1
Mechanism Design and Approximation

Our world is an interconnected collection of economic and computational systems. Within such a system, individuals optimize their actions to achieve their own, perhaps selfish, goals; and the system combines these actions with its basic laws to produce an outcome. Some of these systems perform well, e.g., the national residency matching program which assigns medical students to residency programs in hospitals, e.g., auctions for online advertising on Internet search engines; and some of these systems perform poorly, e.g., financial markets during the 2008 meltdown, e.g., gridlocked transportation networks. The success and failure of these systems depends on the basic laws governing the system. Financial regulation can prevent disastrous market meltdowns, congestion protocols can prevent gridlock in transportation networks, and market and auction design can lead to mechanisms for allocating and exchanging goods or services that yield higher profits or increased value to society.

The two sources for economic considerations are the preferences of individuals and the performance of the system. For instance, bidders in an auction would like to maximize their gains from buying; whereas, the performance of the system could (i.e., from the perspective of the seller) be measured in terms of the revenue it generates. Likewise, the two sources for computational considerations are the individuals who must optimize their strategies, and the system which must enforce its governing rules. For instance, bidders in the auction must figure out how to bid, and the auctioneer must calculate the winner and payments from the bids received. While these calculations may seem easy when auctioning a painting, they both become quite challenging when, e.g., the Federal Communications Commission (FCC) auctions cell phone spectrum for which individual lots have a high degree of complementarities.

These economic and computational systems are complex. The space

of individual strategies is complex and the space of possible rules for the system is complex. Optimizing among strategies or system rules in complex environments should lead to complex strategies and system rules, yet the individuals' strategies or system rules that are successful in practice are often remarkably simple. This simplicity may be a consequence of individuals and designers preference for ease of understanding and optimization (i.e., tractability) or robustness to variations in the scenario, especially when these desiderata do not significantly sacrifice performance.

This text focuses on a combined computational and economic theory for the study and design of mechanisms. A central theme will be the tradeoff between optimality and other desirable properties such as simplicity, robustness, computational tractability, and practicality. This tradeoff will be quantified by a theory of approximation which measures the loss of performance of a simple, robust, and practical approximation mechanism in comparison to the complicated and delicate optimal mechanism. The theory provided does not necessarily suggest mechanisms that should be deployed in practice, instead, it pinpoints salient features of good mechanisms that should be a starting point for the practitioner.

In this chapter we will explore mechanism design for routing and congestion control in computer networks as an example. Our study of this example will motivate a number of questions that will be addressed in subsequent chapters of the text. We will conclude the chapter with a formal discussion of approximation and the philosophy that underpins its relevance to the theory of mechanism design.

1.1 An Example: Congestion Control and Routing in Computer Networks

We will discuss novel mechanisms for congestion control and routing in computer networks to give a preliminary illustration of the interplay between strategic incentives, approximation, and computation in mechanism design. In this discussion, we will introduce basic questions that will be answered in the subsequent chapters of this text.

Consider a hypothetical computer network where network users reside at computers and these computers are connected together through a network of routers. Any pair of routers in this network may be connected by a network link and if such a network link exists then each router

can route a message directly through the other router. We will assume
that the network is completely connected, i.e., there is a path of network
links between all pairs of users. The network links have limited capacity;
meaning, at most a fixed number of messages can be sent across the link
in any given interval of time. Given this limited capacity the network
links are a resource that may be over demanded. To enable the sending
of messages between users in the network we will need mechanisms for
congestion control, i.e., determining which messages to route when a
network link is over-demanded, and *routing*, i.e., determining which path
in the network each message should take.

There are many complex aspects of this congestion control problem:
dynamic demands, complex networks, and strategic user behavior. Let
us ignore the first two issues at first and focus on the latter: strategic
user behavior. Consider a static version of this routing problem over
a single network link with unit capacity: each user wishes to send a
message across the link, but the link only has capacity for one message.
How shall the routing protocol select which message to route?

That the resource that the users (henceforth: agents) are vying for
is a network link is not important; we will therefore cast the problem
as a more general single-item resource allocation problem. An implicit
assumption in this problem is that it is better to allocate the item to
some agents over others. For instance, we can model the agents as having
value that each gains for receiving the item and it would be better if the
item went to an agent that valued it highly.

Definition 1.1   The *single-item allocation* problem is given by

- a single indivisible *item* available,
- n strategic *agents* competing for the item, and
- each agent i has a *value v_i* for receiving the item.

The objective is to maximize the *social surplus*, i.e., the value of the
agent who receives the item.

The social surplus is maximized if the item is allocated to the agent
with the highest value, denoted $v_{(1)}$. If the values of the agent are pub-
licly known, this would be a simple allocation protocol to implement. Of
course, e.g., in our routing application, it is rather unlikely that values
are publicly known. A more likely situation is that each agent's value
is known privately to that agent and unknown to all other parties. A
mechanism that wants to make use of this private information must

then solicit it. Consider the following mechanism as a first attempt at a single-item allocation mechanism:

  (i) Ask the agents to report their values ($\Rightarrow$ agent i reports b_i),
 (ii) select the agent $i^\star$ with highest report ($\Rightarrow i^\star = \mathrm{argmax}_i\, b_i$), and
(iii) Allocate the item to agent $i^\star$.

Suppose you were one of the agents and your value was \$10 for the item; how would you bid? What should we expect to happen if we ran this mechanism? It should be pretty clear that there is no reason your bid should be at all related to your value; in fact, you should always bid the highest number you can think of. The winner is the agent who thinks of and reports the highest number. The unpredictability of the outcome of the mechanism will make it hard to reason about its performance. There are two natural ways to try to address this unpredictability. First, we can accept that the bids are meaningless, ignore them (or not even solicit them), and pick the winner randomly. Second, we could attempt to penalize the agents for bidding a high amount, for instance, with a monetary payment.

