

# Chapter 1

## Approximation and Mechanism Design

### 1.1 Economics and Computer Science

The central topic of study in this course is approximation in mechanism design. This area lies at the intersection of computer science and economics and offers different insights to each respective field. Economics traditionally studies resource sharing among parties with selfish interests, e.g., exchanging goods, providing public services, auctions, voting, etc. Computer science studies resource sharing in computational settings, e.g., routing, computing, caching, etc. The computer science motivation for considering economic mechanism design comes from computer networks where these computational resource sharing questions become economic due to the interests of multiple users of the computer network. Algorithms and protocols for computer networks will only work well if they are designed with economic principles in mind. A combined approach from both fields is necessary for understanding the theory of computer networks.

In so far as the traditional economic theory is insufficient to address the concerns of resource sharing problems in computer networks, the study of such has potential to impact the economic theory. The main contribution and constraint imposed when looking at economic resource sharing questions from a computational perspective is in tractability and informational assumptions. Adding these constraints exposes a different economic phenomena which are central to the development of a more general economic theory. In order to tackle these new problems, traditional methodologies from computer science can be applied to economic questions. In particular the computer science method of *approximation* has an interesting and important role to play in the design and analysis of economic mechanisms.

### 1.2 Mechanism Design

*Mechanism design* give a theory for the design of protocols, services, laws, or other “rules of interaction” in which selfish behavior leads to good outcomes. By “selfish behavior” we

mean that each participant, hereafter *agent*, individually tries to maximize their own utility. Such behavior we define as rational. By “leads” we mean *in equilibrium*. A set of agent strategies is in equilibrium if no agent prefers to unilaterally change their strategy. Finally, by “good behavior” we mean that the objective of the designer is optimized. The most natural economic objectives are *social surplus*, the sum of the utilities of all parties, and *profit* the total payments made to the mechanism.

Our goal is a good theory of mechanism design. Such a theory would ideally satisfy the following three criteria:

**Informative:** It captures the essence of the problem but ignores details of specific settings.

**Prescriptive:** It gives concrete suggestions for how a good mechanism should be designed.

**Predictive:** It described how mechanisms are designed.

The last point is especially interesting as economics is traditionally a predictive science where one endeavors to understand why the world is the way it is. Mechanism design differs from most traditional economic studies as it also makes a suggestions it makes to designers.

The question of designing an *optimal mechanism* for a given setting and objective can be viewed as a standard optimization problem. Given incentive constraints, imposed by game theoretic strategizing, and feasibility constraints, imposed by the setting, optimize the designer’s given objective. I.e., put all the constraints into a solver and out will come a solution. It is immediately clear that this approach is uninformative. What have learned about the optimal mechanism? Furthermore, the optimization suggested is often computationally intractable. The resulting mechanism is likely to be complex and impractical. Only in overly idealistic settings does this lead to crisp informative predictions, but then these idealistic settings are unrealistic themselves. In conclusion, optimal mechanism design is neither prescriptive, predictive, nor informative.

## 1.3 Approximation

In settings where optimality is impossible (by any of the above critiques) one should instead try to approximate. The formal definition of an approximation is given below. Notice that approximation factors are always at least one as it is impossible to perform better than optimal. Approximations are defined differently for minimization and maximization problems so that regardless of the type of optimization a small approximation ratio is good and a large approximation ratio is bad.

**Definition 1.1** A mechanism is a  $\beta$ -approximation to OPT if its performance is

- at least  $\text{OPT} / \beta$  (for maximization problems) and
- at most  $\beta \cdot \text{OPT}$  (for minimization problems).

Depending on the problem and the approximation mechanism, approximation factors can range from  $(1 - \epsilon)$ , or arbitrarily close approximations, to linear factor approximations (or sometimes even worse). Notice a linear factor approximation is one where as the setting grows, i.e., more agents or more resources, the approximation factor gets worse. Table 1.3 summarizes the kinds of approximations we will encounter most frequently.

Factor	arbitrary close	constant	log	linear
possible?	rare	sometimes	often	almost always
informative?	no	yes	sometimes	rarely
performance?	excellent	good	ok	bad

Table 1.1: Common approximation factors and their properties.

