Algorithmic Mechanism Design:

Auctions

Preamble

Mechanism design can be described as *reverse* game theory: instead of analyzing the outcome of a predefined game between strategic players, how do we design rules (a *mechanism*) to guarantee an outcome, independent of players' strategies? This is a topic that lies directly in the crossroads of economics and computer science. *Algorithmic* mechanism design, in particular, applies the knowledge of theoretical CS to satisfy economic needs, an approach very much in line with that of 15-251.

A particularly relevant mechanism is the auction, a market with the special case of a single seller. The auction problem is important because it tackles the two-fold problem of maximizing social surplus/good while also enforcing participant honesty, even in the face of private agendas. On top of these properties, mechanisms must also be computable in polynomial-time, so that auction procedures and results may be carried out correctly, yet efficiently.

The problem of designing well-behaved auctions is commonly studied because having only one seller greatly simplifies the analytical landscape. Yet, results are easily extensible to more realistic market scenarios. Today, the field is directly used in designing mechanisms for auctions of search engine ad slots, government deeds, property, inheritances, or even everyday items on Ebay.

1 Introductory Concepts and Definitions

Formally, an auction consists of n buyers/bidders, competing to buy some k number of goods from a single seller.

The Buyers' Side

Each buyer, i ($1 \leq i \leq n$), has his own *private* valuation of the good(s), v_i, which is unknown to the seller and all other buyers. In a *single-parameter environment*, which we will be focusing on, each buyer only has *one* metric, or parameter, for evaluating a good. In real-world terms, this often boils down to the price or monetary value one puts on an item.

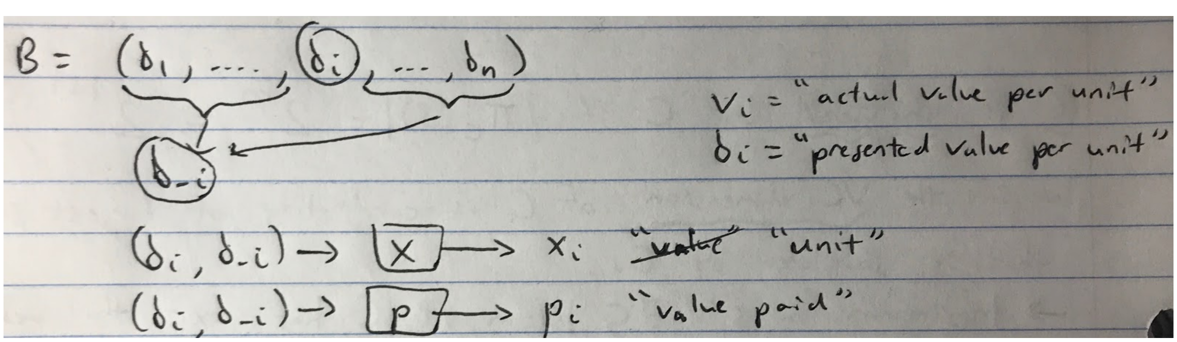
Using this valuation and other information known to him, i casts a bid, b_i, for the item(s). Bids are always known to the seller, and in the interest of discouraging dishonest strategy, we fix sealed-bid auctions, hiding bids from other bidders. So after a bid is cast, there is nothing to do but sit and wait...

The results are in! Clearly, there are winners and losers--those who won a good and those who did not. But what's a more quantitative way to look at this?

When reasoning about the outcome of an auction, one thing we look at is each bidder's utility, or "payoff". For our purposes we will use a quasilinear utility model [1], meaning that losers are assumed to not only gain nothing, but also pay nothing, resulting in a utility of 0. The utility of a winner is $v_i - p_i$, where v_i is the value attributed to the item(s) won, and p_i is the value paid to the seller.

The Seller's Side

On the note of payments, p_i, let's take a look at the seller's side. Having no competition, the seller in an auction is, in essence, the *mechanism designer* -- our point of view. Given the vector of all bids, $B = (b_1, \ldots, b_n)$, the seller's job is to decide **who gets what, and who pays what.** So we see that p_i is really the result of a *function* taking i and B as input. Likewise, any auction design consists of two rules, appropriately formalized as mathematical functions [1]:



1) Allocation rule, $x(b_i, b_{-i})$. The output of this function is x_i, which describes what, of the available goods, is allocated to buyer *i* after he casts b_i and his competitors cast $b_{-i} = B \setminus \{b_i\}$.

2) Payment rule, $p(b_i, b_{-i})$. The output of this function is p_i, the value buyer *i* must pay to the seller after he casts b_i and his competitors cast b_{-i}.
    a) Recall that utility $u_i = v_i - p_i$. Note v_i is more aptly defined as a "per-unit" valuation, s.t. the value i puts on an allocation of x_i is equivalent to $v_i x_i$.
    b) On a similar note, the same can be said of one's bid, b_i.

It is through the assignment of these two rules that we can design auctions tailored to specific scenarios and outcomes. With these definitions taken care of, we are off to the races!

2 Single-item Auctions, Social Surplus, and DSIC

First-Price Auctions

For our most basic case of auction design we examine the single-item auction, with variable *n* buyers and $k = 1$. How shall we design our *x* and *p* functions? As a seller interested in making money, it makes sense to use the most straightforward idea: award the item to the highest bidder, and charge that bidder whatever his bid was. This design seems reasonable enough: bidders can lie about their valuation, but if they overbid to win the auction, they'll have to pay the price--literally. So we should be good, right?

