Approximation in Economic Design

Jason D. Hartline[1]

Draft: April 15, 2012

[1]Northwestern University, Evanston, IL 60208. Email: hartline@eecs.northwestern.edu.

Contents

Author's Note

This text is suitable for advanced undergraduate or graduate courses; it has been developed at Northwestern U. as the primary text for such a course since 2008.

This text provides a look at select topics in economic mechanism design through the lens of approximation. It reviews the classical economic theory of mechanism design wherein a Bayesian designer looks to find the mechanism with optimal performance in expectation over the distribution from which the preferences of the participants are drawn. It then adds to this theory practical constraints such as simplicity, tractability, and robustness. The central question addressed is whether these practical mechanisms are good approximations of the optimal ones. The resulting theory of approximation in mechanism design is based on results that come mostly from the theoretical computer science literature. The results presented are the ones that are most directly compatible with the classical (Bayesian) economic theory and are not representative of the entirety of the literature.

– Jason D. Hartline

Chapter 1

Approximation and Mechanism Design

Our world is an interconnected collection of economic and computational systems wherein individuals optimize to achieve their own, perhaps selfish, goals subject to basic laws of the system. 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 exchanging goods or services that are good in terms of revenue or social benefit.

The two sources for economic considerations are the preferences for 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 result of computational tractability or due to desired robustness, 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 in will be a tradeoff between optimality and other desirable properties such as simplicity, robustness, computational tractability, and practicality. This tradeoff will be quantified by a theory for approximation which measures the loss of a simple, robust, and practical approximation 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 test. 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 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 and approximation in mechanism design.

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?

There is nothing special about the fact that the resource that the users (henceforth: agents) are vying for is a network link; 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 that receives the item.*

The social surplus is maximized if the item is allocated to agent with the highest value, denoted $v_{(1)}$. If the values of the agent are publicly 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 an single-item allocation mechanism.

1. Ask agents to report their values. ($\Rightarrow$ agent i reports b_i)
2. Select the agent with highest report. ($\Rightarrow i^* = \operatorname{argmax}_i b_i$)
3. Allocate the item to agent i^*.

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. Clearly, we will not be able to say nice things about this mechanism. 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.

Mechanism 1.1 (Lottery)**.**

1. *Select a uniformly random agent.*
2. *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 $\frac{1}{n}\sum_i v_i$. 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 1.2. *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 (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. A single-item auction 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.*

Mechanism 1.2 (First-price Auction)**.**

1. *Ask agents to report their values. ($\Rightarrow$ agent i reports b_i)*
2. *Select the agent with highest report. ($\Rightarrow i^* = \operatorname{argmax}_i b_i$)*
3. *Allocate the item to agent i^*.*
4. *Charge this agent her bid, b_{i^*}.*

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, so bidding a low amount is desirable. Of course, if they bid too low, then they 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,

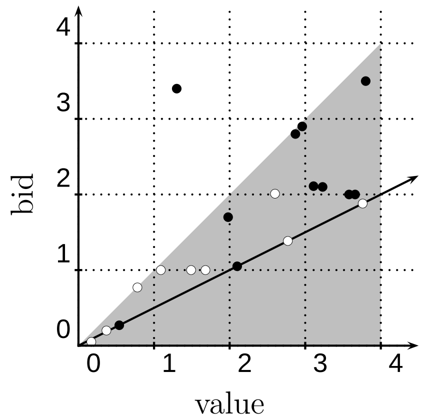


Figure 1.1: An in-class experiment: 21 student were endowed with values from $U[0,4]$, they were told that the numbers were $U[0,4]$, they were asked to submit bids for a 2-agent 1-item first-price auction. The bids of the students were collected and randomly paired for each auction; the winner was paid the difference between his value and his bid in dollars. Winning bids are shown as "•" and losing bids are shown as "∘". The grey area denotes strategies that are not dominated. The 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.

it would be a challenge to show that the equilibrium is any good. Therefore, we turn to auctions that are strategically simpler.

The English auction is a stylized version of the ascending-price 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.

Mechanism 1.3 (English Auction)**.**

1. *Gradually raise an offer price up from zero.*
2. *Allow agents 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 agent left.*
4. *Allocate the item to this remaining agent and charge 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 1.4. *The English 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 each agent. 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 was as bad as an n-approximation. A conclusion we should make from this exercise is that transfers are very important for surplus maximization when agents have private values.

Unfortunately, despite the good properties of the English auction there are two drawbacks that will prevent our using it for routing and congestion control in computer networks. First, mechanisms for setting messages in computer networks must be very fast. Ascending auctions are slow and, thus, impractical. Second, the English 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 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. *Allocate the item to the agent with the highest bid.*

3. *Charge this agent the second-highest bid.*

In order to predict agent behavior in the second-price auction, notice that its outcome can be viewed as a simulation of the English auction. Since the dominant strategy in the English auction is to "drop at your value" then the only way a bidder could achieve the same outcome in the simulation is to input her 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 1.5. *Truthful bidding is a dominant strategy in the second-price auction.*

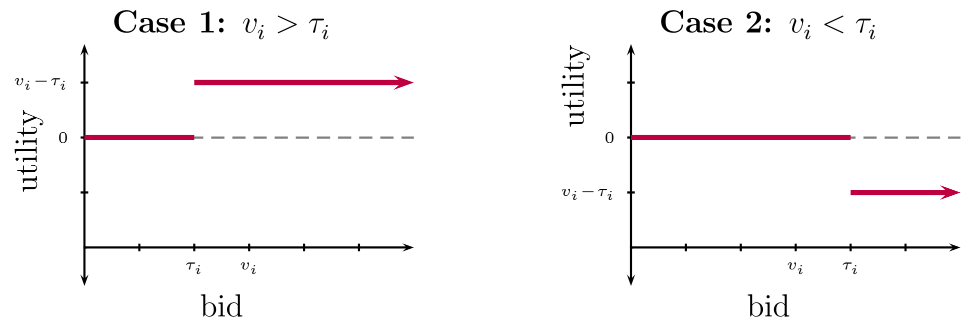


Figure 1.2: Utility as a function of bid in the second-price auction.

Proof. We show that truthful bidding is a dominant strategy for agent i. Fix the bids of all other agents and let $\tau_i = \max_{j \neq i} v_j$. Notice that given this τ_i there are only two possible outcomes for agent i. If she bids $b_i > \tau_i$ then she wins, pays τ_i (which is the second-highest bid), and has utility $u_i = v_i - \tau_i$. On the other hand, if she bids $b_i < \tau_i$ 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 $v_i < \tau_i$ and the case that $v_i > \tau_i$. See Figure 1.2.

Agent i would like to maximize her utility. In Case 1, this is achieved by any bid greater than τ_i. In Case 2, it is achieved by any bid less than τ_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 τ_i. Thus, bidding truthfully, i.e., $b_i = v_i$, is a dominant strategy. □

Notice that in the proof of the theorem τ_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 τ_i. We henceforth refer to τ_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.

In the remainder of this section we explore a number of orthogonal questions related to practical implementations of congestion control. We first address the issue of payments. The routing protocol in today's Internet, for instance, does not have 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 English 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 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 use 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. 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 maximized surplus (among all mechanisms) regardless of the values of the agents, for residual surplus it is clear neither the second-price auction nor the lottery 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.

An underling fact in the above discussion that separates the objectives of surplus and residual surplus is that for surplus maximization there is a single optimal mechanism for any profile of agent values, where as 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, one way to do it 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 an 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 when 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 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 use posted prices. Consider the following mechanism.

Mechanism 1.5 (Uniform Pricing)**.** *For a given price p, serve the first agent willing to pay p, ties can be broken randomly.*

For instance, if we assumed all agents arrive at once and $p = 0$ 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 $p = 2\epsilon$, 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 p 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. Denote the *cumulative distribution function* as $F(z) = \mathbf{Pr}_{v\sim F}[v < z]$. As an example, assume that F is the uniform distribution on interval $[0, 1]$, denoted $U[0, 1]$ and satisfying $F(z) = z$.

There is a very natural way to choose p: 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 p so that the probability that the agent's value is above p is exactly $1/n$, denoted $F^{-1}(1-1/n)$. For the uniform distribution $p = 1 - 1/n$. Unlike the second-price auction, posting a uniform price of p may result in no winners (if all agent values are below p) or an agent other than that with the highest value may win (if there are more than one agents with value above p).

Theorem 1.6. *For any i.i.d. distribution F, the uniform pricing of $p = F^{-1}(1-1/n)$ is an $\frac{e}{e-1} \approx 1.58$ 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 *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 B 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., B can serve multiple agents if it so chooses.

The first step in the proof is the simple observation that $B \geq \text{REF}$. This is clear as both mechanisms serve each agent with ex ante probability $1/n$, but REF has a supply constraint whereas B does not. B could simulate REF and get the exact same surplus, or it could do something even better.

In fact, B will do something better. First, observe that B's optimization is independent between agents. Second, observe that the socially optimal way to serve an agent with probability $1/n$ is to offer her price p. Mechanism B's surplus is just the sum over the n agents of the surplus from offering that agent a price of p. Therefore, $B = n \cdot \mathbf{E}[v \mid v \geq p] \cdot \mathbf{Pr}[v \geq p] = \mathbf{E}[v \mid v \geq p]$.

Finally, we get a bound on APX's surplus in terms of B; in particular, it gives an $\frac{e}{e-1}$-approximation to B. The probability that there are no agents are above the threshold in the uniform pricing mechanism is $(1-1/n)^n \leq 1/e$. 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 p, i.e., the expected value of this agent is $\mathbf{E}[v \mid v \geq p] = B \geq \text{REF}$. Therefore, the expected surplus of uniform pricing is $\text{APX} \geq (1 - 1/e)\ \text{REF}$. □

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 1.5). In fact, the proof generalizes to any auction wherein each agent faces a critical value that is a function of the other agents' reports, the agent wins and pays the critical value if her bid exceeds it, and otherwise she loses.

Mechanism 1.6 (Second-price Routing Mechanism)**.**

1. *Solicit sealed bids.*

2. *Find the set of messages that can be routed simultaneously with the largest total value.*

3. *Charge each routed message its critical value.*

Theorem 1.7. *The second-price routing mechanism has truthful bidding as a dominant strategy.*

Corollary 1.8. *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 payments cancel.

Unfortunately, this is far from the end of the story. Step 2 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 fine 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 1.9. *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 2 in the mechanism with an approximation algorithm and still retain the dominant-strategy incentive property?*

Question 1.7. *If not, can we design a computationally tractable approximation mechanism for routing?*

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 1.8. *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 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.

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 exactly suggesting a 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 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 simplify and, for instance, the mechanism design problem can 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 finding the optimal set of messages to route. Unfortunately, there are many environments and objectives where analysis has failed to simplify the problem and mechanism design for these environments is considered "unsolved."

1.3 Approximation

In environments 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. A good mechanism is one with a small approximation factor.

Definition 1.10. *For an environment given implicitly, denote an* approximation mechanism *and its performance by* APX, *and a* reference mechanism *and its performance by* REF.