Definition 1.2   The *lottery mechanism* is:

 (i) select a uniformly random agent, and
(ii) allocate the item to this agent.

The *social surplus* of a mechanism is total value it generates. In this routing example the social surplus is the value of the message routed. It is easy to calculate the expected surplus of the lottery. It is $1/n \sum_i v_i$. This surplus is a bit disappointing in contrast to the surplus available in the case where the values of the messages were publicly known, i.e., $v_{(1)} = \max_i v_i$. In fact, by setting $v_1 = 1$; $v_i = \epsilon$ (for $i \neq 1$); and letting ϵ go to zero we can observe that the surplus of the lottery approaches $v_{(1)}/n$; therefore, its worst-case is at best an n approximation to the optimal surplus $v_{(1)}$. Of course, the lottery always obtains at least an nth of $v_{(1)}$; therefore, its worst-case approximation factor is exactly n. It is fairly easy to observe, though we will not discuss the details here, that this approximation factor is the best possible by any mechanism without payments.

Theorem 1.1   *The surplus of the lottery mechanism is an n approximation to the highest agent value.*

If instead it is possible to charge payments, such payments, if made proportionally to the agents' bids, could discourage low-valued agents from making high bids. This sort of dynamic allocation and pricing mechanism is referred to as an *auction*.

Definition 1.3 A *Single-item auction* is a solution to the single-item allocation problem that solicits bids, picks a winner, and determines payments.

A natural allocation and pricing rule that is used, e.g., in government procurement auctions, is the *first-price auction*.

Definition 1.4 The *first-price auction* is:

 (i) ask agents to report their values ($\Rightarrow$ agent i reports b_i),
 (ii) select the agent $i^\star$ with highest report ($\Rightarrow i^\star = \operatorname{argmax}_i b_i$),
(iii) allocate the item to agent $i^\star$, and
(iv) charge this winning agent her bid, $b_{i^\star}$.

To get some appreciation for the strategic elements of the first price auction, note that an agent who wins wants to pay as little as possible, thus bidding a low amount is desirable. Of course, if the agent bids too low, then she probably will not win. Strategically, this agent must figure out how to balance this tradeoff. Of course, since agents may not report their true values, the agent with the highest bid may not be the agent with the highest-valued message. See Figure 1.1.

We will be able to analyze the first-price auction and we will do so in Chapter 2. However, for two reasons, there is little hope of generalizing it beyond the single-network-link special case (i.e., to large asymmetric computer networks) which is our eventual goal. First, calculating equilibrium strategies in general asymmetric environments is not easy; consequently, there would be little reason to believe that agents would play by the equilibrium. Second, it would be a challenge to show that the equilibrium is any good. Therefore, we turn to auctions that are strategically simpler.

The ascending-price auction is a stylized version of the auction popularized by Hollywood movies; art, antiques, and estate-sale auction houses such as Sotheby's and Christie's; and Internet auction houses such as eBay.

Definition 1.5 The *ascending-price auction* is:[1]

---

[1] The ascending-price auction is also referred to as the *English auction* and it contrasts to the Dutch (descending-price) auction.
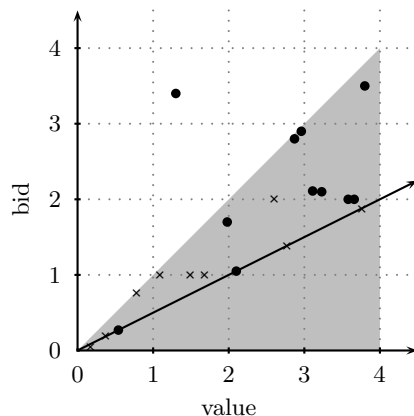
Figure 1.1 An in-class experiment: 21 student were endowed with values uniformly drawn from the interval $[0, 4]$ (denoted as $v_i \sim U[0, 4]$), they were told their own values and that the distribution of values was $U[0, 4]$, they were asked to submit bids for a two-agent one-item first-price auction. The bids of the students were collected and randomly paired for each auction; the winner was paid the difference between his value and his bid in dollars (real money). Winning bids are shown as "●" and losing bids are shown as "✕". The grey area denotes strategies that are not dominated. The black line $b = v/2$ denotes the equilibrium strategy in theory. In economic experiments, just like our in class experiment, bidders tend to overbid the equilibrium strategy. A few students knew the equilibrium strategy in advance of the in-class experiment.

  (i) gradually raise an offered price up from zero,
 (ii) allow agents to drop out when they no longer wish to win at the offered price,
(iii) stop at the price where the second-to-last agent drops out, and
(iv) allocate the item to the remaining agent and charges her the stopping price.

Strategically this auction is much simpler than the first-price auction. What should an agent with value v do? A good strategy would be "drop when the price exceeds v." Indeed, regardless of the actions of the other agents, this is a good strategy for the agent to follow, i.e., it is a *dominant strategy*. It is reasonable to assume that an agent with an obvious dominant strategy will follow it.

Since we know how agents are behaving, we can now make conclusions as to what happens in the auction. The second-highest-valued agent will drop out when the ascending prices reaches her value, $v_{(2)}$. The highest-valued agent will win the item at this price. We can conclude that this

auction maximizes the *social surplus*, i.e., the sum of the utilities of all parties. Notice that the utility of losers are zero, the utility of the winner is $v_{(1)} - v_{(2)}$, and the utility of the seller (e.g., the router in the congestion control application) is $v_{(2)}$, the payment received from the winner. The total is simply $v_{(1)}$, as the payment occurs 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.