Unfortunately, first-price auctions are still frequently manipulated. Today, there is a strategy called bid-shading which, put simply, uses data to estimate the expected valuation of competitors, and calculate the lowest possible bid one could win an auction with. In modern online auctions over items like Google ad space, bid-shading is so profitable and common it's become a service [4]. As you can imagine, this leads to a lot of under-bidding. There are many other reasons to lie about your valuation as well--say your goal isn't to win the item but force your competitors to pay as much as possible (Why? Maybe you're secretly in league with the seller!). With a bit of work, dishonest biding can cause a determined rival to slowly but surely bid above his actual valuation.

Bottom-line, first-price auctions leave a lot of leeway for manipulating an auction by bidding dishonestly, simply because there's *no real incentive not to*. This not only hurts the seller when buyers conspire to bid less than their true valuations, but also because it makes it difficult to reason about the outcomes of the auction. As the seller/auction designer, how can we analyze buyer utility and the like, when bids have nothing to do with actual valuations?

A Concrete Auction Model

Let's first back up a bit and describe a more concrete model of what we want in an auction. In the preamble, we mentioned 3 things: **maximizing social surplus, honesty, and poly-time.** As we know now, the second property is paramount to ensure the first and third. But what is social surplus?

Formally, social surplus is $\boldsymbol{\Sigma v_i x_i}$ **, for** $\boldsymbol{i \in [1,n]}$ [1]. Interpreting this equation, "maximizing social surplus" has to do with making sure that the winners of the auction are the buyers who value their received item(s) the most. This measure is used as an *objective* performance guarantee, one that, economically, results in the best outcome for everyone. While we'll move on to maximize social surplus rather than seller profit, there's a field specifically devoted to this (while generally refraining from destroying the social good)--revenue maximizing auctions.

Lastly, we'll formalize the idea of "honesty". In an ideal world, buyers would always bid honestly, with $b_i = v_i$. Since buyers are transparent about their v_i, we can efficiently design and run an allocation rule x that maximizes social surplus. But in real life, how do we ensure that *rational* buyers will always act this way?

The answer is to make honest bidding Dominant-Strategy Incentive-Compatible, or, DSIC [1]. A strategy--in this case, choosing a bid for a desired allocation--is DSIC if 1) It is the dominant strategy and 2) it never leads to negative utility, thus minimizing risk (a different topic we will not dive into for now). As in all game theory, 1) means that of all the alternative bidding strategies, there is none that achieves a better result (in terms of utility) for our buyer, *in any instance of the auction*. 2) is self explanatory, and both properties are up to the auctioneer to guarantee.

To summarize, in a well-behaved auction:

1) It is DSCI to bid honestly.
2) Assuming honest bidding, social surplus is maximized, and
3) Along with properties like s.s., computing outcomes is polynomial time.

Vickrey (aka Second-Price) Auctions

Modern single-item auctions frequently employ the well-known Vickrey Auction [2], for the simple reason that it's proven to satisfy all three of the aforementioned properties. Here's how it works:

- Allocate the item to the highest bidder, so that $x(b_i, b_{-i}) = 1$ if $b_i > \max(b_{-i})$, $= 0$ otherwise. However,
- Charge the highest bidder *the second-highest bidder's bid*. $p(b_i, b_{-i}) = \max(b_{-i})$ if $b_i > \max(b_{-i})$, $= 0$ otherwise.

(We assume ties are broken arbitrarily and ignore this possibility here.)

We present a short proof that Vickrey Auctions satisfy the three properties [3]:

1) Vickrey Auctions are DSCI. For arbitrary player i, fix valuation v_i and the bids of other buyers, b_{-i}. Due to the binary nature of the allocation and payment rules, we only have two possible utilities for i: i loses (0 - 0 = 0), or i wins (v_i - max(b_{-i})). We case on the possible values of v_i.
<u>Case 1:</u> v_i < max(b_{-i}). (v_i - max(b_{-i})) < 0, therefore it is preferable to *lose* the auction and bid at v_i. Note all other losing bids ($b_i < v_i$) achieve the same utility of 0.
<u>Case 2:</u> $v_i \geq$ max(b_{-i}). (v_i - max(b_{-i})) ≥ 0, therefore it is preferable to *win* the auction and bid at v_i. Note all other winning bids ($b_i > v_i$) achieve the same utility of (v_i - max(b_{-i})), since payment depends on max(b_{-i}), not b_i.

	v_i < max(b_{-i})	$v_i \geq$ max(b_{-i})
Utility of bid < v_i	0 (lose)	0 in worst-case (lose)
Utility of bid = v_i	**0 (lose)**	**(v_i - max(b_{-i})) ≥ 0**
Utility of bid > v_i	(v_i - max(b_{-i})) < 0	(v_i - max(b_{-i})) ≥ 0

2) To maximize social surplus = $\Sigma v_i(x_i)$ when x_i = 1 for one value of i and 0 for all others, we want x_i to equal 1 when v_i is maximum, i.e. award the item to the buyer with the highest valuation. If buyers bid honestly, $v_i = b_i$ for all i, so this is equivalent to awarding the item to the highest bidder, which Vickrey Auctions do.