1. *For any environment,* APX *is a* β-approximation *to* REF *if* APX $\geq$ REF $/\beta$.
2. *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.
3. *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 $p = 1-1/n$ is a $\frac{e}{e-1}$ approximation to the second-price auction. More generally, for any i.i.d., single-item environment uniform pricing is a $\frac{e}{e-1}$-approximation to the second-price auction. Finally, for any single-item environment the lottery is an n-approximation to 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 consider the approximation ability of one class of mechanisms to another. For instance, in the preceding section we compared 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 7 we describe how the wide prevalence reserve-price-based mechanisms and posted pricings is 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 Chapters 4 and 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 an 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 in single-item environments. This result suggests that transfers are very important for mechanism design. On the other hand, we also saw that posted-pricing could obtain an $\frac{e}{e-1}$-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 spectra 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 principal was 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

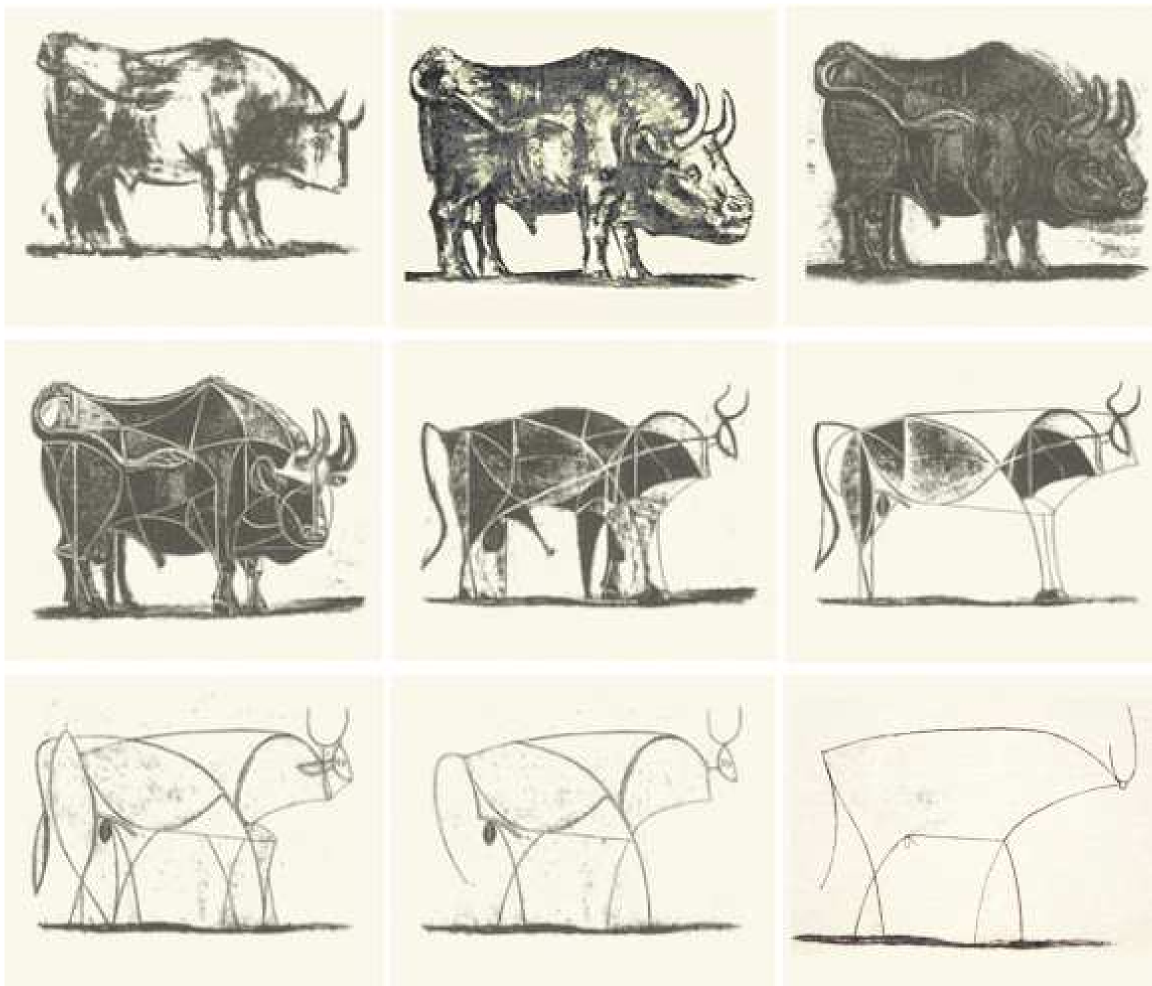


Figure 1.3: Picasso's December, 1945 to January, 1946 abstractionist study of a bull highlights one of the main points of approximation: identifying the salient features of the object of study. Picasso drew these in order from left to right, top to bottom.

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, Chapters 4 and 7 describe posted pricing mechanisms that are approximately optimal are robust to timing effects; for this reason an online auction house such eBay may want to sellers to switch from auctions to "buy it now" (a.k.a., posted) pricings.

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 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 and then exactly optimizing a mechanism to it 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 analysis of 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 approximations, 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.

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 2-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 2-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 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 her mechanism in her given environment and decide to use the better of the two. In this fashion the principal's ad hoc mechanism, if used, is provably a 2-approximation as well. Fourth, mechanisms 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.

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) brute-force search over this restricted class. While very little is learned from brute-force search, properties of the restricted class of mechanisms can be informative.

Chapter Notes

Routing and congestion control are a central problems in computer systems such as the Internet. Demers et al. (1989) analyze "fair queuing" which 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.

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 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."

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.

Chapter 2

Equilibrium

The theory of *equilibrium* attempts to predict what happens in a game when players behave strategically. This is a central concept to this text as, in mechanism design, we are optimizing over games to find the games with good equilibria. Here, we review the most fundamental notions of equilibrium. They will all be static notions in that players are assumed to understand the game and will play once in the game. While such foreknowledge is certainly questionable, some justification can be derived from imagining the game in a dynamic setting where players can learn from past play. Readers should look elsewhere for formal justifications.

This chapter reviews equilibrium in both complete and incomplete information games. As games of incomplete information are the most central to mechanism design, special attention will be paid to them. In particular, we will characterize equilibrium when the private information of each agent is single-dimensional and corresponds, for instance, to a value for receiving a good or service. We will show that auctions with the same equilibrium outcome have the same expected revenue. Using this so-called *revenue equivalence* we will describe how to solve for the equilibrium strategies of standard auctions in symmetric environments.

Emphasis is placed on demonstrating the central theories of equilibrium and not on providing the most comprehensive or general results. For that readers are recommended to consult a game theory textbook.

2.1 Complete Information Games

In games of compete information all players are assumed to know precisely the payoff structure of all other players for all possible outcomes of the game. A classic example of such a game is the *prisoner's dilemma*, the story for which is as follows.

> Two prisoners who have jointly committed a crime, are being interrogated in separate quarters. Unfortunately, the interrogators are unable to prosecute either prisoner without a confession. Each prisoner is offered the following deal: If she confesses and their accomplice does not, she will be released and her accomplice will serve the full sentence of ten years in prison. If they both confess, they will

> share the sentence and serve five years each. If neither confesses, they will both be prosecuted for a minimal offense and each receive a year of prison.

This story can be expressed as the following *bimatrix game* where entry (a, b) represents row player's payoff a and column player's payoff b.

	silent	confess
silent	(-1,-1)	(-10,0)
confess	(0,-10)	(-5,-5)

A simple thought experiment enables prediction of what will happen in the prisoners' dilemma. Suppose the row player is silent. What should the column player do? Remaining silent as well results in one year of prison while confessing results in immediate release. Clearly confessing is better. Now suppose that the row player confesses. Now what should the column player do? Remaining silent results in ten years of prison while confessing as well results in only five. Clearly confessing is better. In other words, no matter what the row player does, the column player is better of by confessing. The prisoners dilemma is hardly a dilemma at all: the *strategy profile* (confess, confess) is a *dominant strategy equilibrium*.

Definition 2.1. *A* dominant strategy equilibrium *(DSE) in a complete information game is a strategy profile in which each player's strategy is as least as good as all other strategies regardless of the strategies of all other players.*

DSE is a strong notion of equilibrium and is therefore unsurprisingly rare. For an equilibrium notion to be complete it should identify equilibrium in every game. Another well studied game is *chicken*.

> James Dean and Buzz (in the movie *Rebel without a Cause*) face off at opposite ends of the street. On the signal they race their cars on a collision course towards each other. The options each have are to swerve or to stay their course. Clearly if they both stay their course they crash. If they both swerve (opposite directions) they escape with their lives but the match is a draw. Finally, if one swerves and the other stays, the one that stays is the victor and the other the loses.[1]

A reasonable bimatrix game depicting this story is the following.

	stay	swerve
stay	(-10,-10)	(1,-1)
swerve	(-1,1)	(0,0)

Again, a simple thought experiment enables us to predict how the players might play. Suppose James Dean is going to stay, what should Buzz do? If Buzz stays they crash and Buzz's payoff is -10, but if Buzz swerves his payoff is only -1. Clearly, of these two options

[1] The actual chicken game depicted in *Rebel without a Cause* is slightly different from the one described here.

Buzz prefers to swerve. Suppose now that Buzz is going to swerve, what should James Dean do? If James Dean stays he wins and his payoff is 1, but if he swerves it is a draw and his payoff is zero. Clearly, of these two options James Dean prefers to stay. What we have shown is that the strategy profile (stay, swerve) is a mutual best response, a.k.a., a *Nash equilibrium.*

Definition 2.2. *A* Nash equilibrium *in a game of complete information is a strategy profile where each players strategy is a best response to the strategies of the other players as given by the strategy profile.*

In the examples above, the strategies of the players correspond directly to actions in the game, a.k.a., *pure strategies.* In general, Nash equilibrium strategies can be randomizations over actions in the game, a.k.a., *mixed strategies.*

2.2 Incomplete Information Games

Now we turn to the case where the payoff structure of the game is not completely known. We will assume that each agent has some private information and this information affects the payoff of this agent in the game. We will refer to this information as the agent's type and denote it by t_i for agent i. The profile of types for the n agents in the game is $\mathbf{t} = (t_1, \ldots, t_n)$.

A *strategy* in a game of incomplete information is a function that maps an agent's type to any of the agent's possible actions in the game (or a distribution over actions for mixed strategies). We will denote by $s_i(\cdot)$ the strategy of agent i and $\mathbf{s} = (s_1, \ldots, s_n)$ a *strategy profile.*

The auctions described in Chapter 1 were games of incomplete information where an agent's private type was her value for receiving the item, i.e., $t_i = v_i$. As we described, strategies in the English auction were $s_i(v_i) =$ "drop out when the price exceeds v_i" and strategies in the second-price auction were $s_i(v_i) =$ "bid $b_i = v_i$." We refer to this latter strategy as *truthtelling.* Both of these strategy profiles are in *dominant strategy equilibrium* for their respective games.