Theorem 1.2   *The ascending-price auction maximizes the social surplus in dominant strategy equilibrium.*

What is striking about this result is that it shows that there is essentially no loss in surplus imposed by the assumption that the agents' values are privately known only to themselves. Of course, we also saw that the same was not true of routing mechanisms that cannot require the winner to make a payment in the form of a monetary *transfer* from the winner to the seller. Recall, the lottery mechanism could be as bad as an n approximation. A conclusion we should take from this exercise is that transfers are very important for surplus maximization when agents have private values.

Unfortunately, despite the good properties of the ascending-price auction there are two drawbacks that will prevent our using it for routing and congestion control in computer networks. First, mechanisms for sending messages in computer networks must be very fast. Ascending auctions are slow and, thus, impractical. Second, the ascending-price auction does not generalize to give a routing mechanisms in networks beyond the single-network-link special case. Challenges arise because ascending prices would not generally find the social surplus maximizing set of messages to route. A solution to these problems comes from Nobel laureate William Vickrey who observed that if we simulate the ascending-price auction with sealed bids we arrive at the same outcome in equilibrium without the need to bother with an ascending price.

Definition 1.6   The *second-price auction* is:[2]

(i)   accept sealed bids,
(ii) allocate the item to the agent with the highest bid, and
(iii) charge this winning agent the second-highest bid.

---

[2]   The second-price auction is also referred to as the Vickrey auction.

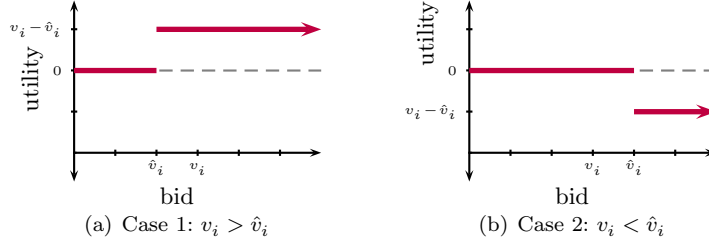(a) Case 1: $v_i > \hat{v}_i$ (b) Case 2: $v_i < \hat{v}_i$

Figure 1.2 Utility as a function of bid in the second-price auction.

In order to predict agent behavior in the second-price auction, notice that its outcome can be viewed as a simulation of the ascending-price auction. Via this viewpoint, there is a one-to-one correspondence between bidding b in the second-price auction and dropping out at price b is the ascending-price auction. Since the dominant strategy in the ascending-price auction is for an agent to drop out at when the price exceeds her value; it is similarly a dominant strategy for the agent to bid her true value in the second-price auction. 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. Fix the bids of all other agents and let $\hat{v}_i = \max_{j \neq i} v_j$. Notice that given this $\hat{v}_i$ there are only two possible outcomes for agent i. If she bids $b_i > \hat{v}_i$ then she wins, pays $\hat{v}_i$ (which is the second-highest bid), and has utility $u_i = v_i - \hat{v}_i$. On the other hand, if she bids $b_i < \hat{v}_i$ then she loses, pays nothing, and has utility $u_i = 0$. This analysis allows us to plot the utility of agent i as a function of her bid in two relevant cases, the case that $v_i < \hat{v}_i$ and the case that $v_i > \hat{v}_i$. See Figure 1.2.

Agent i would like to maximize her utility. In Case 1, this is achieved by any bid greater than $\hat{v}_i$. In Case 2, it is achieved by any bid less than $\hat{v}_i$. 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 $\hat{v}_i$. Thus, bidding truthfully, i.e., $b_i = v_i$, is a dominant strategy. $\qquad\square$

Notice that, in the proof of the theorem, $\hat{v}_i$ is the infimum of bids

that the bidder can make and still win, and the price charge to such a winning bidder is exactly $\hat{v}_i$. We henceforth refer to $\hat{v}_i$ as agent i's *critical value*. It should be clear that the proof above can be easily generalized, in particular, to any auction where each agent faces such a critical value that is a function only of the other agents' reports. This observation will allow the second-price auction to be generalized beyond single-item environments.

Corollary 1.4 *The second-price auction maximizes the social surplus in dominant strategy equilibrium.*

Proof By the definition of the second-price auction, the agent with the highest bid wins. By Theorem 1.3 is a dominant strategy equilibrium for agents to bid their true values. Thus, in equilibrium the agent with the highest bid is identically the agent with the highest value. The social surplus is maximized. □

In the remainder of this section we explore a number of orthogonal issues related to practical implementations of routing and congestion control. Each of these vignettes will conclude with motivating questions that will be addressed in the subsequent chapters. First, we address the issue of payments. The routing protocol in today's Internet, for instance, does not allow the possibility of monetary payments. How does the routing problem change if we also disallow monetary payments? The second issue we address is speed. While the second-price auction is faster than the ascending-price auction, still the process of soliciting bids, tallying results, and assigning payments may be too cumbersome for a routing mechanism. A simpler posted-pricing mechanism would be faster, but how can we guarantee good performance with a posted pricing? Finally, the single-link case is far from providing a solution to the question of routing and congestion control in general networks. How can we extend the second-price auction to more general environments?