3) Informally, finding x_i and p_i for one of the n buyers relies on finding the maximum v_i among all *other* buyers, which can be accomplished with a simple linear scan in O(n). The cost for finding the outcome for all buyers is then O(n^2).

Therefore we have that Vickrey Auctions ensure great outcomes for all parties: there is a dominant strategy buyers can *always* use, *independent* of other bid strategies. On the seller's side, one can have faith that buyers are bidding honestly at $b_i = v_i$, allowing us to compute the outcome efficiently and also verify that we are indeed maximizing the social good.

William Vickrey | Canadian-American Professor | 1996 Nobel Memorial Prize in Economic Sciences

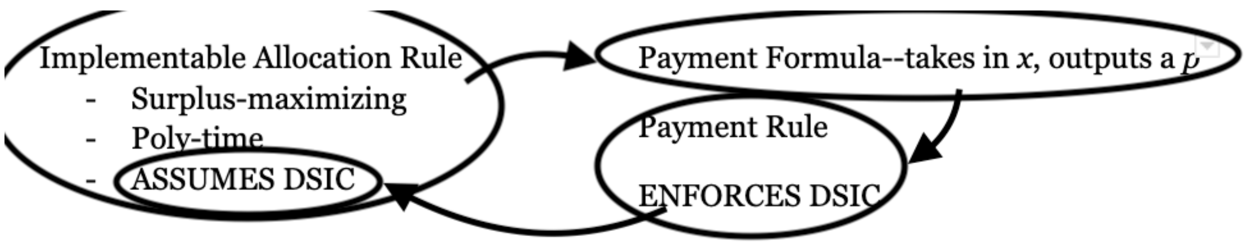
3 Implementable DSIC Auctions and Myerson's Lemma

Auction design strategy

Now that we have seen a case of good auction design, we want to work to generalize it as much as possible. Without knowing which problem instances on which this is possible, how may we approach the design of an arbitrary "well-behaved" auction?

One thing we'll note is that, although the second property, DSIC honesty, is the most desired and important, the first and third properties of surplus-maximization and poly-time are relatively easy to implement, if we ASSUME DSIC honesty. Therefore we propose the following approach [1]:

1) Assume DSIC honesty, and work to construct an allocation rule x that maximizes social surplus and is poly-time.
2) Given x, construct a payment rule p that enforces DSIC and is poly-time, *if possible.*



An allocation rule is implementable [1] if there exists a payment rule that causes the resulting auction to have the property of DSIC honesty. How do we know if a given x satisfies this?

Myerson's Lemma

Define an allocation rule x to be monotone [1] iff it is *non-decreasing*, i.e. for all bids $b_1 < b_2$, $x(b_1) \leq x(b_2)$. Myerson' s Lemma states:

1) *Monotone <=> Implementable.*
2) Given ANY monotone allocation rule x, there is a UNIQUE payment rule p that ensures that honest bidding strategy, $b_i = v_i$, is DSCI for all bidders. Further, this p may be obtained using an *equation*.

It's appropriate to say that this is *huge*. Myerson's Lemma allows us to carry out our auction design strategy for *any conceivable set-up*, given that we can find a poly-time, surplus-maximizing, and monotone allocation rule for it, which really isn't so hard if you think about it. See section 5 Myerson's Applications for examples--for now, we'll work to tackle a partial proof.

Proof of Part 1 (<=):
(*Completed* from partial proof from [1])
Fix allocation and payment rules $x(b_i, b_{-i})$ and $p(b_i, b_{-i})$. Assume that in the auction *(x, p)*, it is DSIC to bid honestly at $b_i = v_i$. Fix i and b_{-i} arbitrarily; to simplify things we'll write $x(z, b_{-i})$ and $p(z, b_{-i})$ as $x(z)$ and $p(z)$.

Fix an arbitrary positive bid z and bid y, $0 \le y < z$. As it stands, buyer i's valuation per-unit allocation, v_i, could be any value, and DSIC of $b_i = v_i$ must hold for that value.
Let $v_i = y$:

$\text{Utility}(y) \ge \text{Utility}(z)$, since $z \ne v_i = y$

$v_i x(y) - p(y) \ge v_i x(z) - p(z)$

$yx(y) - p(y) \ge yx(z) - p(z)$

$p(y) - p(z) \le yx(y) - yx(z)$

Let $v_i = z$:

$\text{Utility}(y) \le \text{Utility}(z)$, since $y \ne v_i = z$

$v_i x(y) - p(y) \le v_i x(z) - p(z)$

$zx(y) - p(y) \le zx(z) - p(z)$

$p(y) - p(z) \ge zx(y) - zx(z)$

Therefore we have:

$z(x(y) - x(z)) \le p(y) - p(z) \le y(x(y) - x(z))$, so

$z(x(y) - x(z)) \le y(x(y) - x(z))$.

Since $y < z$, it must be that $x(y) - x(z) \le 0$.
Therefore $x(y) \le x(z)$ for all $0 \le y < z$, i.e. x is monotone.