Note that a designer would of course prefer to approximate their objective arbitrarily precisely. However, this is both often not possible and uninformative. It is uninformative because arbitrarily precise approximations are usually based on brute-force search over a restricted search space. Therefore, we do not learn much more from them than we do from optimal mechanism design. Linear approximations are usually fairly easy to come by and consequently do not provide much informative insight. Mechanism design problems often have simple logarithmic approximations as well and these are again usually not very informative. The “sweet spot” in this table is constant approximations which are both informative and pretty good. There is often a critical piece of understanding required to break the approximation gap between super-constant and constant approximation.

Nonetheless, if you were approached by a seller (henceforth: principal) to design a mechanism and you returned to triumphantly reveal an elegant mechanism that gives them a 2-approximation to their profit, you would probably find them a bit discouraged. After all, your mechanism leaves have of their profit on the table.

So what then is the point of a 2-approximation? First, a 2-approximation provides informative conclusions that can guide the design of better mechanisms for specific settings. Second, the approximation factor of two is a theoretical result that holds in a large set of settings, in specific settings the mechanism may perform better. It is easy, via simulation to evaluate the mechanism performance on specific settings to see how close to optimal it actually is. Third, in many settings the optimal mechanism is not understood at all, meaning the principals alternative to your 2-approximation is an ad hoc mechanism with no performance guarantee. This principal is of course free to simulate your mechanism and their mechanism in their given setting and decide to use the better of the two. In this fashion the principal’s ad hoc mechanism, if used, is because provably a 2-approximation. Finally, mechanism that are 2-approximations in theory arise in practice. In fact, that it is a 2-approximation explains why the mechanism arises. Even though it is not optimal, it is close enough. If was far from being optimal the principal (hopefully) would have figured this out and adopted a different approach.

In summary, this is a course on approximation in mechanism design. We will search for positive results which will usually be in the form of a constant approximation. Where we

fail to find positive results, we will search for negative results in the form of an impossibility of constant approximation.

## 1.4 Three Examples: Path Auctions, Optimal Auctions, and Combinatorial Auctions

We motivate the kinds of questions we will be asking with three examples: *path auctions*, *optimal auctions*, and *combinatorial auctions*. The setting of path auctions connects auction-type questions to economics-type questions. Instead of looking for a shortest path in a graph, we hope to buy a cheap path to send a message. The challenge is that we do not know the costs of the edges, instead they are owned by selfish agents (e.g., Internet service providers), and these selfish agents will only route your message if it is in their financial best interest. How can we design an auction to procure a path with low total cost? The optimal auction question is related to an eBay like setting where a seller wants to optimize their revenue from selling an item in an auction. What auction mechanism should the seller use? The final example we wish to allocate a set of items among bidders with preferences over bundles of items. A special case of such a setting is when each bidder has a particular bundle they desire. This puts an economic twist on an optimization problem, *Weighted Set Packing*, which is well known to be computationally intractable. In all the above settings, we may be unable to design an optimal mechanism; how can we approximate the optimal mechanism?

### 1.4.1 Path Auctions

Suppose a sender wishes to send a message from one computer to another in a computer network and there is a transmission cost on each edge. How can they send the message for the cheapest? This is one of the classic combinatorial optimization problems, a.k.a., the shortest path problem. The input is a graph  $G = (V, E)$  on vertices  $V$  and edges  $E$ , a source  $s$ , a destination  $t$ , and costs on edges, e.g.,  $c_e$  for edge  $e$ . The this algorithmic problem can be solved by Dijkstra's Algorithm which is essentially a modified breadth first search.

What happens when the costs are *private information of selfish agents*. E.g., if the edges are links in a computer network, these links are owned by Internet service providers. There is a cost for routing packets on these links. Each service provider knows the cost of their link, but the sender of the message does not. Furthermore, the service provider will only send the message if its in their best interest. I.e., the benefit of sending the message must outweigh the cost. Consider what might happen in the mechanism below that does not take the incentives of the service provider into account.