Definition 2.3. *A* dominant strategy equilibrium *(DSE) is a strategy profile* $\mathbf{s}$ *such that for all* i*,* t_i*, and* $\mathbf{b}_{-i}$ *(where* $\mathbf{b}_{-i}$ *generically refers to the actions of all players but* i*), agent* i*'s utility is maximized by following strategy* $s_i(t_i)$*.*

Notice that aside from strategies being defined as a map from types to actions, this definition of DSE is identical to the definition of DSE for games of complete information.

2.3 Bayes-Nash Equilibrium

Naturally, many games of incomplete information do not have dominant strategy equilibria. Therefore, we will also need to generalize Nash equilibrium to this setting. Recall that equilibrium is a property of a strategy profile. It is in equilibrium if each agent does not want to change her strategy given the other agents' strategies. Meaning, for an agent i, we want to the fix other agent strategies and let i optimize her strategy (meaning: calculate her best response for all possible types t_i she may have). This is an ill specified optimization as just knowing the other agents' strategies is not enough to calculate a best response. Additionally, i's best response depends additionally on i's beliefs on the types of the other agents. The standard economic treatment addresses this by assuming a common prior.

Definition 2.4. *Under the* common prior assumption*, the agent types* $\mathbf{t}$ *are drawn at random from a* prior distribution $\mathbf{F}$ *(a joint probability distribution over type profiles) and this prior distribution is* common knowledge*.*

The distribution $\mathbf{F}$ over $\mathbf{t}$ may generally be correlated. Which means that an agent with knowledge of her own type must do *Bayesian updating* to determine the distribution over the types of the remaining bidders. We denote this conditional distribution as $\mathbf{F}_{-i}\big|_{t_i}$. Of course, when the distribution of types is independent, i.e., $\mathbf{F}$ is the *product distribution* $F_1 \times \cdots \times F_n$, then $\mathbf{F}_{-i}\big|_{t_i} = \mathbf{F}_{-i}$.

Notice that a prior $\mathbf{F}$ and strategies $\mathbf{s}$ induces a distribution over the actions of each of the agents. With such a distribution over actions, the problem each agent faces of optimizing her own action is fully specified.

Definition 2.5. *A* Bayes-Nash equilibrium (BNE) *for a game G and common prior* $\mathbf{F}$ *is a strategy profile* $\mathbf{s}$ *such that for all i and t_i, $s_i(t_i)$ is a best response when other agents play* $\mathbf{s}_{-i}(\mathbf{t}_{-i})$ *when* $\mathbf{t}_{-i} \sim \mathbf{F}_{-i}\big|_{t_i}$.

To illustrate BNE, consider using the first-price auction to sell a single item to one of two agents, each with valuation drawn independently and identically from $U[0,1]$, i.e., $\mathbf{F} = F \times F$ with $F(z) = \mathbf{Pr}_{v \sim F}[v < z] = z$. Here each agent's type is her valuation. We will calculate the BNE of this game by the "guess and verify" technique. First, we guess that there is a symmetric BNE with $s_i(z) = z/2$ for $i \in \{1, 2\}$. Second, we calculate agent 1's expected utility with value v_1 and bid b_1 under the standard assumption that the agent's utility u_i is her value less her payment (when she wins).

$$\mathbf{E}[u_1] = (v_1 - b_1) \times \mathbf{Pr}[1 \text{ wins}] .$$

Calculate $\mathbf{Pr}[1 \text{ wins}] = \mathbf{Pr}[b_2 \leq b_1] = \mathbf{Pr}[v_2/2 \leq b_1] = \mathbf{Pr}[v_2 \leq 2b_1] = \mathbf{Pr}[F(2b_1)] = 2b_1$, so,

$$\begin{aligned} \mathbf{E}[u_1] &= (v_1 - b_1) \times 2b_1 \\ &= 2v_1 b_1 - 2b_1^2. \end{aligned}$$

Third, we optimize agent 1's bid. Agent 1 with value v_1 should maximize this quantity as a function of b_1, and to do so, can differentiate the function and set its derivative equal to zero. The result is $\frac{d}{db_1}(2v_1 b_1 - 2b_1^2) = 2v_1 - 4b_1 = 0$ and we can conclude that the optimal bid is $b_1 = v_1/2$. This proves that agent 1 should bid as prescribed if agent 2 does; and vice versa. Thus, we conclude that the guessed strategy profile is in BNE.

In Bayesian games it is useful to distinguish between stages of the game in terms of the knowledge sets of the agents. The three stages of a Bayesian game are *ex ante*, *interim*, and *ex post*. The ex ante stage is before values are drawn from the distribution. Ex ante, the agents know this distribution but not their own types. The interim stage is immediately after the agents learn their types, but before playing in the game. In the interim, an agent knows her own type and assumes the other agent types are drawn from the prior distribution conditioned on her own type, i.e., via *Bayesian updating*. In the ex post stage, the game is played and the actions of all agents are known.

2.4 Single-dimensional Games

We will focus on a conceptually simple class of single-dimensional games that is relevant to the auction problems we have already discussed. In a single-dimensional game, each agent's private type is her value for receiving an abstract service, i.e., $t_i = v_i$. A game has an outcome $\mathbf{x} = (x_1, \ldots, x_n)$ and payments $\mathbf{p} = (p_1, \ldots, p_n)$ where x_i is an indicator for whether agent i indeed received their desired service, i.e., $x_i = 1$ if i is served and 0 otherwise. Price p_i will denote the payment i makes to the mechanism. An agent's value can be positive or negative and an agent's payment can be positive or negative. An agent's utility is linear in her value and payment and specified by $u_i = v_i x_i - p_i$. Agents are risk-neutral expected utility maximizers.

A game G maps actions $\mathbf{b}$ of agents to an outcome and payment. Formally we will specify these outcomes and payments as:

- $x_i^G(\mathbf{b})$ = outcome to i when actions are $\mathbf{b}$, and
- $p_i^G(\mathbf{b})$ = payment from i when actions are $\mathbf{b}$.

Given a game G and a strategy profile $\mathbf{s}$ we can express the outcome and payments of the game as a function of the valuation profile. From the point of view of analysis this description of the the game outcome is much more relevant. Define

- $x_i(\mathbf{v}) = x_i^G(\mathbf{s}(\mathbf{v}))$, and
- $p_i(\mathbf{v}) = p_i^G(\mathbf{s}(\mathbf{v}))$.

We refer to the former as the *allocation rule* and the latter as the *payment rule* for G and $\mathbf{s}$ (implicit). Consider an agent i's interim perspective. She knows her own value v_i and believes the other agents values to be drawn from the distribution $\mathbf{F}$ (conditioned on her value). For G, $\mathbf{s}$, and $\mathbf{F}$ taken implicitly we can specify agent i's interim allocation and payment rules as functions of v_i.

- $x_i(v_i) = \mathbf{Pr}[x_i(v_i) = 1 \mid v_i] = \mathbf{E}[x_i(\mathbf{v}) \mid v_i]$, and
- $p_i(v_i) = \mathbf{E}[p_i(\mathbf{v}) \mid v_i]$.

With linearity of expectation we can combine these with the agent's utility function to write

- $u_i(v_i) = v_i x_i(v_i) - p_i(v_i)$.

Finally, we say that a strategy $s_i(\cdot)$ is *onto* if every action b_i agent i could play in the game is prescribed by s_i for some value v_i, i.e., $\forall b_i \exists v_i \, s_i(v_i) = b_i$. We say that a strategy profile is *onto* if the strategy of every agent is onto. For instance, the truthtelling strategy in the second-price auction is onto. When the strategies of the agents are onto, the interim allocation and payment rules defined above completely specify whether the strategies are in equilibrium or not. In particular, BNE requires that each agent (weakly) prefers playing the action corresponding (via their strategy) to her value than the action corresponding to any other value.

Fact 2.6. *For single-dimensional game G and common prior $\mathbf{F}$, an onto strategy profile $\mathbf{s}$ is in BNE if and only if for all i, v_i, and z,*

$$v_i x_i(v_i) - p_i(v_i) \geq v_i x_i(z) - p_i(z),$$

where G, $\mathbf{F}$, and $\mathbf{s}$ are implicit in the definition of $x_i(\cdot)$ and $p_i(\cdot)$.

It is easy to see that the restriction to onto strategies is only required for the "if" direction of Fact 2.6; the "only if" direction holds for all strategy profiles.

2.5 Characterization of Bayes-Nash equilibrium

We now discuss what Bayes-Nash equilibria look like. For instance, when given G, $\mathbf{s}$, and $\mathbf{F}$ we can calculate the interim allocation and payment rules $x_i(v_i)$ and $p_i(v_i)$ of each agent. We want to succinctly describe properties of these allocation and payment rules that can arise as BNE.

Theorem 2.7. *When values are drawn from a continuous joint distribution $\mathbf{F}$; G, $\mathbf{s}$, and $\mathbf{F}$ are in BNE only if for all i,*

1. *(monotonicity) $x_i(v_i)$ is monotone non-decreasing, and*
2. *(payment identity) $p_i(v_i) = v_i x_i(v_i) - \int_0^{v_i} x_i(z)\, dz + p_i(0)$,*

where often $p_i(0) = 0$. If the strategy profile is onto then the converse also holds.

Proof. We will prove the theorem in the special case where the support of each agent i's distribution is $[0, \infty]$. Focusing on a single agent i, who we will refer to as Alice, we drop subscripts i from all notations.

We break this proof into three pieces. First, we show, by picture, that the game is in BNE if the characterization holds and the strategy profile is onto. Next, we will prove that a game is in BNE only if the monotonicity condition holds. Finally, we will prove that a game is in BNE only if the payment identity holds.