Proof of Part 2 and Payment Formula: A formal proof of the payment rule formula involves a good deal of calculus and limit math, and has been omitted, as a recreation of it would likely be unoriginal from the following primary source:

- https://timroughgarden.org/f13/l/l3.pdf
  (Lecture 3, Part 5) (Tim Roughgarden, Stanford CS364)

The payment formula proven by Myerson's is

$$p_i(b_i, \mathbf{b}_{-i}) = \sum_{j=1}^{\ell} z_j \cdot \text{jump in } x_i(\cdot, \mathbf{b}_{-i}) \text{ at } z_j,$$

for piecewise x and

$$p_i(b_i, \mathbf{b}_{-i}) = \int_0^{b_i} z \cdot \tfrac{d}{dz} x_i(z, \mathbf{b}_{-i}) dz$$

for general x.

Proof of Part 1 (=>) and the Correctness of the Payment Formula:
We informally do this by verifying select examples, using the graphical visualizations generated by our demo. See subsequent sections 4 and 5.

4 Demo: Semi-Automated Auction Design using Myerson's Lemma, with Visualization

Website: https://aliu39.github.io/auction_demo
GitHub: https://github.com/aliu39/auction_demo

Coded with JavaScript, React, and HTML Canvas, this interactive demo website takes in an inputted JavaScript function, representing allocation rule x. *Assuming fixed* b_{-i} (coded into the function), x takes in a single parameter b as a bid, and *must be monotone for the program to work correctly*.

Demo Part 1: Myerson's Visual Proof
Given any positive valuation v, the demo will plot 3 columns of 3 plots of b vs $x(b)$, with the following regions shaded in:

1) individual social surplus $v * x(b)$
2) payment $p(b)$
3) utility = $v * x(b) - p(b)$

The columns reflect $b = v$, $b < v$, and $b > v$ respectively. With 3) in mind, it should be visually apparent that utility(b) ≤ utility(v) for all b ≠ v.

The sources of inspiration for these visualizations can be found in the "visual proofs" of:

- Source [3]
- https://timroughgarden.org/f13/l/l3.pdf (Part 6, Page 7)

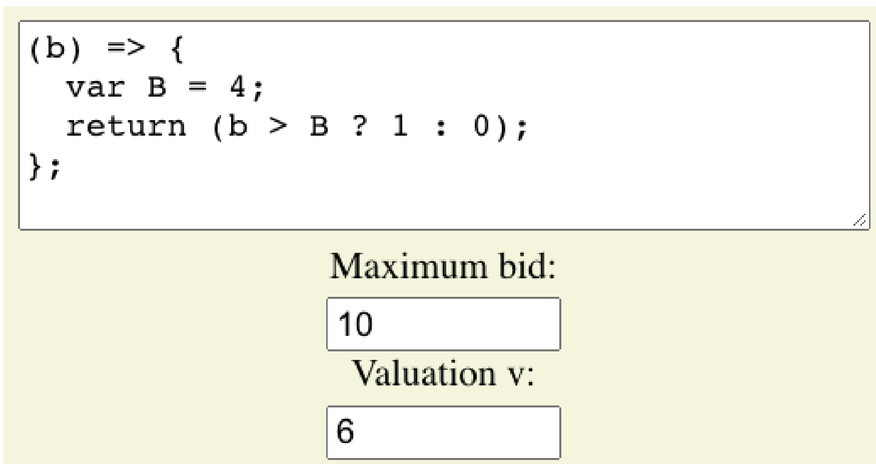
Demo Part 2: Payment Function Plot
Using the payment formula given by Myerson's Lemma (again, *assuming x is monotone*), the program computes and plots the payment rule, p. The integral and differential are estimated with an appropriately small dz value.

5 Myerson’s Applications

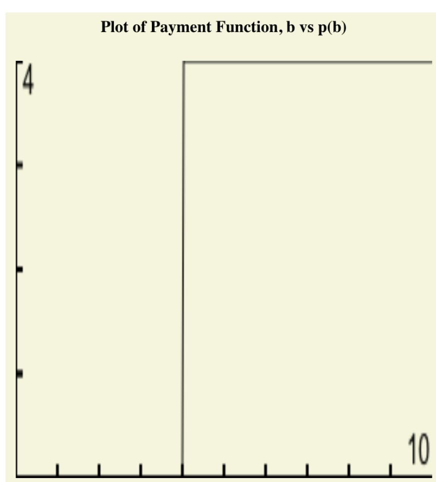
Using the demo, we will visually find and verify the correctness of Myerson’s Lemma on a few different inputs--namely, different types of DSIC-honesty implementable auctions.

<u>Revisiting Vickrey Auctions</u>
As proved, Vickrey Auctions satisfy our desired properties of surplus maximization and DSIC honesty. But what does the second-price rule look like visually?