#### Mechanism 1.1

1. Ask agents to report costs. ( $\Rightarrow$  agent  $e$  reports  $b_e$ )
2. Find shortest path with costs  $b_e$ . ( $\Rightarrow$  path  $P_b$ )

3. *Send message on this path.*

Note that the sender's cost here, thus far, is zero. The edges pay the cost. Unfortunately, none of the edges will want to participate if they are expected to pay the cost for no reward. Thus, we should expect each edge  $e$  to report  $b_e = \infty$ . With such reports, the mechanism will fail to find a path at all.

A simple fix to the above mechanism would be to pay the agents for their efforts. The most natural approach would be to pay each agent their declared cost.

### **Mechanism 1.2 (First Price Path Auction)**

1. *Ask agents to report costs. ( $\Rightarrow$  agent  $e$  reports  $b_e$ )*

2. *Find shortest path with costs  $b_e$ . ( $\Rightarrow$  path  $P_b$ )*

3. *Send message on this path.*

4. *Pay each  $e$  on  $P_b$  their bid  $b_e$ .*

First notice that agents may not report their true costs. Therefore, the path found may not be the shortest path for the actual costs, a.k.a.,  $P_c$ . So while the total cost of this mechanism is just the sum of the declared costs of the edges on  $P_b$ , i.e.,  $\sum_{e \in P_b} b_e$ , it is hard to reason about what this might be.

### **Single-item Auctions**

To get some intuition for what might happen it is instructive to consider the special case where there are only two vertices  $s$  and  $t$  and all edges are “parallel links”, i.e., each goes directly from  $s$  to  $t$ . What does our mechanism do in this special case? It picks the edge with the cheapest declared cost and pays it its declaration. In this special case is isomorphic to a *single-item reverse auction problem* and which is isomorphic to the *single item auction problem* the proposed mechanism is the *first-price auction*. Unlike the more general path auction, this auction and problem has been studied a great deal in economics and is very well understood.

**Definition 1.2 (Single-Item Auction)** A seller has a single item to sell to a number of interested buyers, each buyer has a value for receiving the item. The seller may solicit bids, and pick a winner and payments.

To get some appreciation for the strategic elements of the first price auction note that a bidder who wins wants to pay as little as possible, so bidding a low amount is desirable. Of course, if they bid too low, then they probably will not win. Strategically this bidder must figure out how to balance this tradeoff. To make the problem more specific assume there are two bidders and their values are drawn uniformly at random from the interval  $[0, 1]$ . How should a bidder with value  $v$  bid? Clearly something less than  $v$ , but how much?

Later in the course we will show that this bidder should bid  $v/2$  and the analysis is possible because of the inherent symmetry of the setting. Unfortunately our original path auction setting is highly asymmetric. So this approach is not likely to be fruitful.

Since we have distilled the problem to a single-item auction setting we can consider other single-item auctions as solutions. Probably the most natural is the English auction.

### Mechanism 1.3 (English Auction)

1. *Gradually raise an offer price up from zero.*
2. *Allow bidders to drop out when they no longer wish to win at the offer price.*
3. *Stop when at price at which there is only one bidder left.*
4. *The remaining bidder wins and pays this final price.*

Strategically this auction is much simpler than the first-price auction. What should a bidder with value  $v$  do? A good strategy would be “drop when the price exceeds  $v$ .” Indeed, regardless of the actions of the other bidders, this is a good strategy for the bidder to follow, i.e., it is a *dominant strategy*. Since all bidders have such a dominant strategy it is reasonable to assume that they follow these strategies.

Since we know how bidders are behaving we can now make conclusions as to what happens in the auction. In fact, the bidder with the highest value, denoted  $v_{(1)}$ , will win and this bidder will pay the second highest bidder’s value,  $v_{(2)}$ . Furthermore, this auction maximizes the *social surplus*, the sum utilities of all players. Notice that the utility of losers are zero, the utility of the winner is  $v_{(1)} - v_{(2)}$ , and the utility of the seller is  $v_{(1)}$ , the payment received from the winner. The total is simply  $v_{(1)}$  as the payments occur once positively (for the seller) and once negatively (for the winner) and these terms cancel. Of course  $v_{(1)}$  is the optimal surplus possible. We could not give the item to anyone else and get more value out of it.

Unfortunately, despite the good properties of this auction and how easy it is to analyze. It does not generalize to give a good path auctions. Challenges arise because of the structure of feasible outcomes of path auctions. We discuss the bad, for mechanism design, structural properties of paths later.