1.1.1 Non-monetary payments

Most Internet mechanisms, including its congestion control mechanisms, do not currently permit monetary transfers. There are historical, social, and infrastructural reasons for this. The Internet was initially developed as a research platform and its users were largely altruistic. Since its development, the social norm is for Internet resources and services to be free and unbiased. Indeed, the "net neutrality" debates of the early

2000's were largely on whether to allow differentiated service in routers based on the identity of the source or destination of messages (and based on contracts that presumably would involve payments). Finally, micropayments in the Internet would require financial infrastructure which is currently unavailable at reasonable monetary and computational overhead.

One solution that has been considered, and implemented (but not widely adopted) for similar resource allocation tasks (e.g., filtering unsolicited electronic mail, a.k.a., spam) is *computational payments* such as "proofs of work." With such a system an agent could "prove" that her message was high-valued by having her computer perform a large, verifiable, but otherwise, worthless computational task. Importantly, unlike monetary payments, computational payments would not represent utility transferred from the winner to the router. Instead, computational payments are utility lost to society.

The *residual surplus* of a mechanism with computational payments is the total value generated less any payments made. The residual surplus for a single-item auction is thus the value of the winner less her payment. For the second-price auction, the residual surplus is $v_{(1)} - v_{(2)}$. For the lottery, the residual surplus is $\frac{1}{n} \sum_i v_i$, which is the same as the surplus as there are no payments.

While the second-price auction maximizes surplus (among all mechanisms) regardless of the values of the agents, for the objective of residual surplus it is clear that neither the second-price auction nor the lottery mechanism is best regardless of agent values. Consider the bad input for the lottery, where $v_1 = 1$ and $v_i = \epsilon$ (for $i \neq 1$). If we let ϵ go to zero, the second-price auction has residual surplus $v_{(1)} = 1$ (which is certainly optimal) and the lottery has expected surplus $1/n$ (which is far from optimal). On the other hand, if we consider the all-ones input, i.e., $v_i = 1$ for all i, then the residual surplus of the second-price auction is $v_{(1)} - v_{(2)} = 0$ (which is far from optimal), whereas the lottery surplus is $v_{(1)} = 1$ (which is clearly optimal). Of course, on the input with $v_1 = v_2 = 1$ and $v_i = \epsilon$ (for $i \geq 3$) both the lottery and the second-price auction have residual surplus far from what we could achieve if the values were publicly known or monetary transfers were allowed.

The underlying fact in the above discussion that separates the objectives of surplus and residual surplus is that for surplus maximization there is a single mechanism that is optimal for any profile of agent values, namely the second-price auction; whereas there is no such mechanism for residual surplus. Since there is no absolute optimal mechanism we must

trade-off performance across possible profiles of agent values. There are two ways to do this. The first approach is to assume a distribution over value profiles and then optimize residual surplus in expectation over this distribution. Thus, we might trade off low residual surplus on a rare input for high residual surplus on a likely input. This approach results in a different "optimal mechanism" for different distributions. The second approach begins with the solution to the first approach and asks for a single mechanism that bests approximates the optimal mechanism in worst-case over distributions. This second approach may be especially useful for applications of mechanism design to computer networks because it is not possible to change the routing protocol to accommodate changing traffic workloads.

Question 1.1   In what settings does the second-price auction maximize residual surplus? In what settings does the lottery maximize residual surplus?

Question 1.2   For any given distribution over agent values, what mechanism optimizes residual surplus for the distribution?

Question 1.3   If the optimal mechanism for a distribution is complicated or unnatural, is there a simple or natural mechanism that approximates it?

Question 1.4   In worst-case over distributions of agent values, what single mechanism best approximates the optimal mechanism for the distribution?

1.1.2 Posted Pricing

Consider again the original single-item allocation problem to maximize surplus (with monetary payments). Unfortunately, even a single-round, sealed-bid auction such as the second-price auction may be too complicated and slow for congestion control and routing applications. An even simpler approach would be to just post a take-it-or-leave-it price. Consider the following mechanism.

Definition 1.7   For a given price $\hat{v}$, the *uniform-pricing* mechanism serves the first agent willing to pay $\hat{v}$ (breaking ties in arrival order randomly).

For instance, if we assumed all agents arrive at once and $\hat{v} = 0$ this uniform pricing mechanism is identical to the aforementioned lottery.

Recall that the lottery mechanism is very bad when there are many low-valued agents and a few high-valued agents. The bad example had one agent with value one, and the remaining $n-1$ agents with value ϵ. This uniform-pricing mechanism, however, is more flexible. For instance, for this example we could set $\hat{v} = 2\epsilon$, only the high-valued agent will want to buy, and the surplus would be one. Such a posted-pricing mechanism is very practical and, therefore, especially appropriate for our application to Internet routing.

Of course, the price $\hat{v}$ needs to be chosen well. Fortunately in the routing example where billions of messages are sent every day, it is reasonable to assume that there is some distributional knowledge of the demand. Imagine that the value of each agent i is drawn independently and identically from distribution F. The *cumulative distribution function* for random variable v drawn from distribution $\boldsymbol{F}$ specifies the probability that it is at most z, denoted $F(z) = \mathbf{Pr}_{v \sim F}[v < z]$. For example the uniform distribution on interval $[0, 1]$ is denoted $U[0, 1]$ and its cumulative distribution function is $F(z) = z$.