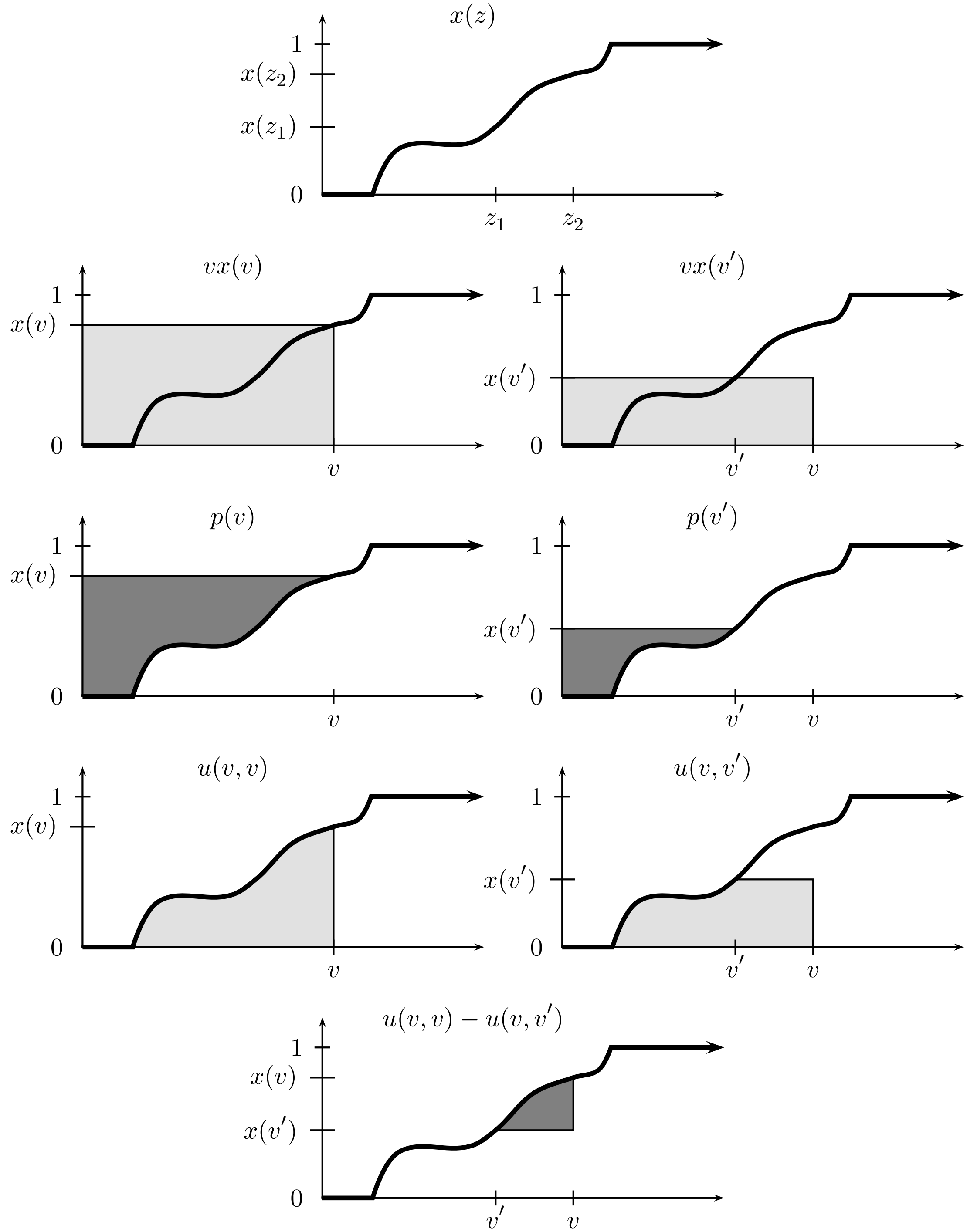


Figure 2.1: The left column shows (shaded) the surplus, payment, and utility of Alice playing action $s(v = z_2)$. The right column shows (shaded) the same for Alice playing action $s(v' = z_1)$. The final diagram shows (shaded) the difference between Alice's utility for these strategies. Monotonicity implies this difference is non-negative.

Note that if Alice with value v deviates from the equilibrium and takes action $s(v')$ instead of $s(v)$ then she will receive outcome and payment $x(v')$ and $p(v')$. This motivates the definition,

$$u(v, v') = vx(v') - p(v'),$$

which corresponds to Alice utility when she makes this deviation. For Alice's strategy to be in equilibrium it must be that for all v, and v', $u(v, v) \geq u(v, v')$, i.e., Alice derives no increased utility by deviating. The strategy profile $\mathbf{s}$ is in equilibrium if and only if the same condition holds for all agents. (The "if" direction here follows from the assumption that strategies map values onto actions. Meaning: for any action in the game there exists a value v' such that $s(v')$ is that action.)

1. G, $\mathbf{s}$, and $\mathbf{F}$ are in BNE if $\mathbf{s}$ is onto and monotonicity and the payment identity hold.

   We prove this by picture. Though the formulaic proof is simple, the pictures provide useful intuition. We consider two possible values z_1 and z_2 with $z_1 < z_2$. Supposing Alice has the high value, $v = z_2$, we argue that Alice does not benefit by simulating her strategy for the lower value, $v' = z_1$, i.e., by playing $s(v')$ to obtain outcome $x(v')$ and payment $p(v')$. We leave the proof of the opposite, that when $v = z_1$ and Alice is considering simulating the higher strategy $v' = z_2$, as an exercise for the reader.

   To start with this proof, we assume that $x(v)$ is monotone and that $p(v) = vx(v) - \int_0^v x(z)\,dz$.

   Consider the diagrams in Figure 2.1. The first diagram (top, center) shows $x(\cdot)$ which is indeed monotone as per our assumption. The column on the left show Alice's surplus, $vx(v)$; payment, $p(v)$, and utility, $u(v) = vx(v) - p(v)$, assuming that she follow the BNE strategy $s(v = z_2)$. The column on the right shows the analogous quantities when Alice follows strategy $s(v' = z_1)$ but has value $v = z_2$. The final diagram (bottom, center) shows the difference in the Alice's utility for the outcome and payments of these two strategies. Note that as the picture shows, the monotonicity of the allocation function implies that this difference is always non-negative. Therefore, there is no incentive for Alice to simulate the strategy of a lower value.

   As mentioned, a similar proof shows that Alice has no incentive to simulate her strategy for a higher value. We conclude that she, with value v, (weakly) prefers to play the BNE strategy $s(v)$.

2. G, $\mathbf{s}$, and $\mathbf{F}$ are in BNE only if the allocation rule is monotone.

   If we are in BNE then for all valuations, v and v', $u(v, v) \geq u(v, v')$. Expanding we require

   $$vx(v) - p(v) \geq vx(v') - p(v').$$

   We now consider z_1 and z_2 and take turns setting $v = z_1$, $v' = z_2$, and $v' = z_1$, $v = z_2$. This yields the following two inequalities:

   $$v = z_2, v' = z_1 \Longrightarrow z_2 x(z_2) - p(z_2) \geq z_2 x(z_1) - p(z_1), \text{ and} \tag{2.1}$$
   $$v = z_1, v' = z_2 \Longrightarrow z_1 x(z_1) - p(z_1) \geq z_1 x(z_2) - p(z_2). \tag{2.2}$$

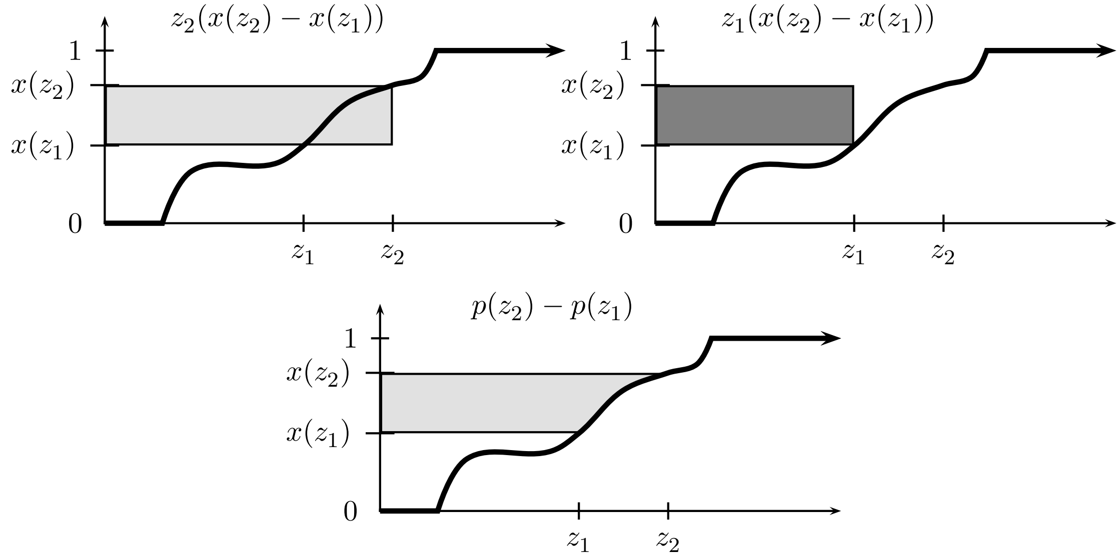


Figure 2.2: Upper (top, left) and lower bounds (top, right) for the difference in payments for two strategies z_1 and z_2 imply that the difference in payments (bottom) must satisfy the payment identity.

Adding these inequalities and canceling the payment terms we have,

$$z_2 x(z_2) + z_1 x(z_1) \geq z_2 x(z_1) + z_1 x(z_2).$$

Rearranging,

$$(z_2 - z_1)(x(z_2) - x(z_1)) \geq 0.$$

For $z_2 - z_1 > 0$ it must be that $x(z_2) - x(z_1) \geq 0$, i.e., $x(\cdot)$ is monotone non-decreasing.

3. G, $\mathbf{s}$, and $\mathbf{F}$ are in BNE only if the payment rule satisfies the payment identity.

   We will give two proofs that payment rule must satisfy $p(v) = vx(v) - \int_0^v x(z)\,dz$; the first is a calculus-based proof under the assumption that and each of $x(\cdot)$ and $p(\cdot)$ are differentiable and the second is a picture-based proof that requires no assumption.

   Calculus-based proof: Fix v and recall that $u(v,z) = vx(z) - p(z)$. Let $u'(v,z)$ be the partial derivative of $u(v,z)$ with respect to z. Thus, $u'(v,z) = vx'(z) - p'(z)$, where $x'(\cdot)$ and $p'(\cdot)$ are the derivatives of $p(\cdot)$ and $x(\cdot)$, respectively. Since truthfulness implies that $u(v,z)$ is maximized at $z = v$. It must be that

$$u'(v,v) = vx'(v) - p'(v) = 0.$$

   This formula must hold true for all values of v. For remainder of the proof, we treat this identity formulaically. To emphasize this, substitute $z = v$:

$$zx'(z) - p'(z) = 0.$$

Solving for $p'(z)$ and then integrating both sides of the equality from 0 to v we have,

$$\begin{aligned} p'(z) &= zx'(z), \text{ so} \\ \int_0^v p'(z)dz &= \int_0^v zx'(z)\,dz, \text{ so} \\ p(v) - p(0) &= \Big[zx(z)\Big]_0^v - \int_0^v x(z)\,dz \\ &= vx(v) - \int_0^v x(z)\,dz. \end{aligned}$$

Adding $p(0)$ from both sides of the equality, we conclude that the payment identity must hold.

Picture-based proof: Consider equations (2.1) and (2.2) and solve in each for $p(z_2) - p(z_1)$ in each:

$$z_2(x(z_2) - x(z_1)) \geq p(z_2) - p(z_1) \geq z_1(x(z_2) - x(z_1)).$$

The first inequality gives an upper bound on the difference in payments for two types z_2 and z_1 and the second inequality gives a lower bound. It is easy to see that the only payment rule that satisfies these upper and lower bounds for all pairs of types z_2 and z_1 has payment difference exactly equal to the area to the left of the allocation rule between $x(z_1)$ and $x(z_2)$. See Figure 2.2. The payment identity follows by taking $z_1 = 0$ and $z_2 = v$. □

As we conclude the proof of the BNE characterization theorem, it is important to note how little we have assumed of the underlying game. We did not assume it was a single-round, sealed-bid auction. We did not assume that only a winner will make payments. Therefore, we conclude for any potentially wacky, multi-round game the outcomes of all Bayes-Nash equilibria have a nice form.