A solution to this problem comes from Nobel laureate William Vickrey who observed that if we simulate the English auction with sealed bids we arrive at the same outcome in equilibrium without the need to think about an ascending price.

### Mechanism 1.4 (Second-Price Auction)

1. *Accept sealed bids.*
2. *The highest bidder wins and pays the second highest bid.*

To see what happens in this auction notice that as this is just a simulation of the English auction and dominant strategies in the English auction were to “drop at your value” then the only way a bidder could achieve the same outcome in the simulation is to input their true value. While this intuitive argument can be made formal, instead we will argue directly that *truthful* bidding is a dominant strategy in the second price auction.

**Theorem 1.3** *Truthful bidding is a dominant strategy in the second price auction.*

*Proof:* We show that truthful bidding is a dominant strategy for agent  $i$ . Let fix the bids of all other agents and let  $t_i = \max_{j \neq i} v_j$ . Notice that given this  $t_i$  there are only two possible outcomes for agent  $i$ . If  $i$  bids  $b_i > t_i$  then  $i$  wins, pays  $t_i$  (which is the second highest bid), and has utility  $u_i = v_i - t_i$ . On the other hand, if  $i$  bids  $b_i < t_i$  then  $i$  loses, pays nothing, and has utility  $u_i = 0$ . This analysis allows is to plot the utility of agent  $i$  as a function of their bid in two relevant cases, the case that  $v_i < t_i$  and the case that  $v_i > t_i$ .

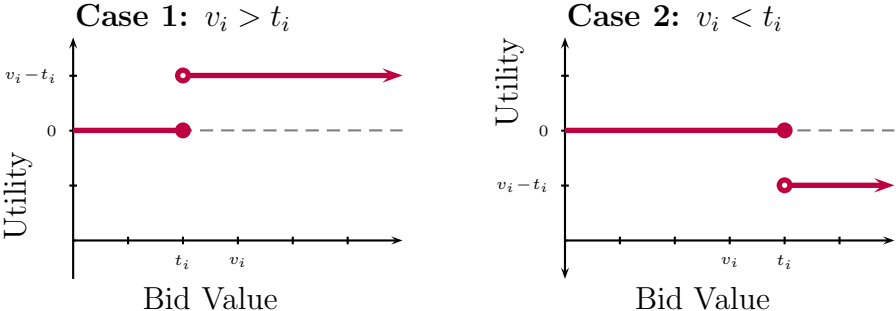


Figure 1.1: Utility as a function of bid in the second-price auction.

Agent  $i$  would like to maximize their utility. In case 1, this is achieved by any bid greater than  $t_i$ . In case 2, it is achieved by any bid less than  $t_i$ . Notice that in either case bidding  $b_i = v_i$  is a good choice. Since the same bid is a good choice regardless of which case we are in, the same bid is good for any  $t_i$ . Thus, bidding truthfully, i.e.,  $b_i = v_i$  is a dominant strategy. ■

Notice that in the proof of the theorem  $t_i$  is the infimum of bids that the bidder can make and still win, and the price charge to such a winning bidder is exactly  $t_i$ . We henceforth refer to  $t_i$  as agent  $i$ 's *critical value*.

### The Second-price Path Auction

Finally we are ready to propose an auction for procuring a path in a graph. The last step of this auction, i.e., paying each winner their critical value, is non-trivial. An intuitive way to perform this for agent  $i$  would be to hold the bids of all other agents fixed and imagine increasing the agent's bid until the shortest path changes.<sup>1</sup> The bid of the agent at the

---

<sup>1</sup>We increase the bid to find the critical value because this is a procurement problem and the bid represents a cost.

point that the path changes is the agent's critical value. Later we will give a more formulaic approach to calculating critical values.

### **Mechanism 1.5 (Second-price Path Auction)**

1. *Solicit sealed bids.*
2. *Find the shortest path with the costs equal to the bids.*
3. *Pay each winner their critical value.*

We will skip for now making a formal argument that these critical values exist. However, for path auctions it is not hard to argue. If an edge is selected and we increase its cost the path selected will increase in cost until its costs exceeds that of a path that does not contain this edge. If we continue to increase this edge cost it will, of course, the other path will remain cheaper. Thus, the edge has a critical value.