There is a very natural way to choose $\hat{v}$: mimic the outcome of the second-price auction as much as possible. Notice that with n identically distributed agents, the *ex ante* (meaning: before the values are drawn) probability that any particular agent wins is $1/n$. To mimic the outcome of the second-price auction on any particular agent we could set a price $\hat{v}$ so that the probability that the agent's value is above $\hat{v}$ is exactly $1/n$, this price can be found by inverting the cumulative distribution function as $\hat{v} = F^{-1}(1 - 1/n)$. For the uniform distribution, the solution to this inverse is $\hat{v} = 1 - 1/n$. Unlike the second-price auction, posting a uniform price of $\hat{v}$ may result in no winners (if all agent values are below $\hat{v}$) or an agent other than that with the highest value may win (if there are more than one agents with value above $\hat{v}$).

Theorem 1.5 *For values drawn independently and identically from any distribution F, the uniform pricing of $\hat{v} = F^{-1}(1 - 1/n)$ is an $e/e-1 \approx 1.58$ approximation to the optimal social surplus.*

Proof The main idea of this proof is to compare three mechanisms. Let REF denote the second-price auction and its surplus (our reference mechanism). Let APX denote the uniform pricing and its surplus (our approximation mechanism). The second-price auction, REF, optimizes surplus, subject to the *ex post* (meaning: after the mechanism is run) *supply constraint* that at most one agent wins, and chooses to sell to each agent with ex ante probability $1/n$. Consider for comparison a third

mechanism UB that maximizes surplus subject to the constraint that each agent is served with ex ante probability at most $1/n$, but has no supply constraint, i.e., UB can serve multiple agents if it so chooses.

The first step in the proof is the simple observation that UB upper bounds REF, i.e., UB $\geq$ REF. This is clear as both mechanisms serve each agent with ex ante probability $1/n$, but REF has an ex post supply constraint whereas UB does not. UB could simulate REF and get the exact same surplus, or it could do something even better. Conclude,

$$\text{UB} \geq \text{REF}. \tag{1.1}$$

In fact, UB will do something better than REF. First, observe that UB's optimization is independent between agents. Second, observe that the socially optimal way to serve an agent with ex ante probability $1/n$ is to offer her price $\hat{v} = F^{-1}(1 - 1/n)$. We now wish to calculate UB's *expected surplus*. Let $\mathbf{E}[v \mid v \geq \hat{v}]$ denote the expected value of an agent given that her value v is above the price $\hat{v}$. If we sell to an agent and all we know is that her value is above the price, this quantity is the expected surplus generated. By the choice of price $\hat{v}$, the probability than an agent has a value v that exceeds the price $\hat{v}$ is $\mathbf{Pr}[v \geq \hat{v}] = 1/n$, and when an agent's value is below the price her surplus is zero. Thus, her (total) expected surplus in UB is exactly $\mathbf{E}[v \mid v \geq \hat{v}] \cdot \mathbf{Pr}[v \geq \hat{v}]$. By linearity of expectation, UB's (total) expected surplus is just the sum over the n agents of the surplus of each agent's surplus. Therefore,

$$\begin{aligned} \text{UB} &= n \cdot \mathbf{E}[v \mid v \geq \hat{v}] \cdot \mathbf{Pr}[v \geq \hat{v}] \\ &= \mathbf{E}[v \mid v \geq \hat{v}]. \end{aligned} \tag{1.2}$$

Finally, we get a lower bound on APX's surplus that we can relate to REF via its upper bound UB. Recall that the price in the uniform-pricing mechanism is selected so that the probability that any given agent has value exceeding the price is exactly $1/n$. The probability that there are no agents who are above the price is equal to the probability that all agents are below the price, which is equal to the product of the probabilities that each agent is below the threshold, i.e., $(1-1/n)^n \leq 1/e$.[3] Therefore, the probability that the item is sold by uniform pricing is at least $1 - 1/e$. If the item is sold, it is sold to an arbitrary agent with value conditioned to be at least $\hat{v}$, and the expected value of any such agent

---

[3] The *natural number* is $e \approx 2.178$. That $\lim_{n \to \infty} (1 - 1/n)^n = 1/e$ can be verified by taking the natural logarithm and applying L'Hopital's rule; the non-negativity of the derivative of $(1 - 1/n)^n$ implies it is is monotone non-decreasing; therefore, $1/e$ is an upper bound on $(1 - 1/n)^n$ for any finite n.

is $\mathbf{E}[v \mid v \geq \hat{v}]$. Therefore, the expected surplus of uniform pricing is,

$$\text{APX} \geq (1 - {}^{1}\!/_{e})\mathbf{E}[v \mid v \geq \hat{v}]. \tag{1.3}$$

Combining equations (1.1), (1.2), and (1.3) it is apparent that APX $\geq$ $(1 - {}^{1}\!/_{e})$ REF. $\square$

Question 1.5 When are simple, practical mechanisms like posted pricing a good approximation to the optimal mechanism?

1.1.3 General Routing Mechanisms

Finally we are ready to propose a mechanism for congestion control and routing in general networks. The main idea in the construction is the notion of critical values that was central to showing that the second-price auction has truthtelling as a dominant strategy (Theorem 1.3). In fact, that proof generalizes to any auction wherein each agent faces a critical value (that is not a function of her bid), the agent wins and pays the critical value if her bid exceeds it, and otherwise she loses.

Definition 1.8 The *second-price routing mechanism* is:

 (i) solicit sealed bids,
 (ii) find the set of messages that can be routed simultaneously with the largest total value, and
(iii) charge the agents of each routed message their critical values.

Theorem 1.6 *The second-price routing mechanism has truthful bidding as a dominant strategy.*

Corollary 1.7 *The second-price routing mechanism maximizes the social surplus.*

The proof of the theorem is similar to the analogous result for the second-price single-item auction, but we will defer its proof to Chapter 3. The corollary follows because the bids are equal to the agents' values, the mechanism is defined to be optimal for the reported bids, and the payments cancel.