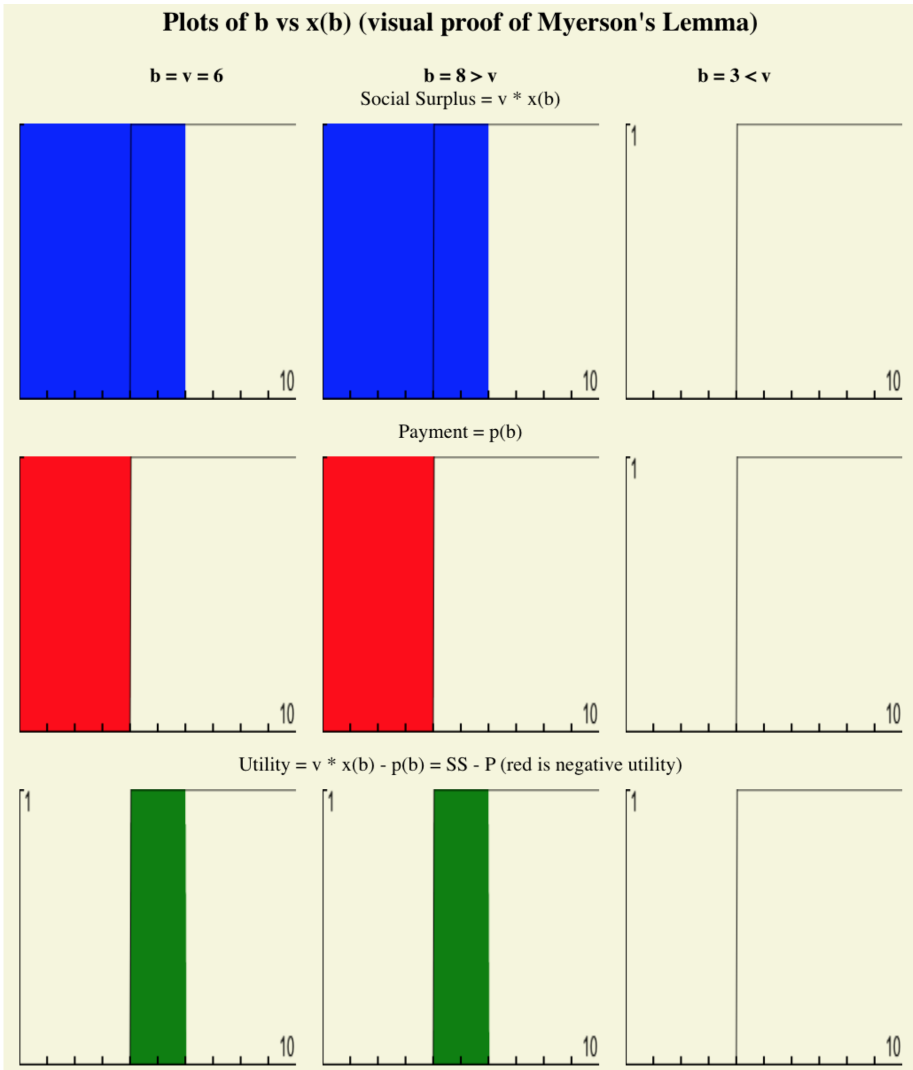
Below is sample input for a specific instance of a single-item allocation rule, which awards the item to the highest bidder. We fix b_{-i} as something like [2, 1, 3, 4], where B = 4, the maximum of b_{-i}, is hard-coded into the JavaScript function.



The resultant plot of b (more accurately, b_i) vs $x(b)$ is a piecewise constant function with a singular jump from 0 to 1 at b = B. It is apparent that x is non-decreasing/monotone, so applying the discrete Myerson’s payment formula for piecewise x, we should get $p(b)$ = B for b > B, 0 else. The demo reflects this:



Further, for a select valuation v_i = 6, $b = v_i$ (honest bidding) is verified to be the dominant strategy for maximizing individual utility. See row 3 of the b vs x(b) plots:



Note the graphical representations elucidate some very powerful visualization tools:
- Individual social surplus, $v * x(b_i)$, is a rectangular region from b = 0 to v, with height $x(b_i)$.
- **Payment, $p(b_i)$, is the area *above* the x curve for b = 0 to b_i !!!**
- Utility is simply the difference between these two regions. **When $b < v$, utility is limited by low individual social surplus; when $b > v$, *negative* utility is generated by high payment** (this is better demonstrated on non-piecewise x). Hence, DSIC honesty is maintained at $b = v$.

We leave it to the reader to visualize (or directly verify, using the demo!) why utility = $v * x(b) - p(b)$ is suboptimal for *all* values of $b < v$ and $b > v$, rather than just the presented b values. And, why the same properties hold for *all* $0 \le v \le$ (max bid).