2.6 Characterization of Dominant Strategy Equilibrium

Dominant strategy equilibrium is a stronger equilibrium concept than Bayes-Nash equilibrium. All dominant strategy equilibria are Bayes-Nash equilibria, but as we have seen, the opposite is not true; for instance, there is no DSE in the first-price auction. Recall that a strategy profile is in DSE if each agent's strategy is optimal for them regardless of what other agents are doing. The DSE characterization theorem below follows from the BNE characterization theorem.

Theorem 2.8. *G and $\mathbf{s}$ are in DSE only if for all i,*

1. *(monotonicity) $x_i(\mathbf{v}_{-i}, v_i)$ is monotone non-decreasing in v_i, and*
2. *(payment identity) $p_i(\mathbf{v}_{-i}, v_i) = v_i x_i(\mathbf{v}_{-i}, v_i) - \int_0^{v_i} x_i(\mathbf{v}_{-i}, z)\,dz + p_i(\mathbf{v}_{-i}, 0)$,*

where $(\mathbf{v}_{-i}, z)$ denotes the valuation profile with the ith coordinate replaced with z. If the strategy profile is onto then the converse also holds.

It was important when discussing BNE to explicitly refer to $x_i(v_i)$ and $p_i(v_i)$ as the probability of allocation and the expected payments because a game played by agents with values drawn from a distribution will inherently, from agent i's perspective, have a randomized outcome and payment. In contrast, for games with DSE we can consider outcomes and payments in a non-probabilistic sense. A deterministic game, i.e., one with no internal randomization, will result in deterministic outcomes and payments. For our single-dimensional game where an agent is either served or not served we will have $x_i(\mathbf{v}) \in \{0, 1\}$. This specification along with the monotonicity condition implied by DSE implies that the function $x_i(\mathbf{v}_{-i}, v_i)$ is a step function in v_i. The reader can easily verify that the payment required for such a step function is exactly the critical value, i.e., τ_i at which $x_i(\mathbf{v}_{-i}, \cdot)$ changes from 0 to 1. This gives the following corollary.

Corollary 2.9. *A deterministic game G and strategies $\mathbf{s}$ are in DSE only if for all i,*

1. *(step-function) $x_i(\mathbf{v}_{-i}, v_i)$ steps from 0 to 1 at $\tau_i = \inf\{z : x_i(\mathbf{v}_{-i}, z) = 1\}$, and*
2. *(critical value) for* $p_i(\mathbf{v}_{-i}, v_i) = \begin{cases} \tau_i & \text{if } x_i(\mathbf{v}_{-i}, v_i) = 1 \\ 0 & \text{otherwise} \end{cases} + p_i(\mathbf{v}_{-i}, 0).$

If the strategy profile is onto then the converse also holds.

Notice that the above theorem deliberately skirts around a subtle tie-breaking issue. Consider the truthtelling DSE of the second-price auction on two agents. What happens when $v_1 = v_2$? One agent should win and pay the other's value, but, as this results in a utility of zero, from the perspective of utility maximization both agents are indifferent as to which of them it is. One natural tie-breaking rule is the lexicographical one, i.e., in favor of agent 1 winning. For this rule, agent 1 wins when $v_1 \in [v_2, \infty)$ and agent 2 wins when $v_2 \in (v_1, \infty)$. The critical values are $t_1 = v_2$ and $t_2 = v_1$. We will usually prefer the randomized tie-breaking rule because of its symmetry.

2.7 Revenue Equivalence

We are now ready to make one of the most significant observations in auction theory. Namely, mechanisms with the same outcome in BNE have the same expected revenue. In fact, not only do they have the same expected revenue, but each agent has the same expected payment in each mechanism. We state this result as a corollary of Theorem 2.7 which is intuitively clear. The payment identity means that the payment rule is precisely determined by the allocation rule and the payment of the lowest type, i.e., $p_i(0)$.

Corollary 2.10. *Consider any two mechanisms where 0-valued agents pay nothing, if the mechanisms have the same BNE outcome then they have same expected revenue.*

We can now quantitatively compare the second-price and first-price auctions from a revenue standpoint. Consider the case where the agent's values are distributed independently and identically. What is the equilibrium outcome of the second-price auction? The agent with the highest valuation wins. What is the equilibrium outcome of the first-price auction? This question requires a little more thought. Since the distributions are identical, it is reasonable to expect that there is a symmetric equilibrium, i.e., one where $s_i = s_{i'}$ for all i and i'. Furthermore, it is reasonable to expect that the strategies are monotone, i.e., an agent with a higher value will out bid an agent with a lower value. Under these assumptions, the agent with the highest value wins. Of course, in both auctions a 0-valued agent will pay nothing. Therefore, we can conclude that the two auctions obtain the same expected revenue.

As an example of revenue equivalence consider first-price and second-price auctions for selling a single item to two agents with values drawn from $U[0, 1]$. The expected revenue of the second-price auction is $\mathbf{E}\left[v_{(2)}\right]$. In Section 2.3 we saw that the symmetric strategy of the first-price auction in this environment is for each agent to bid half their value. The expected revenue of first-price auction is therefore $\mathbf{E}\left[v_{(1)}/2\right]$. An important fact about uniform random variables is that in expectation they evenly divide the interval they are over, i.e., $\mathbf{E}\left[v_{(1)}\right] = 2/3$ and $\mathbf{E}\left[v_{(2)}\right] = 1/3$. How do the revenues of these two auctions compare? Their revenues are identically 1/3.

Corollary 2.11. *When agent's values are independent and identically distributed, the second-price and first-price auction have the same expected revenue.*

Of course, much more bizarre auctions are governed by revenue equivalence. As an exercise the reader is encourage to verify that the *all-pay auction*; where agents submit bids, the highest bidder wins, and all agents pay their bid; is revenue equivalent to the first- and second-price auctions.

2.8 Solving for Bayes-Nash Equilibrium

While it is quite important to know what outcomes are possible in BNE, it is also often important to be able to solve for the BNE strategies. For instance, suppose you were a bidder bidding in an auction. How would you bid? In this section we describe an elegant technique for calculating BNE strategies in symmetric environments using revenue equivalence. Actually, we use something a little stronger than revenue equivalence: *interim payment equivalence*. This is the fact that if two mechanisms have the same allocation rule, they must have the same payment rule (because the payment rules satisfy the payment identity). As described previously, the interim payment of agent i with value v_i is $p_i(v_i)$.

Suppose we are to solve for the BNE strategies of mechanism M. The approach is to express an agent's payment in M as a function of the agent's action, then to calculate the agent's expected payment in a strategically-simple mechanism M' that is revenue equivalent to M (usually a "second-price implementation" of M). Setting these terms equal and solving for the agents action gives the equilibrium strategy.

We give the high level the procedure below. As a running example we will calculate the equilibrium strategies in the first-price auction with two $U[0,1]$ agents, in doing so we will use a calculation of expected payments in the strategically-simple second-price auction in the same environment.

1. *Guess* what the outcome might be in Bayes-Nash equilibrium.

   E.g., for the first-price auction with two agents with values $U[0,1]$, in BNE, we expect the agent with the highest value to win. Thus, guess that the highest-valued agent always wins.

2. *Calculate* the expected payment of an agent with fixed value in a strategically-simple auction with the same equilibrium outcome.

   E.g., recall that it is a dominant strategy equilibrium (a special case of Bayes-Nash equilibrium) in the second-price auction for each agent to bid their value. I.e., $b_1 = v_1$ and $b_2 = v_2$. Thus, the second-price auction also selects the agent with the highest value to win. So, let us calculate the expected payment of player 1 when their value is v_1.

   $$\mathbf{E}[p_1(v_1)] = \mathbf{E}[p_1(v_1) \mid 1 \text{ wins}]\, \mathbf{Pr}[1 \text{ wins}] + \mathbf{E}[p_1(v_1) \mid 1 \text{ loses}]\, \mathbf{Pr}[1 \text{ loses}]\,.$$

   Calculate each of these components for the second-price auction:

   $$\begin{aligned}\mathbf{E}[p_1(v_1) \mid 1 \text{ wins}] &= \mathbf{E}[v_2 \mid v_2 < v_1] \\ &= v_1/2.\end{aligned}$$

   The first step follows by the definition of the second-price auction and its dominant strategy equilibrium (i.e., $b_2 = v_2$). The second step follows because in expectation a uniform random variable evenly divides the interval it is over, and once we condition on $v_2 < v_1$, v_2 is $U[0, v_1]$.

   $$\begin{aligned}\mathbf{Pr}[1 \text{ wins}] &= \mathbf{Pr}[v_2 < v_1] \\ &= v_1.\end{aligned}$$

   The first step follows from the definition of the second-price auction and its dominant strategy equilibrium. The second step is because v_1 is uniform on $[0,1]$.

   $$\mathbf{E}[p_1(v_1) \mid 1 \text{ loses}] = 0.$$

   This is because a loser pays nothing in the second-price auction. This means that we do not need to calculate $\mathbf{Pr}[1 \text{ loses}]$. Plug these into the equation above to obtain:

   $$\mathbf{E}[p_1(v_1)] = v_1^2/2.$$

3. *Solve* for bidding strategies from expected payments.

   E.g., by revenue equivalence the expected payment of player 1 with value v_1 is $v_1^2/2$ in both the first-price and second-price auction. We can recalculate this expected payment in the first-price auction using the bid of the player as a variable and then solve for what that bid must be.

   Again,

   $$\mathbf{E}[p_1(v_1)] = \mathbf{E}[p_1(v_1) \mid 1 \text{ wins}] \, \mathbf{Pr}[1 \text{ wins}] + \mathbf{E}[p_1(v_1) \mid 1 \text{ loses}] \, \mathbf{Pr}[1 \text{ loses}] \, .$$

   Calculate each of these components for the first-price auction where agent 1 follows strategy $s_1(v_1)$:

   $$\mathbf{E}[p_i(v_i) \mid 1 \text{ wins}] = s_1(v_1).$$

   This by the definition of the first-price auction: if you win you pay your bid.

   $$\mathbf{Pr}[1 \text{ wins}] = \mathbf{Pr}[v_2 < v_1] = v_1.$$

   This is the same as above.

   $$\mathbf{E}[p_1(v_1) \mid 1 \text{ loses}] = 0.$$

   This is because a loser pays nothing in the first-price auction. This means that we do not need to calculate $\mathbf{Pr}[1 \text{ loses}]$. Plug these into the equation above to obtain:

   $$\mathbf{E}[p_1(v_1)] = s_1(v_1) \cdot v_1.$$

   This must equal the expected payment calculated in the previous step for the second-price auction, implying:

   $$v_1^2/2 = s_1(v_1) \cdot v_1.$$

   We can solve for $s_1(v_1)$ and get

   $$s_1(v_1) = v_1/2.$$

4. Verify initial guess was correct.

   Indeed, if agents follow symmetric strategies $s_1(z) = s_2(z) = z/2$ then the agent with the highest value wins.