Unfortunately, this is far from the end of the story. Step ((ii)) of the mechanism is known as *winner determination*. To understand exactly what is happening in this step we must be more clear about our model for routing in general networks. For instance, in the Internet, the route that messages take in the network is predetermined by the Border Gateway Protocol (BGP), which enforces that all messages routed to the same

destination through any given router follow the same path. There are no opportunities for load-balancing, i.e., for sending messages to the same destination across different paths so as to keep the loads on any given path at a minimum. Alternatively, we could be in a novel network where the routing can determine which messages to route and which path to route them on.

Once we fix a model, we need to figure out how to solve the optimization problem implied by winner determination. Namely, how do we find the subset of messages with the highest total value that can be simultaneously routed? In principle, we are searching over subsets that meet some complicated feasibility condition. Purely from the point of optimization, this is a challenging task. The problem is related to the infamous *disjoint paths* problems: given a set of pairs of vertices in a graph, find a subset of pairs that can be connected via disjoint paths. This problem is *NP hard* to solve. Meaning: it is at least as hard as any problem in the equivalence class of *NP-complete* problems for which it is widely believed that finding optimal solutions is computationally intractable.

Theorem 1.8 *The disjoint-paths problem is NP hard.*

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 routing mechanism as a solution to the general network routing problem.

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

Question 1.6 Can we replace Step ((ii)) in the mechanism with an approximation algorithm and still retain the dominant-strategy incentive property?

Question 1.7 If not, can we (by some other method) design a computationally tractable approximation mechanism for routing?

Question 1.8 Is there a general theory for designing approximation mechanisms from approximation algorithms?

1.2 Mechanism Design

Mechanism design gives a theory for the design of protocols, services,

laws, or other "rules of interaction" in which selfish behavior leads to good outcomes. "Selfish behavior" means that each participant, hereafter *agent*, individually tries to maximize her own utility. Such behavior we define as rational. "Leads" means *in equilibrium*. A set of agent strategies is in equilibrium if no agent prefers to unilaterally change her strategy. Finally, the "good"-ness of an outcome is assessed with respect to the criteria or goals of the designer. Natural economic criteria are *social surplus*, the sum of the utilities of all parties; and *profit*, the total payments made to the mechanism less any cost for providing the outcome.

A theory for mechanism design should satisfy the following four desiderata:

Informative: It pinpoints salient features of the environment and characteristics of good mechanisms therein.

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

Predictive: The mechanisms that the theory predicts should be the same as the ones observed in practice.

Tractable: The theory should not assume super-natural ability for the agents or designer to optimize.

Notice that optimality is not one of the desiderata, nor is suggesting a specific mechanism to a practitioner. Instead, intuition from the theory of mechanism design should help guide the design of good mechanisms in practice. Such guidance is possible through informative observations about what good mechanisms do. Observations that are robust to variations in modeling details are especially important.

Sometimes the theory of *optimal mechanism design* meets the above desiderata. The question of designing an optimal mechanism can be viewed as a standard optimization problem: given incentive constraints, imposed by game theoretic strategizing; feasibility constraints, imposed by the environment; and the distribution of agent preferences, optimize the designer's given objective. In ideal environments the given constraints may simplify and, for instance, allow the mechanism design problem to be reduced to a natural optimization problem without incentive constraints or distribution. We saw an example of this for routing in general networks: in order to invoke the second-price mechanism we only needed to find the optimal set of messages to route. Unfortunately, there are many environments and objectives where the optimal mechanism design problem not simplify as nicely.

1.3 Approximation

In environments where optimal mechanisms do not meet the desiderata above, approximation can provide a remedy. In the formal definition of an approximation, below, a good mechanism is one with a small approximation factor.

Definition 1.9 For an environment given implicitly, denote an *approximation mechanism* and its performance by APX, and a *reference mechanism* and its performance by REF.

(i) For any environment, APX is a β *approximation* to REF if APX $\geq \frac{1}{\beta}$ REF.

(ii) For any class of environments, a class of mechanisms is a β approximation to REF if for any environment in the class there is a mechanism APX in the class that is a β approximation to REF.

(iii) For any class of environments, a mechanism APX is a β approximation to REF if for any environment in the class APX is a β approximation to REF.

In the preceding section we saw each of these types of approximation. For i.i.d. $U[0,1]$, n-agent, single-item environments, posting a uniform price of $\hat{v} = 1 - 1/n$ is a $e/e-1$ approximation to the second-price auction. More generally, for any i.i.d. single-item environment, uniform pricing is a $e/e-1$ approximation to the second-price auction. Finally, for any single-item environment the lottery gives an n approximation to the social surplus of the second-price auction.

Usually we will employ the approximation framework with REF representing the optimal mechanism. For instance, in the preceding section we compared a posted-pricing mechanism to the surplus-optimal second-price auction for i.i.d., single-item environments. For such a comparison, clearly REF $\geq$ APX, and therefore the approximation factor is at least one. It is often instructive to compare the approximation ability of one class of mechanisms to another. For instance, in the preceding section we compared the surplus of a lottery, as the optimal mechanism without payments, to the surplus of the second-price auction, the optimal mechanism (in general). This kind of apples-to-oranges comparison is useful for understanding the relative importance of various features of a mechanism or environment.

