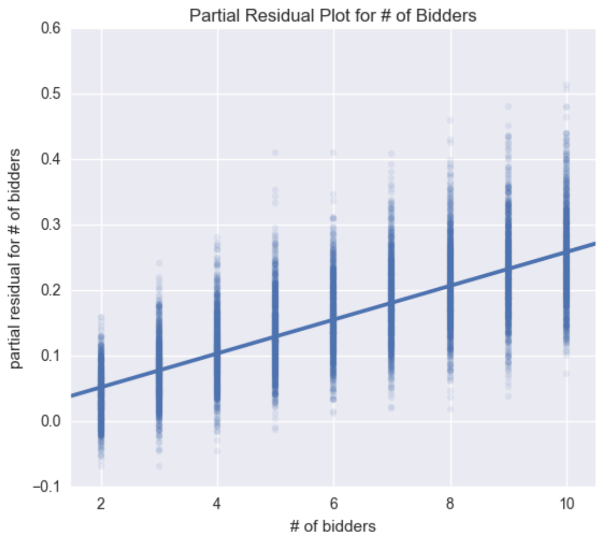
The code used to simulate the original and approximated mechanisms and conduct analyses of the data can be found on my Github account at https://github.com/brad-ross-35/approximated-auctions. The file *auction.py* contains the code used to run a single auction for a given quota and set of suppliers and using both the original and approximated mechanisms. The file *simulation.py* contains the code used to repeatedly simulate auctions using randomly generated parameters. Finally, the file *analysis.py* contains the code used to run analyses on the data and generate the plots presented in this paper.

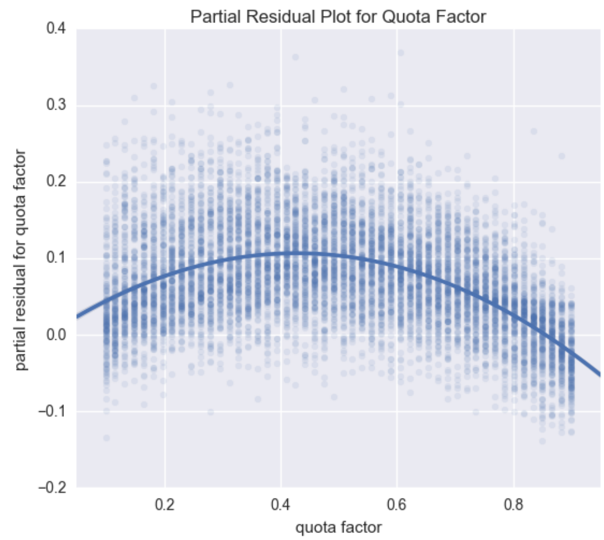
6.2 Simulation Data Summary

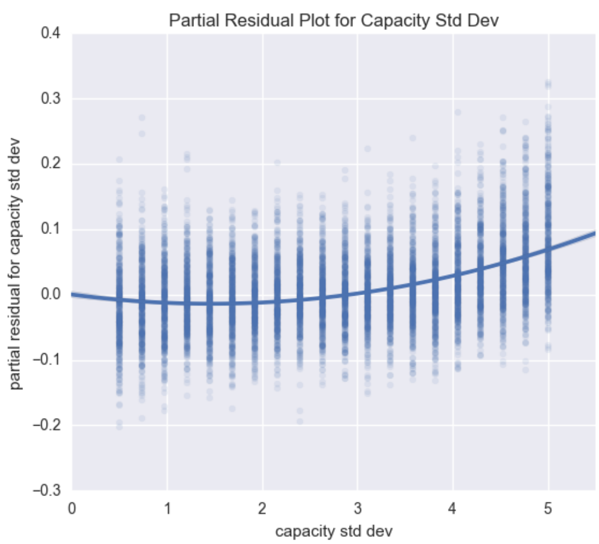
Variable	*Mean*	*Std. Dev*	*Min*	*Max*
number of bidders n_j	6	2.582	2	10
quota factor γ_j	0.5	0.2356	0.1	0.9
capacity std. dev. σ_C	2.75	1.366	0.5	5
average error $\hat{\mu_r}$	0.157	0.0981	0.0	0.545

6.3 Partial Residual Plots for Regression Specified by Equation 4.1

A partial residual plot for a given predictor x_i is a scatter plot of the residual r_j for each data point j plus the x_i terms of the fitted regression versus the value of x_i for data point j, x_{ij}. More formally, for a data point x_{ij}, the graph plots points $(x_{ij}, r_j + \sum_{k=0}^{d_i} \beta_{ik} x_{ij}^k)$, where d_i is the number of polynomial terms in the regression model and β_{ik} is the regression coefficient for the x_i^k term of the regression. The plot then provides a visual representation of x_i's relationship with the outcome variable. In fact, if one were to run a regression on the values in the plot (which is shown in the plots below), one would find that the coefficients produced were exactly the same as those run in the original regression.


Partial Residual Plot for # of Bidders
partial residual for # of bidders
# of bidders
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
2
4
6
8
10


Partial Residual Plot for Quota Factor
partial residual for quota factor
quota factor
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
0.2
0.4
0.6
0.8


Partial Residual Plot for Capacity Std Dev
partial residual for capacity std dev
capacity std dev
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
1
2
3
4
5