**Theorem 1.4** *The second-price path auction has truthful bidding as a dominant strategy.*

**Corollary 1.5** *The second-price path auction maximizes the social surplus.*

The proof of the above theorem is identical to the analogous result for the second price single-item auction. The corollary follows because payments cancel.

Of course in our original path procurement setting we were hoping to procure a path at low cost to ourselves, not necessarily to procure the path the the true minimum total cost. Thus, it is not clear whether the second-price path auction is actually achieving our objective. We close this section with an example that suggests that the second-price path auction may indeed have bad performance and questions that we will address later.

**Example 1.6 (path auction)** A graph with two  $s-t$  paths. A three-hop “upper path” with cost of 1 on each edge and a one-hop “lower path” with cost of 10. See Figure 1.2, below. Notice:

- $G =$  two vertex disjoint  $s-t$  paths:  $P = (1, 1, 1)$  and  $P' = (10)$ .
- the second-price payments are  $= 8 \times 3 = 24$ .
- second cheapest path  $= 10$

Thus, the total payment is a lot more than the even the cost of the second cheapest path.

**Question 1.1** Does the second price path auction minimize payments?

**Question 1.2** If not, does the total payment approximate the minimum cost path (or second minimum cost path)?

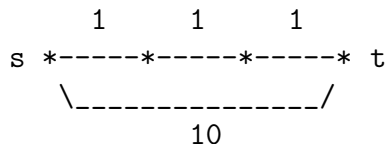


Figure 1.2: Example graph where the second-price path auction has large “overpayment”.

Notice, it is not entirely fair to compare the payments of the second-price path auction to the minimum cost path. For instance, even in the single-item auction example the second-price auction revenue is below the highest bid. Therefore, the second cheapest cost path may be a more reasonable target. Notice that in the single-item setting the second-price auction revenue is exactly the second highest bid.

**Question 1.3** If not, can we design a mechanism where the total payments do approximate the second minimum cost path?

## 1.4.2 Optimal Auctions

Suppose a seller has a single item to sell to at most one of a number of interested buyers. How should they sell this item to maximize their profit? For instance, consider running the second-price auction in a setting where there are two bidders whose valuations are known to be drawn independently at random from the uniform distribution on  $[0, 1]$ . The following fact will be helpful in our analysis.

**Fact 1.7** *In expectation, i.i.d. random variables chosen uniformly from a given interval will evenly divide the interval.*

Our equilibrium analysis in the preceding section showed that the winner in the second price auction is the bidder with the highest valuation and this bidder pays the second highest valuation. With two uniform random variables on  $[0, 1]$  the expected highest and second highest valuation are  $2/3$  and  $1/3$  (by Fact 1.7). Therefore the expected revenue is  $1/3$ .

A natural question to ask at this point is whether it is possible to do better and how. Consider the following definition of the second-price auction with reservation price  $r$  for selling a single item. This auction is of interest, for example, if the seller has valuation  $r$  for retaining the item. It is instructive to think of the second-price auction with reserve price  $r$  as being implemented by running the second-price auction with an additional bid placed on behalf of the seller with value  $r$ . If this bid wins, then the seller keeps the item. The truthfulness of this auction follows trivially from this viewpoint.

**Definition 1.8 (Second-price Auction with reservation price  $r$ )** The *second-price auction with reservation price  $r$* , sells the item if any bidder bids above  $r$ . The price the winning bidder pays the maximum of the second highest bid and  $r$ .

We now show that the seller can increase their expected profit in our example scenario by pretending to have a valuation of  $1/2$  for the item and running the second-price auction with reservation price  $r = 1/2$ . In particular when  $v_i$  are independent and identically distributed uniformly on  $[0, 1]$ , a reservation price of  $1/2$  in the second-price auction yields an expected profit of  $5/12$  which is more than  $1/3$ . To show this we consider three cases based on the randomization in the bidders' valuations.

**Case 1** (both bids are less than  $1/2$ ): This happens with probability  $1/4$  and in this case the auction does not sell the item and has no profit.

**Case 2** (both bids are above  $1/2$ ): This happens with probability  $1/4$  and in this case the expected profit is the expected value of the second highest value. This is the second order statistic of random variables that are uniform on  $[1/2, 1]$  which can be calculated, similarly to before, to be  $2/3$ .