1.3.1 Philosophy of Approximation

While it is, no doubt, a compelling success of the theory of mechanism design that its mechanisms are so prevalent in practice, optimal mechanism design cannot claim the entirety of the credit. These mechanisms are employed by practitioners well beyond the environments for which they are optimal. Approximation can explain why: the mechanisms that are optimal in ideal environments may continue to be approximately optimal much more broadly. It is important for the theory to describe how broadly these mechanisms are approximately optimal and how close to optimal they are. Thus, the theory of approximation can complement the theory of optimality and justify the wide prevalence of certain mechanisms. For instance, in Chapter 4 and Chapter 7 we describe how the widely prevalent reserve-price-based mechanisms and posted pricings are corroborated by their approximate optimality.

There are natural environments for mechanism design wherein every "undominated" mechanism is optimal. If we consider only optimal mechanisms we are stuck with the full class from which we can make no observations about what makes a mechanism good; on the other hand, if we relax optimality, we may be able to identify a small subclass of mechanisms that are approximately optimal, i.e., for any environment there is a mechanism in the subclass that approximates the optimal mechanism. This subclass is important in theory as we can potentially observe salient characteristics of it. It is important in practice because, while it is unlikely for a real mechanism designer to be able to optimize over all mechanisms, optimizing over a small class of, hopefully, natural mechanisms may be possible. For instance, a conclusion that we will make precise in Chapter 4 and Chapter 7 is that reserve-price-based mechanisms and posted pricings are approximately optimal in a wide range of environments including those with multi-dimensional agent preferences.

Approximation provides a lens with which to explore the salient features of an environment or mechanism. Suppose we wish to determine whether a particular feature of a mechanism is important. If there exists a subclass of mechanisms without that feature that gives a good approximation to the optimal mechanism, then the feature is perhaps not that important. If, on the other hand, there is no such subclass then the feature is quite important. For instance, previously in this chapter we saw that mechanisms without transfers cannot obtain better than a linear approximation to the optimal social surplus in single-item environments. This result suggests that transfers are very important for mechanism
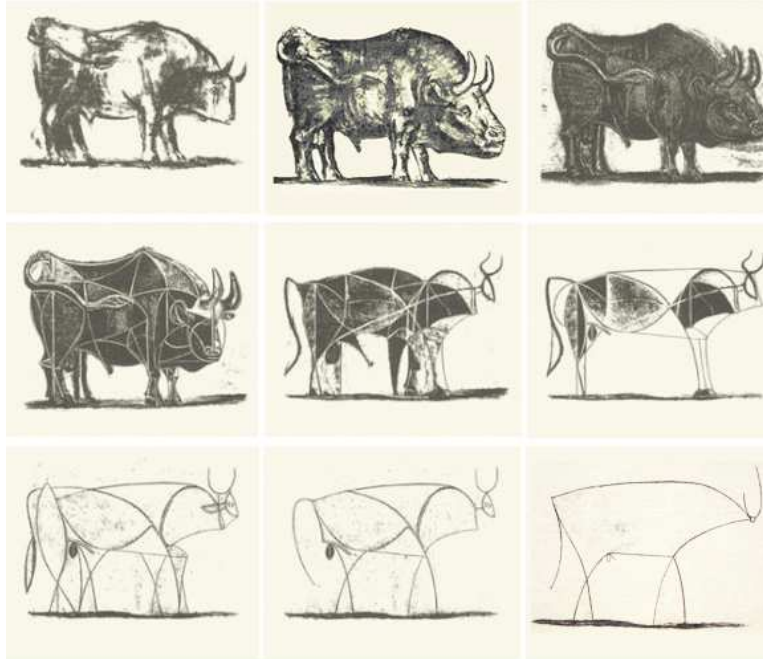
Figure 1.3 Picasso's December, 1945 to January, 1946 abstractionist study of a bull highlights one of the main points of approximation: identifying the salient features of the object of study. Picasso drew these in order from left to right, top to bottom.

design. On the other hand, we also saw that posted-pricing mechanism could obtain an $e/{e-1}$ approximation to the surplus-optimal mechanism. Posted pricings do not make use of competition between agents, therefore, we can conclude that competition between agents is not that important. Essentially, approximation provides a means to determine which aspect of an environment are details and which are not details. The approximation factor quantifies the relative importance on the spectrum between unimportant details to salient characteristics. Approximation, then allows for design of mechanisms that are not so dependent on details of the setting and therefore more robust. See Figure 1.3 for an illustration of this principle. In particular, in Chapter 4 we will formally observe that revenue-optimal auctions when agent values are drawn from a distribution can be approximated by a mechanism in which the only distributional dependence is a single number; moreover, in Chapter 5 we

will observe that some environments permit a single (prior-independent) mechanism to approximate the revenue-optimal mechanism under any distributional assumption.

Suppose the seller of an item is worried about collusion, risk attitudes, after-market effects, or other economic phenomena that are usually not included in standard ideal models for mechanism design. One option would be to explicitly model these effects and study optimal mechanisms in the augmented model. These complicated models are difficult to analyze and optimal mechanisms may be overly influenced by insignificant-seeming modeling choices. Optimal mechanisms are precisely tuned to details in the model and these details may drive the form of the optimal mechanism. On the other hand, we can consider approximations that are robust to various out-of-model phenomena. In such an environment the comparison between the approximation and the optimal mechanism is unfair because the optimal mechanism may suffer from out-of-model phenomena that the approximation is robust to. In fact, this "optimal mechanism" may perform much worse than our approximation when the phenomena are explicitly modeled. For example, Chapter 4 and Chapter 7 describe posted pricing mechanisms that are approximately optimal and robust to timing effects; for this reason an online auction house, such as eBay, may prefer its sellers to use "buy it now" posted pricings instead of auctions.