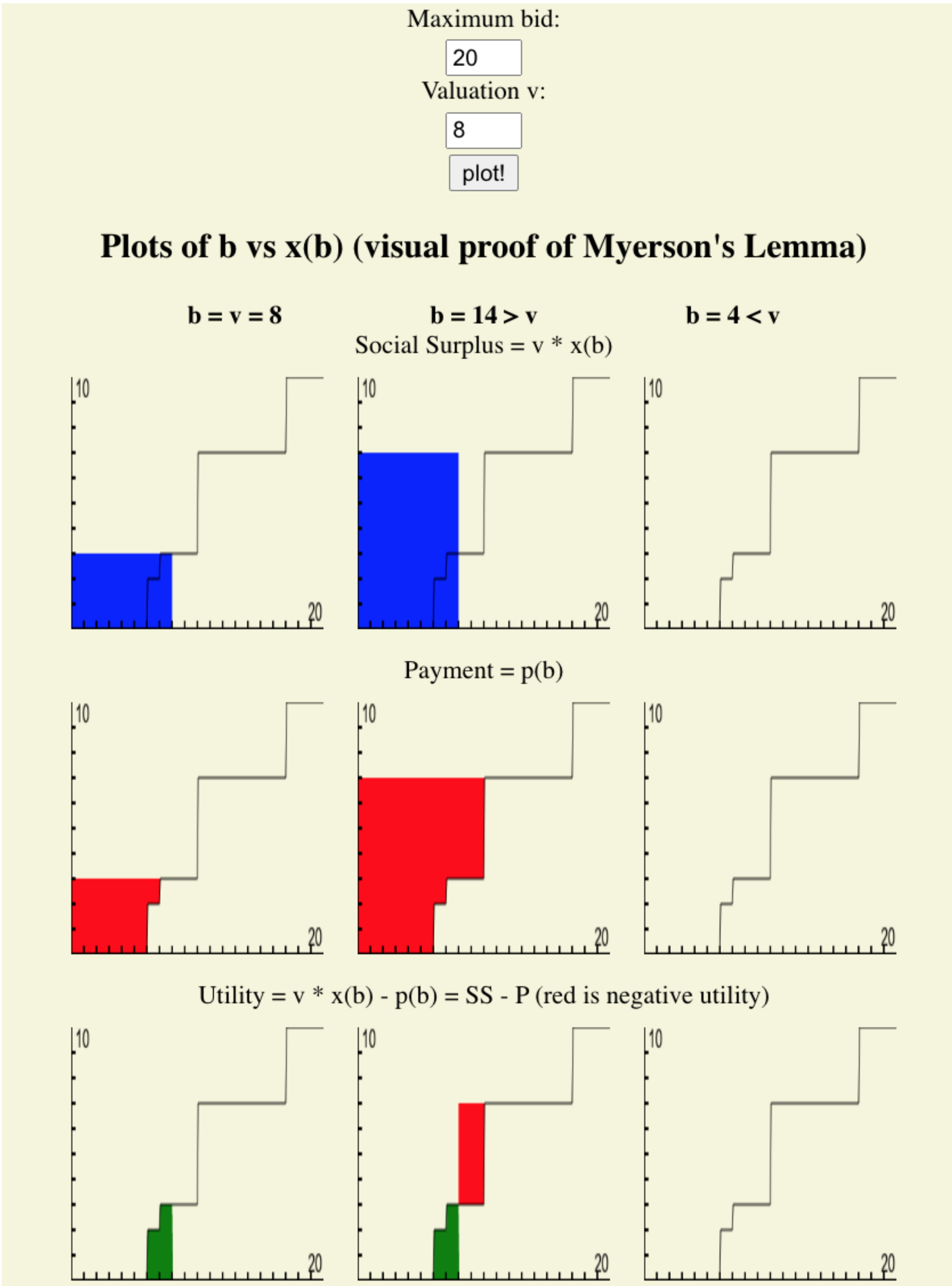
Ex2: Sponsored Search

Let's use Myerson's Lemma and our demo to look at a more complex auction problem. Sponsored search is named for the auction of search engine "sponsored" result slots to advertising companies. The goods for sale are *k* slots on a search results page, with each company having a single-parameter (price) valuation for each slot. Logically, we assume that slots higher on the page are universally valued higher than lower slots (users are more likely to click these).

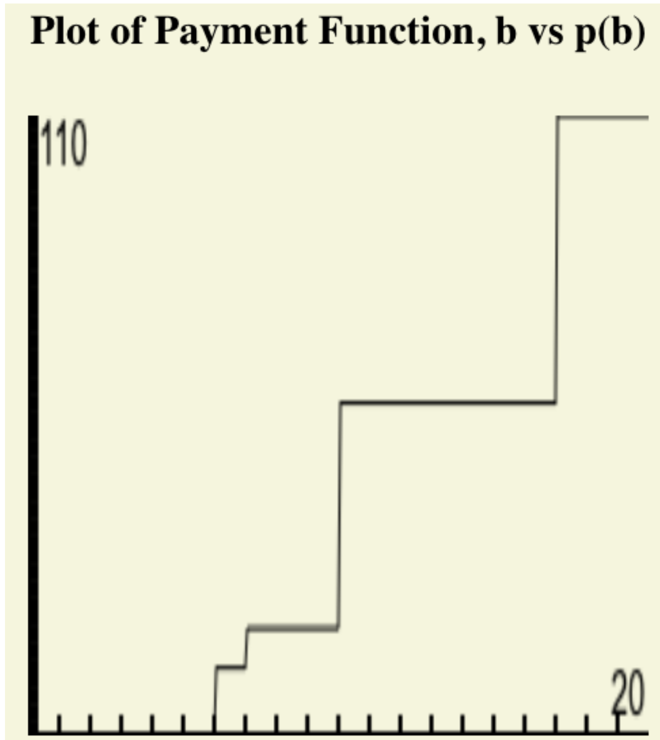
Think back to our design approach. Assuming DSIC honesty, how can we maximize surplus = $\Sigma v_i x_i$? Recalling the desired property of computational efficiency, let's try the simplest method, a greedy algorithm.

```
// SPONSORED SEARCH
b => {
  var k = 4; //slots up for grabs
  var xs = [10, 7, 3, 2]; //"unit" allocation of each slot, highest to lowest
  var bs = [1, 1, 6, 17, 3, 4, 10, 7, 5]; //the other bids (b(-i))

  //find how many bids b is lower than
  var i = 0;
  for (var j in bs) {
    if (b < bs[j]) {
      i++;
    }
  }

  if (i >= k) { //b loses the auction
    return 0;
  } else { //b wins the ith highest slot
    return xs[i];
  }
};
```

From the plot we can see this is again a piecewise constant non-decreasing function, which means it is provably implementable with Myerson's Lemma:

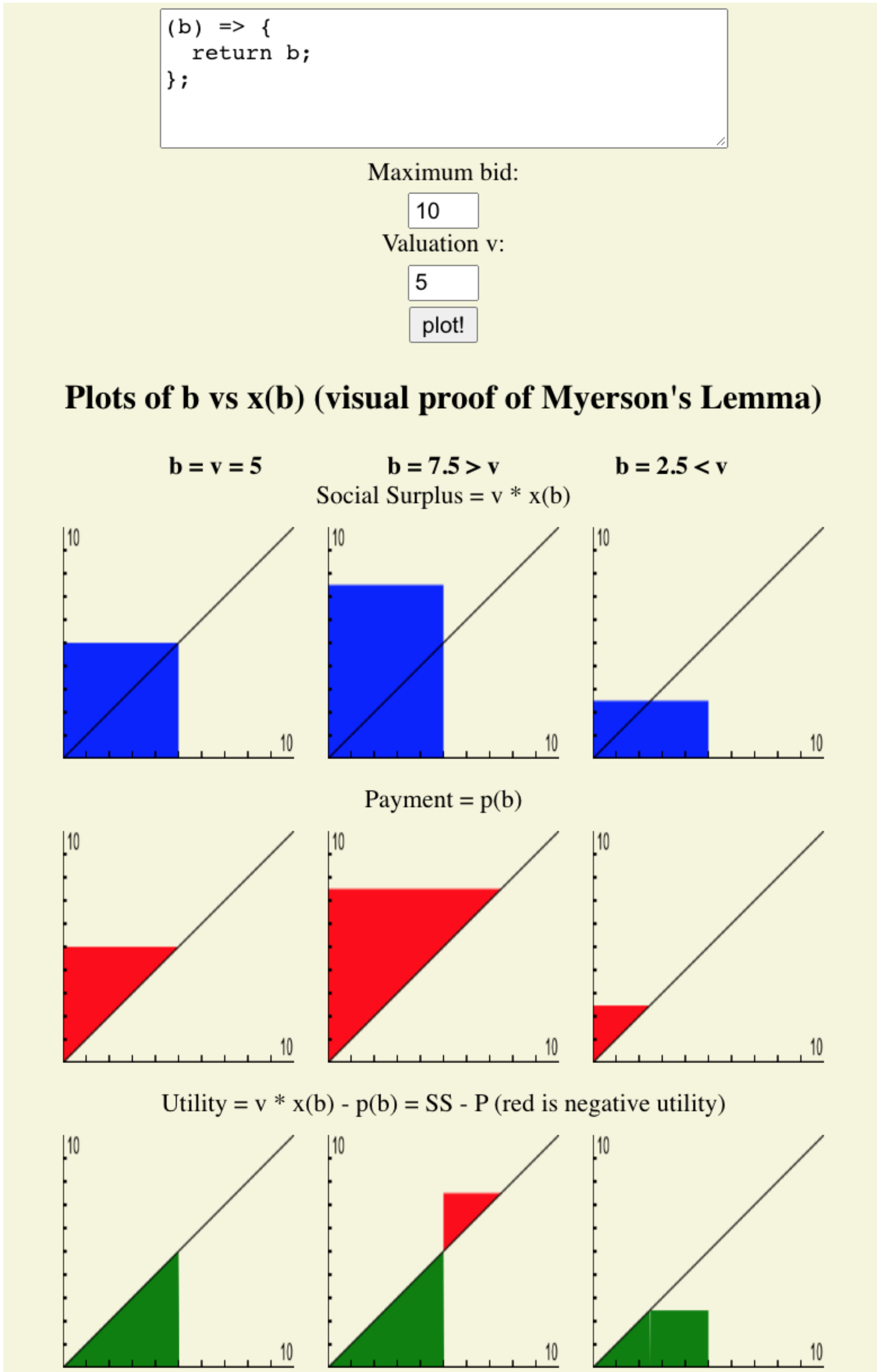


Due to the discrete number of values where (d/dz)(x(z)) is non-zero, the payment function is again piecewise constant, with p(b) equal to the aggregate difference between winning bids (less than or equal to b) and the bids directly below them.



Other Applications

Linear allocation (“you get what you bid”):