In the above first-price auction example it should be readily apparent that we did slightly more work than we had to. In this case it would have been enough to note that in both the first- and second-price auction a loser pays nothing. We could therefore simply equate the expected payments conditioned on winning:

$$\mathbf{E}[p_1(v_1) \mid 1 \text{ wins}] = \underbrace{v_1/2}_{\text{second-price}} = \underbrace{s_1(v_i)}_{\text{first-price}}.$$

We can also work through the above framework for the *all-pay* auction where the agents submit bids, the highest bid wins, but all agents pay their bid. The all-pay auction is also is revenue equivalent to the second-price auction. However, now we compare the total expected payment (regardless of winning) to conclude:

$$\mathbf{E}[p_1(v_1)] = \underbrace{v_1^2/2}_{\text{second-price}} = \underbrace{s_1(v_i)}_{\text{all-pay}}.$$

I.e., the BNE strategies for the all-pay auction are $s_i(z) = z^2/2$. Remember, of course, that the equilibrium strategies solved for above are for single-item auctions and two agents with values uniform on $[0, 1]$. For different distributions or numbers of agents the equilibrium strategies will generally be difference.

2.9 The Revelation Principle

We are interested in designing mechanisms and, while the characterization of Bayes-Nash equilibrium is elegant, solving for equilibrium is still generally quite challenging. The final piece of the puzzle, and the one that has enabled much of modern mechanism design is the *revelation principle*. The revelation principle states, informally, that if we are searching among mechanisms for one with a desirable equilibrium we may restrict our search to single-round, sealed-bid mechanisms in which truthtelling is an equilibrium.

Definition 2.12. *A* direct revelation *mechanism is single-round, sealed bid, and has action space equal to the type space, (i.e., an agent can bid any value they might have)*

Definition 2.13. *A direct revelation mechanism is* Bayesian incentive compatible *(BIC) if truthtelling is a Bayes-Nash equilibrium.*

Definition 2.14. *A direct revelation mechanism is* dominant strategy incentive compatible *(DSIC) if truthtelling is a dominant strategy equilibrium.*

Theorem 2.15. *Any mechanism M with good BNE (resp. DSE) can be converted into a BIC (resp. DSIC) mechanism M' with the same BNE (resp. DSE) outcome.*

Proof. We will prove the BNE variant of the theorem. Let $\mathbf{s}$, $\mathbf{F}$, and M be in BNE. Define single-round, sealed-bid mechanism M' as follows:

1. Accept sealed bids $\mathbf{b}$.

2. Simulate $\mathbf{s}(\mathbf{b})$ in M.

3. Output the outcome of the simulation.

We now claim that $\mathbf{s}$ being a BNE of M implies truthtelling is a BNE of M' (for distribution $\mathbf{F}$). Let $\mathbf{s}'$ denote the truthtelling strategy. In M', consider agent i and suppose all other agents are truthtelling. This means that the actions of the other players in M are distributed as $\mathbf{s}_{-i}(\mathbf{s}'_{-i}(\mathbf{v}_{-i})) = \mathbf{s}_{-i}(\mathbf{v}_{-i})$ for $\mathbf{v}_{-i} \sim \mathbf{F}_{-i}\big|_{v_i}$. Of course, in M if other players are playing $\mathbf{s}_{-i}(\mathbf{v}_{-i})$ then since $\mathbf{s}$ is a BNE, i's best response is to play $s_i(v_i)$ as well. Agent i can play this action in the simulation of M is by playing the truthtelling strategy $s'_i(v_i) = v_i$ in M'. $\square$

Notice that we already, in Chapter 1, saw the revelation principle in action. The second-price auction is the revelation principle applied to the English auction.

Because of the revelation principle, for many of the mechanism design problems we consider, we will look first for Bayesian or dominant-strategy incentive compatible mechanisms. The revelation principle guarantees that, in our search for optimal BNE mechanisms, it suffices to search only those that are BIC (and likewise for DSE and DSIC). The following are corollaries of our BNE and DSE characterization theorems.

We defined the allocation and payment rules $\mathbf{x}(\cdot)$ and $\mathbf{p}(\cdot)$ as functions of the valuation profile for an implicit game G and strategy profile $\mathbf{s}$. When the strategy profile is truthtelling, the allocation and payment rules are identical the original mappings of the game from actions to allocations and prices, denoted $\mathbf{x}^G(\cdot)$ and $\mathbf{p}^G(\cdot)$. Additionally, let $x_i^G(v_i) = \mathbf{E}\Big[x_i^G(\mathbf{v}) \mid v_i\Big]$ and $p_i^G(v_i) = \mathbf{E}\Big[p_i^G(\mathbf{v}) \mid v_i\Big]$ for $\mathbf{v} \sim \mathbf{F}$. Furthermore, the truthtelling strategy profile in a direct-revelation game is onto.

Corollary 2.16. *A direct mechanism M is BIC for distribution $\mathbf{F}$ if and only if for all i,*

1. *(monotonicity) $x_i^M(v_i)$ is monotone non-decreasing, and*
2. *(payment identity) $p_i^M(v_i) = v_i x_i^M(v_i) - \int_0^{v_i} x_i^M(z)\,dz + p_i^M(0)$.*

Corollary 2.17. *A direct mechanism M is DSIC if and only if for all i,*

1. *(monotonicity) $x_i^M(\mathbf{v}_{-i}, v_i)$ is monotone non-decreasing in v_i, and*
2. *(payment identity) $p_i^M(\mathbf{v}_{-i}, v_i) = v_i x_i^M(\mathbf{v}_{-i}, v_i) - \int_0^{v_i} x_i^M(\mathbf{v}_{-i}, z)\,dz + p_i^M(\mathbf{v}_{-i}, 0)$.*

Corollary 2.18. *A direct, deterministic mechanism M is DSIC if and only if for all i,*

1. *(step-function) $x_i^M(\mathbf{v}_{-i}, v_i)$ steps from 0 to 1 at $\tau_i = \inf\{z : x_i^M(\mathbf{v}_{-i}, z) = 1\}$, and*
2. *(critical value) for $p_i^M(\mathbf{v}_{-i}, v_i) = \begin{cases} \tau_i & \text{if } x_i^M(\mathbf{v}_{-i}, v_i) = 1 \\ 0 & \text{otherwise} \end{cases} + p_i^M(\mathbf{v}_{-i}, 0)$.*

When we construct mechanisms we will use the "if" directions of these theorems. When discussing incentive compatible mechanisms we will assume that agents follow their equilibrium strategies and, therefore, each agent's bid is equal to her valuation.

Between DSIC and BIC clearly DSIC is a stronger incentive constraint and we should prefer it over BIC if possible. Importantly, DSIC requires fewer assumptions on the agents. For a DSIC mechanisms, each agent must only know her own value; while for a BIC mechanism, each agent must also know the distribution over other agent values. Unfortunately, there will be some environments where we derive BIC mechanisms where no analogous DSIC mechanism is known.

The revelation principle fails to hold in some environments of interest. We will take special care to point these out. Two such environments, for instance, are where agents only learn their values over time, or where the designer does not know the prior distribution (and hence cannot simulate the agent strategies).

Exercises

2.1 Find a symmetric mixed strategy equilibrium in the chicken game described in Section 2.1. I.e., find a probability ρ such that if James Dean stays with probability ρ and swerves with probability $1 - \rho$ then Buzz is happy to do the same.

2.2 In Section 2.3 we characterized outcomes and payments for BNE in single-dimensional games. This characterization explains what happens when agents behave strategically.

Suppose instead of strategic interaction, we care about fairness. Consider a valuation profile, $\mathbf{v} = (v_1, \ldots, v_n)$, an allocation vector, $\mathbf{x} = (x_1, \ldots, x_n)$, and payments, $\mathbf{p} = (p_1, \ldots, p_n)$. Here x_i is the probability that i is served and p_i is the expected payment of i regardless of whether i is served or not.

Allocation $\mathbf{x}$ and payments $\mathbf{p}$ are *envy-free* for valuation profile $\mathbf{v}$ if no agent wants to unilaterally swap allocation and payment with another agent. I.e., for all i and j,

$$v_i x_i - p_i \geq v_i x_j - p_j.$$

Characterize envy-free allocations and payments (and prove your characterization correct). Unlike the BNE characterization, your characterization of payments will not be unique. Instead, characterize the minimum payments that are envy-free. Draw a diagram illustrating your payment characterization. (Hint: You should end up with a very similar characterization to that of BNE.)

2.3 AdWords is Google product in which the search engine sells at auction advertisements that appear along side search results on the search results page. Consider the following *position auction* environment which provides a simplified model of AdWords. There are m advertisement slots that appear along side search results and n advertisers. Advertiser i has value v_i for a click. Slot j has *click-through rate* w_j, meaning, if an advertiser is assigned slot j the advertiser will receive a click with probability w_j. Each advertiser can be assigned at most one slot and each slot can be assigned at most one advertiser. If a slot is left empty, all subsequent slots must be left empty, i.e., slots cannot be skipped. Assume that the slots are ordered from highest click-through rate to lowest, i.e., $w_j \geq w_{j+1}$ for all j.

(a) Find the envy-free (See Exercise 2.2) outcome and payments with the maximum social surplus. Give a description and formula for the envy-free outcome and payments for each advertiser. (Feel free to specify your payment formula with a comprehensive picture.)

(b) In the real AdWords problem, advertisers only pay if they receive a click, whereas the payments calculated, i.e., $\mathbf{p}$, are in expected over all outcomes, click or no click. If we are going to charge advertisers only if they are clicked on, give a formula for calculating these payments $\mathbf{p}'$ from $\mathbf{p}$.

(c) The real AdWords problem is solved by auction. Design an auction that maximizes the social surplus in dominant strategy equilibrium. Give a formula for the payment rule of your auction (again, a comprehensive picture is fine). Compare your DSE payment rule to the envy-free payment rule. Draw some informal conclusions.

2.4 Consider the first-price auction for selling k units of an item to n unit-demand agents. This auction solicits bids and allocates one units to each of the k highest-bidding agents. These winners are charged their bids. This auction is revenue equivalent to the k-unit "second-price" auction where the winners are charged the $k+1$st highest bid, $b_{(k+1)}$. Solve for the symmetric Bayes-Nash equilibrium strategies in the first-price auction when the agent values are i.i.d. $U[0,1]$.

2.5 Consider the position auction environment with $n = m = 2$ (See Exercise 2.3). Consider running the following first-price auction: The advertisers submit bids $\mathbf{b} = (b_1, b_2)$. The advertisers are assigned to slots in order of their bids. Advertisers pay their bid when clicked. Use revenue equivalence to solve for BNE strategies $\mathbf{s}$ when the values of the advertisers are drawn independent and identically from $U[0,1]$.