**Case 3** (otherwise, one bid is less than  $1/2$  and the other is above  $1/2$ ): This happens with probability  $1/2$  and the expected profit is the reservation price  $1/2$ .

The expected profit of the auction is thus,  $1/4 \times 0 + 1/4 \times 2/3 + 1/2 \times 1/2 = 5/12$ .

We have thus shown that given prior knowledge of the distribution from which the bidders' valuations are drawn, it is possible to give an auction with higher expected profit than the second-price auction.

This preliminary analysis leaves a number of questions that we will explore in detail later.

**Question 1.4** What is the optimal auction for any given distribution?

**Question 1.5** If the optimal auction is much more complicated than the second price auction with a reserve price, is the second price auction with a reserve price at least a good approximation to it?

**Question 1.6** Can we design one auction that simultaneously approximates the optimal auction for all distributions?

### 1.4.3 Combinatorial Auctions

For our last example consider the a setting where a number of resources are to be allocated by auction and the bidders have preferences over bundles of bidders. For instance, if we were selling a television and a DVD player, one might desire to have both the TV and the DVD player, but getting just one of the two wouldn't be desirable at all. Consider the stylized special case of this problem where each bidder has a single, specific desired bundle, a.k.a., the *single-minded combinatorial auction* problem.

**Definition 1.9** single-minded combinatorial combinatorial auction There are  $n$  bidders and  $m$  items. Bidder  $i$  has value  $v_i$  for bundle  $J_i \subseteq [m]$ . Item  $j$  may be sold to at most one bidder. A *single-minded combinatorial auction* solicits bids, allocates the items to bidders, and charges each bidder a payment.

We will say an agent  $i$  is *served* if they receive their entire bundle. Of course no auction can serve agent  $i$  and  $i'$  if  $J_i \cap J_{i'} \neq \emptyset$ . Call such a pair of agents *incompatible*. We will call a subset  $S$  of the  $n$  agents *feasible* if no pair of agents in  $S$  are incompatible. In this terminology, any auction must serve a feasible set of agents.

A natural auction to consider here is the generalization of the second-price auction to this setting.

**Mechanism 1.6 (second-price single-minded combinatorial auction)**

1. Find the feasible set  $S$  that maximizes the surplus  $\sum_{i \in S} v_i$ .
2. Charge winners their “critical value”.

By the same discussion as for path auctions, these critical values exist. Therefore this mechanism has a dominant strategy of truthful bidding and it maximizes the social surplus.

**Theorem 1.10** *The second-price single-minded combinatorial auction has truthful bidding as a dominant strategy and maximizes the social surplus.*

Unfortunately, this is far from the end of the story. The problem is that Step 1 of the mechanism requires solving an optimization problem that is well known to be computationally intractable (under standard complexity-theoretic assumptions); the problem is known as Weighted Set Packing and it is  $\mathcal{NP}$ -complete.

**Theorem 1.11** *The Weighted Set Packing problem is  $\mathcal{NP}$ -complete.*

Therefore, if we believe it is impossible for a designer to implement a mechanism for which *winner determination* is computationally intractable, we cannot accept the second-price single-minded combinatorial auction problem.

Algorithmic theory has an answer to intractability: if computing the optimal solution is intractable, try to approximate the optimal solution.

**Question 1.7** Can we replace Step 1 in the mechanism with an approximation algorithm?

**Question 1.8** If not, can we design a computationally tractable approximation mechanism for single-minded combinatorial auctions?

Our construction of second-price auctions by maximizing social surplus and then charging each winner their “critical value” is quite general. We will discuss this more later.

**Question 1.9** Is there a general theory for designing mechanisms from approximation algorithms?

## 1.5 Notes

The path auction problem was proposed for study by Nisan and Ronen [19] and the overpayment of the proposed mechanism was first discussed by Archer and Tardos [2]. Optimal single item auctions in Bayesian settings were solved by Myerson [18]. Bayesian approximation by simple mechanisms was studied by Hartline and Roughgarden [15]. The issue of computational tractability in combinatorial auctions was first pointed out by Lehmann et al. [17].