Finally, there is an issue of non-robustness that is inherent in any optimization over a complex set of objects, such as mechanisms. Suppose the designer does not know the distribution of agent preferences exactly but can learn about it through, e.g., market analysis. Such a market analysis is certainly going to be noisy; exactly optimizing a mechanism to the market analysis may "over fit" to this noise. Both statistics and machine learning theory have techniques for addressing this sort of overfitting. Approximation mechanisms also provide such a robustness. Since the class of approximation mechanisms is restricted from the full set, for these mechanisms to be good, they must pay less attention to details and therefore are robust to sampling noise. Importantly, approximation allows for design and analysis mechanisms for small (a.k.a., *thin*) markets where statistical and machine learning methods are less applicable.

1.3.2 Approximation Factors

Depending on the problem and the approximation mechanism, approximation factors can range from $(1 + \epsilon)$, i.e., arbitrarily close approxima-

tions, to linear factor approximations (or sometimes even worse). Notice a linear factor approximation is one where, as some parameter in the environment grows, i.e., more agents or more resources, the approximation factor gets worse. As examples, we saw earlier an environment in which uniform pricing is a *constant approximation* and the lottery is a *linear approximation.*[4]

In this text we take constant versus super-constant approximation as the separation between good and bad. We will view a proof that a mechanism is a constant approximation as a positive result and a proof that no mechanism (in a certain class) is a constant approximation as a negative result. Constant approximations tend to represent a tradeoff between simplicity and optimality. Properties of constant approximation mechanisms can, thus, be quite informative. Of course, there are many non-mechanism-design environments where super-constant approximations are both useful and informative; however, for mechanism design super-constant approximations tend to be indicative of (a) a bad mechanism, (b) failure to appropriately characterize optimal mechanisms, or (c) an imposition of incompatible modeling assumptions or constraints.

If you were approached by a seller (henceforth: principal) to design a mechanism and you returned to triumphantly reveal an elegant mechanism that gives her a two approximation to the optimal profit, you would probably find her a bit discouraged. After all, your mechanism leaves half of her profit on the table. In the context of this critique we outline the main points of constant, e.g., two, approximations for the practitioner. First, a two approximation provides informative conclusions that can guide the design of even better mechanisms for specific environments. Second, the approximation factor of two is a theoretical result that holds in a large range of environments, in specific environments 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 environments the optimal mechanism is not understood at all, meaning the principal's alternative to your two approximation is an ad hoc mechanism with no performance guarantee. This principal is of course free to simulate your mechanism and her mechanism in her given environment and decide to use the bet-

---

[4] Recall that the approximation factor for uniform pricing bounded by $e/e-1$, an absolute constant that does not increase with various parameters of the auction such as the number of agents. In contrast the approximation factor of the lottery could be as bad as n, the number of agents. As the number of agents increases, so does the approximation bound guaranteed by the lottery.

ter of the two. In this fashion the principal's ad hoc mechanism, if used, is provably a two approximation as well. Fourth, mechanisms that are two approximations in theory arise in practice. In fact, that it is a two approximation explains why the mechanism arises. Even though it is not optimal, it is close enough. If it was far from being optimal the principal (hopefully) would have figured this out and adopted a different approach.

Sometimes it is possible do obtain schemas for approximating the optimal mechanism to within a $(1+\epsilon)$ factor for any ϵ. These schemas tend to be computational approaches that are useful for addressing potential computational intractability of the optimal mechanism design problem. While they do not tend to yield simple mechanisms, they are relevant in complex environments. Often these approximation schemes are based on (a) identifying a restricted class of mechanisms wherein a near-optimal mechanism can be found and (b) conducting a brute-force search over this restricted class. While very little is learned from such a brute-force search, properties of the restricted class of mechanisms can be informative. Many of the optimal mechanisms we describe can in practice only be implemented as approximation schemes.

Chapter Notes

Routing and congestion control are a central problems in computer systems such as the Internet; see Leiner et al. (1997) for a discussion of design criteria. Demers et al. (1989) analyze "fair queuing" which is a lottery-based mechanism for congestion control. Griffin et al. (2002) discuss the Border Gateway Protocol (BGP) which determines the routes messages take in the Internet. The NP-completeness of the disjoint paths problem (and the related problem of integral multi-commodity flow) was established by Even et al. (1976).

William Vickrey's 1961 analysis of the second-price auction is one of the pillars of mechanism design theory. The second-price routing mechanism is a special case of the more general Vickrey-Clarke-Groves (VCG) mechanism which is attributed additionally to Edward Clarke (1971) and Theodore Groves (1973).

Computational payments were proposed as means for fighting unsolicited electronic mail by Dwork and Naor (1992). Hartline and Roughgarden (2008) consider mechanism design with the objective of residual surplus and describe distributional assumptions under which the lottery

is optimal, the second-price auction is optimal, and when neither are optimal. They also give a single mechanism that approximates the optimal mechanism for any distribution of agent values.

Vincent and Manelli (2007) showed that there are environments for mechanism design wherein every "undominated" mechanism is optimal for some distribution of agent preferences. This result implies that optimality cannot be used to identify properties of good mechanisms. Robert Wilson (1987) suggested that mechanisms that are less dependent on the details of the environment are likely to be more relevant. This suggestion is known as the "Wilson doctrine."

The $e/e-1$ approximation via a uniform pricing (Theorem 1.5) is a consequence of Chawla et al. (2010). Wang et al. (2008) and Reynolds and Wooders (2009) discuss why the "buy it now" (i.e., posted-pricing) mechanism is replacing the second-price auction format in eBay.