Chapter Notes

The characterization of Bayes-Nash equilibrium, revenue equivalence, and the revelation principle come from Myerson (1981). The BNE characterization proof presented here comes from Archer and Tardos (2001).

Chapter 3

Optimal Mechanisms

In this chapter we discuss the objectives of social surplus and profit. As we will see, the economics of designing mechanisms to maximize social surplus is relatively simple. The optimal mechanism is a simple generalization of the second-price auction we have already discussed. Furthermore, it is dominant strategy incentive compatible and prior-free, i.e., it is not dependent on distributional assumptions. Social surplus maximization is unique among economic objectives in this regard.

The objective of profit maximization, on the other hand, adds significant new challenge: for profit there is no single optimal mechanism. For any mechanism, there is a distributional setting and another mechanism where this new mechanism has strictly larger profit than the first one.

This non-existence of an absolutely optimal mechanism requires a relaxation of what we consider a good mechanism. To address this challenge, this chapter follows the traditional economics approach of Bayesian optimization. We will assume that the distribution of the agents' preferences is common knowledge, even to the mechanism designer. This designer should then search for the mechanism that maximizes her expected profit when preferences are indeed drawn from the distribution.

As an example we could consider two agents with values drawn independently and identically from $U[0,1]$. The second-price auction obtains revenue equal to the expected second-highest value, $\mathbf{E}\big[v_{(2)}\big] = 1/3$. A natural question is whether more revenue can be had. As a first step, it is similarly easy to calculate that the second-price auction with reserve $1/2$ obtains an expected revenue of $5/12$ (which is higher than $1/3$).[1] Perhaps surprisingly, a seller can make more money by sometimes not selling the item even when there is a buyer willing to pay. In this chapter we show that the second-price auction with reserve $1/2$ is indeed optimal for this two agent example and furthermore we give a concise characterization

[1] The calculation proceeds as follows: There are three cases (i) $1/2 > v_{(1)} > v_{(2)}$, (ii) $v_{(1)} > 1/2 > v_{(2)}$, and (iii), $v_{(1)} > v_{(2)} > 1/2$. Case (i) happens with probability $1/4$ and has no revenue; case (ii) happens with probability $1/2$ and has revenue $1/2$; and case (iii) happens with probability $1/4$ and has expected revenue $\mathbf{E}\big[v_{(2)} \mid \text{case (iii) occurs}\big] = 2/3$. The calculation of the expected revenue in case (iii) follows from the conditional values being $U[1/2,1]$ and the fact that, in expectation, uniform random variables evenly divide the interval they are over. The total expected revenue can then be calculated as $5/12$.

of the optimal auction for any single-dimensional agent environment.

3.1 Single-dimensional Environments

In our previous discussion of Bayes-Nash equilibrium we focused on the agents' incentives. Agents are single-dimensional, i.e., each has a single private value for receiving some abstract service and quasi-linear utility, i.e., the agent's utility is her value for the service less her payment. Recall that the outcome of a single-dimensional game is an allocation $\mathbf{x} = (x_1, \ldots, x_n)$, where x_i is an indicator for whether agent i is served, and payments $\mathbf{p} = (p_1, \ldots, p_n)$, where p_i is the payment made by agent i. Here we formalize the designer's constraints and objectives.

Definition 3.1. *A* general cost environment *is one where the designer must pay a service cost $c(\mathbf{x})$ for the allocation $\mathbf{x}$ produced.*

Definition 3.2. *A* general feasibility environment *is one where there is a feasibility constraint over the set of agents that can be simultaneously served.*

Definition 3.3. *A* downward-closed *feasibility constraint is one where subsets of feasible sets are feasible.*

Of course, downward-closed environments are a special case of general feasibility environments which are a special case of general cost environments. We can express general feasibility environments as general costs environments were $c(\cdot) \in \{0, \infty\}$. We can similarly express downward-closed feasibility environments as the further restriction where $\mathbf{x}' \leq \mathbf{x}$ (i.e., for all i, $x'_i \leq x_i$) and $c(\mathbf{x}) = 0$ and implies that $c(\mathbf{x}') = 0$. We will be aiming for general mechanism design results and the most general results will be the ones that hold in the most general environments. However, we will pay special attention to restrictions on the environment that enable illuminating observations about optimal mechanisms.

The two most fundamental designer objectives are social surplus, a.k.a., social welfare,[2] and profit.

Definition 3.4. *The* social surplus *of an allocation is the cumulative value of agents served less the service cost:*

$$\text{Surplus}(\mathbf{v}, \mathbf{x}) = \sum\nolimits_i v_i \cdot x_i - c(\mathbf{x}).$$

Definition 3.5. *The* profit *of allocation and payments is the cumulative payment of agents less the service cost:*

$$\text{Profit}(\mathbf{p}, \mathbf{x}) = \sum\nolimits_i p_i - c(\mathbf{x}).$$

Implicit in the definition of social surplus is the fact that the payments from the agents are transferred to the service provider and therefore do not affect the objective.[3]

[2] A mechanism that optimizes social surplus is said to be *economically efficient*; though, we will not use this terminology because of possible confusion with *computational efficiency*.

[3] An alternative notion would be to consider only the total value derived by the agents, i.e., the surplus less the total payments. This *residual surplus* was discussed in detail in Chapter 1; mechanisms for optimizing residual surplus are the subject of Exercise 3.1.

The single-item and routing environments that were discussed in Chapter 1 are special cases of downward-closed environments. Single-item environments have

$$c(\mathbf{x}) = \begin{cases} 0 & \text{if } \sum_i x_i \leq 1, \text{ and} \\ \infty & \text{otherwise.} \end{cases}$$

In routing environments, recall, each agent has a message to send betwen a source and destination in the network.

$$c(\mathbf{x}) = \begin{cases} 0 & \text{if messages with } x_i = 1 \text{ can be simultaneously routed, and} \\ \infty & \text{otherwise.} \end{cases}$$

We have yet to see any examples of general cost environments. One natural one is that of a *multicast auction*. The story for this problem comes from live video steaming. Suppose we wish to stream live video to viewers (agents) in a computer network. Because of the high-bandwidth nature of video streaming the content provider must lease the network links. Each link has a publicly known cost. To serve a set of agents, the designer must pay the cost of network links that connect each agent, located at different nodes in the network, to the "root", i.e., the origin of the multicast. The nature of multicast is that the messages need only be transmitted once on each edge to reach the agents. Therefore, the total cost to serve these agents is the minimum cost of the *multicast tree* that connects them.[4]

3.2 Social Surplus

We now derive the optimal mechanism for social surplus. To do this we walk through a standard approach in mechanism design. We completely relax the Bayes-Nash equilibrium incentive constraints and ask and solve the remaining non-game-theoretic optimization question. We then verify that this solution does not violate the incentive constraints. We conclude that the resulting mechanism is optimal.

The non-game-theoretic optimization problem of maximizing surplus is that of finding $\mathbf{x}$ to maximize $\text{Surplus}(\mathbf{v}, \mathbf{x}) = \sum_i v_i x_i - c(\mathbf{x})$. Let OPT be an optimal algorithm for solving this problem. We will care about both the allocation that OPT selects, i.e., $\text{argmax}_{\mathbf{x}} \text{Surplus}(\mathbf{v}, \mathbf{x})$ and its surplus $\max_{\mathbf{x}} \text{Surplus}(\mathbf{v}, \mathbf{x})$. Where it is unambiguous we will use notation $\text{OPT}(\mathbf{v})$ to denote either of these quantities. Notice that the formulation of OPT has no mention of Bayes-Nash equilibrium incentive constraints.

We know from our characterization that the allocation rule of any BNE is monotone, and that any monotone allocation rule can be implemented in BNE with the appropriate payment rule. Thus, relative to the non-game-theoretic optimization, the mechanism design problem of finding a BIC mechanism to maximize surplus has an added monotonicity constraint. As it turns out, even though we did not impose a monotonicity constraint on OPT, it is satisfied anyway.

[4]In combinatorial optimization this problem is known as the *weighted Steiner tree* problem. It is a computationally challenging variant of the *minimum spanning tree* problem.

Lemma 3.6. *For each agent i and all values of other agents $\mathbf{v}_{-i}$, the allocation rule of* OPT *for agent i is a step function.*

Proof. Consider any agent i. There are two situations of interest. Either i is served by $\mathrm{OPT}(\mathbf{v})$ or i is not served by $\mathrm{OPT}(\mathbf{v})$. We write out the surplus of OPT in both of these cases. Below, notation $(\mathbf{v}_{-i}, z)$ denotes the vector $\mathbf{v}$ with the ith coordinate replaced with z.

Case 1 ($i \in \mathrm{OPT}$):

$$\begin{aligned}\mathrm{OPT}(\mathbf{v}) &= \max_{\mathbf{x}} \mathrm{Surplus}(\mathbf{v}, \mathbf{x}) \\ &= v_i + \max_{\mathbf{x}_{-i}} \mathrm{Surplus}((\mathbf{v}_{-i}, 0), (\mathbf{x}_{-i}, 1)).\end{aligned}$$

Define $\mathrm{OPT}_{-i}(\mathbf{v})$ as the second term on the right hand side. Thus,

$$\mathrm{OPT}(\mathbf{v}) = v_i + \mathrm{OPT}_{-i}(\mathbf{v}).$$

Notice that $\mathrm{OPT}_{-i}(\mathbf{v})$ is not a function of v_i.

Case 2 ($i \notin \mathrm{OPT}$):

$$\begin{aligned}\mathrm{OPT}(\mathbf{v}) &= \max_{\mathbf{x}} \mathrm{Surplus}(\mathbf{v}, \mathbf{x}) \\ &= \max_{\mathbf{x}_{-i}} \mathrm{Surplus}((\mathbf{v}_{-i}, 0), (\mathbf{x}_{-i}, 0)).\end{aligned}$$

Define $\mathrm{OPT}(\mathbf{v}_{-i})$ as the term on the right hand side. Thus,

$$\mathrm{OPT}(\mathbf{v}) = \mathrm{OPT}(\mathbf{v}_{-i}).$$

Notice that $\mathrm{OPT}(\mathbf{v}_{-i})$ is not a function of v_i.

OPT chooses whether or not to allocate to agent i, and thus which of these cases we are in, so as to optimize the surplus. Therefore, OPT allocates to i whenever the surplus from Case 1 is greater than the surplus from Case 2. I.e., when

$$v_i + \mathrm{OPT}_{-i}(\mathbf{v}) \geq \mathrm{OPT}(\mathbf{v}_{-i}).$$

Solving for v_i we conclude that OPT allocates to i whenever

$$v_i \geq \mathrm{OPT}(\mathbf{v}_{-i}) - \mathrm{OPT}_{-i}(\mathbf{v}).$$

Notice that neither of the terms on the right hand side contain v_i. Therefore, the allocation rule for i is a step function with critical value $\tau_i = \mathrm{OPT}(\mathbf{v}_{-i}) - \mathrm{OPT}_{-i}(\mathbf{v})$. □

Since the allocation rule induced by OPT is a step function, it satisfies our strongest incentive constraint: with the appropriate payments (i.e., the "critical values") truthtelling is a dominant strategy equilibrium (Corollary 2.18). The resulting surplus maximization mechanism is often referred to as the *Vickrey-Clarke-Groves* (VCG) mechanism, named after William Vickrey, Edward Clarke, and Theodore Groves.