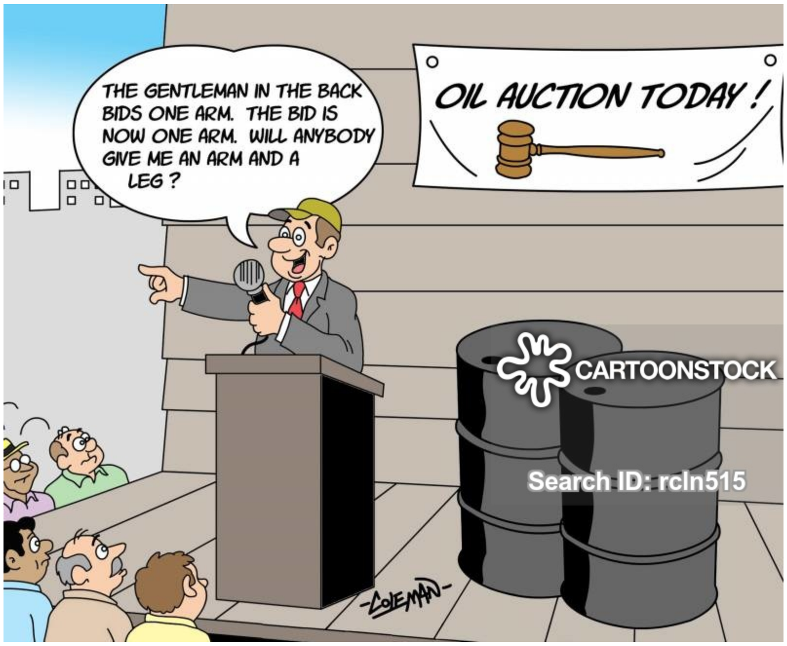


Such continuous/proportional allocations usually don't make sense in real-life--a possible parallel *could* be something like a government selling a virtually unlimited resource, for whatever quantity you name. However, in this case, it might make more sense for pricing to be linear, rather than quadratic (which is what is computed by Myerson's).

However, it's nevertheless helpful to model a continuous allocation rule to apply the payment formula to the *most general case*--we see that all non-decreasing allocations are implementable, not just piecewise ones.

It's also important to note, with continuous allocations we finally see cases of *negative utility*, confirming our earlier graphical observations that "when $b < v$, utility is limited by low individual social surplus; when $b > v$, negative utility is generated by high payment."

6 Real-world Examples



Oil Block Auctions

- Bid for rights to drilling in a "block" of land
- Highest bidder, first-price auction [5]
- Actual valuation of block is *unknown* until auction's end
- Each bidder can use one expert consultant for assistance

Open-outcry vs Sealed-bid:

- What difference does it make when bidders can see the other bids?
- How to enforce honesty in spite of this (as an example, does second-price still work)?

Dutch Auctions

- Traditionally “open-outcry,” meaning an auctioneer yells out prices in-person
- Key feature is that prices are *descending* (ascending auctions are “English”) [5]
- First buyer to raise hand/yell out gets the item at that price
- Less incentive to shade bid, or under-bid, and more incentive to bid high

Internet Auctions

- Now most prevalent form, and the reason for a resurgence in studying auctions
- With it, there are many new ways to manipulate outcomes, such as
- Proxies: using bots / automated programs to fill auctions with favorable bids
- Sniping: waiting until the last second to submit a bid barely above the highest bid

How may we devise implementable (monotone) allocation rules for these kinds of auctions, in a way that maximizes social surplus? When we do, does Myerson’s Lemma still hold despite the increasingly complex ways to manipulate results?

7 Check Your Understanding

1. True or false: the allocation and payment for each bidder is only dependent on his bid, and independent of his valuation.
2. Describe the properties of a well-designed auction, using the formal definitions for each property.
3. What does it mean to maximize the social surplus of an auction's outcome? What does this imply for the seller/auctioneer, and his/her profits?
4. True or false: piecewise allocation rules always have an implementable DSIC payment rule.
5. Generally describe the inequalities we employ to prove the first part of Myerson's Lemma.
6. What's the drawback of bidding *over* your valuation in a DSIC auction designed using Myerson's Lemma? Can you describe this graphically?

8 Mastery List

- What is mechanism design? And how is it different/related to game theory? Why are auctions important to study?
- Know the definitions and differences between a buyer's valuation, bid, and utility, and how this relates to (individual vs group) social surplus.
- Know the formal definitions for an auction's allocation and payment rules.
- Be clear on the models/assumptions used in this write-up, as they can often differ in other mechanism design topics or writings. These include things like sealed-bid auctions, single-parameter environment, and quasi-linear utility
- Remember the main concept of Myerson's Lemma: monotone = implementable, and the existence of a unique DSIC payment rule, which may be derived through a formula.
- Make sure you get down the two-steps of auction design using Myerson's Lemma, beginning with an allocation rule *under certain assumptions*, and deriving a payment rule afterwards.

9 Sources/Acknowledgements

[1] Roughgarden, Tim. 2013, *Algorithmic Game Theory: Mechanism Design Basics*, timroughgarden.org/f13/l/l2.pdf.

[2] Nisan, Noam, et al., editors. 2007, *Algorithmic Game Theory*, www.cambridge.org/files/3914/7629/0049/Algorithmic_Game_Theory.pdf.

[3] Cai, Yang. 2014, *Algorithmic Game Theory: Myerson's Lemma Proof Continued*, www.cs.mcgill.ca/~cai/COMP_MATH_553/LectureNotes04.pdf.

[4] Dixon, Mark. "Everything You Need to Know about First-Price Auctions." *Sortable*, 29 May 2019, sortable.com/blog/ad-ops/everything-you-need-to-know-about-first-price-auctions/.

[5] 2004, *Auctions and Bidding*, ocw.mit.edu/courses/sloan-school-of-management/15-010-economic-analysis-for-business-decisions-fall-2004/lecture-notes/auctions_bidding.pdf.

<u>Note on Source [1]:</u>

Many formal definitions and key concepts (sections 2-3 of this write-up) were taken from this source, but I seeked to make original contributions by

- Rewording into less technical, more understandable terms for readers and myself
- Interacting with material, doing things like..
    - Answering "why define this?" before presenting formal definitions
    - Extending ideas, like exploring how first-price auctions can be manipulated
    - Completing the Proof of Myerson's Lemma Part 1 (which was left as a partial proof / exercise in the original)