Mechanism 3.1. *The* surplus maximization (SM) *mechanism is:*

1. *Solicit and accept sealed bids* $\mathbf{b}$.
2. $\mathbf{x} \leftarrow \mathrm{OPT}(\mathbf{b})$*, and*
3. *for each* i*,* $p_i \leftarrow \mathrm{OPT}(\mathbf{b}_{-i}) - \mathrm{OPT}_{-i}(\mathbf{b})$.

An intuitive description of $\mathrm{OPT}(\mathbf{v}_{-i}) - \mathrm{OPT}_{-i}(\mathbf{v})$ is the *externality* that agent i imposes on the other agents by being served. I.e., because i is served the other agents obtain total surplus $\mathrm{OPT}_{-i}(\mathbf{v})$ instead of the surplus $\mathrm{OPT}(\mathbf{v}_{-i})$ that they would have received if i was not served. Hence, we can interpret the surplus maximization mechanism as serving agents to maximize the social surplus and charging each agent the externality imposed on the others.

By Corollary 2.18 and Lemma 3.6 we have the following theorem; and by the optimality of OPT and the assumption that agents follow the dominant truthtelling strategy, we have the following corollary.

Theorem 3.7. *The surplus maximization mechanism is dominant strategy incentive compatible.*

Corollary 3.8. *The surplus maximization mechanism optimizes social surplus in dominant strategy equilibrium.*

The second-price routing auction from Chapter 1 is simply an instantiation of the surplus maximization mechanism where feasible outcomes are subsets of agents whose messages can be simultaneously routed.

It is useful to view the surplus maximization mechanism as a reduction from the mechanism design problem to the non-game-theoretic optimization problem. Given an algorithm that solves the non-game-theoretic optimization problem, i.e., OPT, we can construct the surplus maximization mechanism from it.

Of course, by revenue equivalence, the payment rule of the surplus maximization mechanism is unique up to the payments each agent would make if her value was zero, i.e., $p_i(\mathbf{v}_{-i}, 0)$ for agent i. For instance $p_i = \mathrm{OPT}_{-i}(\mathbf{v})$ is an DSIC payment rule as well with $p_i(\mathbf{v}_{-i}, 0) = \mathrm{OPT}(\mathbf{v}_{-i})$. This payment rule does not satisfy a natural *no-positive-transfers* condition which requires that agents not be paid to participate. It is also possible to design BNE mechanisms, e.g., with first-price semantics, that implement the same outcome in equilibrium as the surplus maximization mechanism (see Exercise 3.2), though unlike the surplus maximization mechanism given above, design of these BNE mechanisms often requires distributional knowledge.

3.3 Profit

Surplus maximization is singular among objectives in that there is a single mechanism that is optimal regardless of distributional assumptions. Essentially: the agents' incentives already aligned with the designer's objective and one only needs to derive the appropriate payments, i.e., the critical values. For general objectives we should not expect to be so lucky.

A non-game-theoretic optimization problem looks to maximize some objective subject to feasibility. Given the input, one can search over outcomes for the one with the highest objective value relative to this input. The outcome produced on one input need not bear any relation to the outcome produced on a (even slightly) different input. Mechanisms, on the other hand, additionally must address agent incentives which impose constraints on the outcomes that the mechanism produces across all possible misreports of the agents. In other words, the mechanism's outcome on one input is constrained by its outcome on similar inputs. Therefore, a mechanism may need to tradeoff its objective performance across inputs.

When the distribution of agent values is specified, e.g., by the common prior, and the designer has knowledge of this prior, such a tradeoff can be optimized. In particular, the prior assigns a probability to each input and the designer can then optimize expected objective value over this probability distribution. The mechanism that results from such an optimization is said to be *Bayesian optimal*. In this section we derive Bayesian optimal mechanism for the objective of profit.

At various points in the remaining sections of this chapter it will be more convenient (and intuitive) to express certain functions in terms of the integral of their derivative. This notation is mathematically imprecise when the derivative is not defined, e.g., because the function is discontinuous. It can be made precise via the Dirac delta function which integrates to a step function; however, we will not describe these details formally. The reader is welcome to, instead, just assume the functions in question are continuous. An example of this is Theorem 3.10, below.

3.3.1 Quantile Space

In single-dimensional Bayesian mechanism design where an agent's value is distributed according to a continuous distribution F there is a one-to-one mapping between the agent's value and her strength relative to the distribution. For instance, an agent with value $v = 0.9$ drawn from $U[0, 1]$ is stronger than 90% of agents and weaker than 10% of agents with values drawn from the same distribution. We refer to indexing of agent from strong ($q = 0$) to weak ($q = 1$) as *quantile*. Importantly, the distribution of an agent's quantile is always $U[0, 1]$.

Definition 3.9. *The* quantile *q of an agent with value $v \sim F$ is the probability that the agent is weaker than a random draw from F. I.e., $q = 1 - F(v)$.*

It will be convenient to express an agent's value as a function of quantile as $v(q) = F^{-1}(1 - q)$. We will overload notation to define allocation and payment rules in quantile space as well. Specifically, "$x(q)$" and "$p(q)$" will be short-hand notation for $x(v(q))$ and $p(v(q))$, respectively. This convention will be extended to other functions as well: if the function is defined on values but applied to a quantile, then by this application we implicitly mean the function composed with the value function, $v(\cdot)$. We can rederive Theorem 2.7 in quantile space as follows.

Theorem 3.10. *Allocation and payment rules x and p are BIC if and only if for all i,*

1. *(monotonicity) $x_i(q_i)$ is monotone non-increasing in q_i, and*
2. *(payment identity) $p_i(q_i) = -\int_{q_i}^{1} v_i(r)\, x_i'(r)\, dr + p_i(1)$,*

where $x_i'(q) = \frac{d}{dq}x_i(q)$ and $p_i(1)$ is the payment made by agent i when her value is at its lowest.

The payment identity of this theorem and Theorem 2.7 are related by a change of variables and integration by parts. Notice that as $x(q)$ is monotone non-increasing, its derivative $x'(q)$ is non-positive; hence, the negation of the integral guarantees a non-negative expected payment.

Proof. See Exercise 3.3. □

3.3.2 Revenue Curves

We start by removing all the complication of mechanisms for multiple agents and consider only a single agent, Alice, desiring a single item. Suppose Alice's value v is drawn from distribution F. How should we sell the item to Alice to maximize our profit?

Suppose we wish to sell to Alice with ex ante probability $\hat{q}$. The most direct way to do this would be to post a price $v(\hat{q})$ as this is the price at which $\mathbf{Pr}_{v\sim F}[v > v(\hat{q})] = \hat{q}$. The revenue obtained by posting such a price is exactly the price times the probability of sale, i.e., $v(\hat{q}) \cdot \hat{q}$.

Definition 3.11. *The* revenue curve $R(\cdot)$ *specifies the revenue as a function of ex ante probability of sale. I.e., $R(q) = v(q) \cdot q$. $R(1)$ and $R(0)$ are defined to be zero.*[5]

We can clearly optimize revenue by taking the derivative of the revenue curve and setting it equal to zero. For example, if F is the uniform distribution $U[0, 1]$ then $F(z) = z$, $v(q) = 1 - q$, $R(q) = q - q^2$, and $R'(q) = 1 - 2q$. The revenue is optimized by pricing at quantile $\hat{q} = 1/2$ (which corresponds to a price $v(1/2) = 1/2$). The uniform distribution is well-behaved in the sense that the revenue, as a function of quantile, increases up to quantile $1/2$, which obtains a revenue of $1/4$, and then decreases. The importance of the derivative in solving for the optimal price can be noted by observing that the derivative is positive but decreasing as r is increased to $1/2$, where it is zero, and then continues to be negative and decreasing afterwards. This optimal revenue is obtained by allocating to Alice when the derivative of the revenue curve at her quantile, i.e., $R'(q)$, is non-negative.

[5] Notice that for distributions with support $[0, h]$ for some value h, $q \in \{0, 1\}$ implies $v(q) \cdot q = 0$, therefore explicitly defining $R(1) = R(0) = 0$ is unnecessary.

3.3.3 Expected Revenue and Virtual Values

Suppose we are given the allocation rule (in quantile space) of an agent (Alice) as $x(q)$. By the payment identity (in quantile space), the payment rule must be $p(q) = -\int_q^1 v(r)\, x'(r)\, dr$. Since Alice's quantile q is drawn from $U[0, 1]$ we can calculate our expected revenue as follows.

$$\mathbf{E}_q[p(q)] = -\int_{q=0}^{1}\int_{r=q}^{1} v(r)\, x'(r)\, dr\, dq$$

This equation can be simplified by swapping the order of integration.

$$\begin{aligned}\mathbf{E}_q[p(q)] &= -\int_{r=0}^{1}\int_{q=0}^{r} dq\, v(r)\, x'(r)\, dr \\ &= -\int_{r=0}^{1} r\, v(r)\, x'(r)\, dr \\ &= -\int_{q=0}^{1} R(q)\, x'(q)\, dq \qquad (3.1)\end{aligned}$$

Equation (3.1) follows from substituting the definition of $R(\cdot)$ and making a change of variables. Denote $\frac{d}{dq}R(q)$ by $R'(q)$. If we integrate the above, by parts, we obtain:

$$\begin{aligned}\mathbf{E}_q[p(q)] &= \int_{q=0}^{1} R'(q)\, x(q)\, dq - \Big[R(q)\, x(q)\Big]_{q=0}^{1} \\ &= \int_{q=0}^{1} R'(q)\, x(q)\, dq. \qquad (3.2)\end{aligned}$$

Equation (3.2) follows from the definition of revenue curves which requires $R(0) = R(1) = 0$. We conclude this analysis by summarizing equations (3.1) and (3.2) as the following lemma.

Lemma 3.12. *An agent with revenue curve $R(\cdot)$ subject to allocation rule $x(\cdot)$ makes expected payment:*

$$\mathbf{E}_q[p(q)] = -\mathbf{E}_q\big[R(q)\, x'(q)\big] = \mathbf{E}_q\big[R'(q)\, x(q)\big]\,.$$

Both of the identities in Lemma 3.12 are useful for understanding the expected payments of agents in BNE. For instance, the former, from equation (3.1), implies that the same allocation rule (in quantile space) on a higher revenue curve gives more revenue.

Corollary 3.13. *If agents 1 and 2 with revenue curves satisfying $R_1(q) \geq R_2(q)$ for all q are subject to the same (in quantile space) allocation rule, i.e., satisfying $x_1(q) = x_2(q)$, then $\mathbf{E}_q[p_1(q)] \geq \mathbf{E}_q[p_2(q)]$.*

The latter identity from Lemma 3.12, from equation (3.2), gives an approach for optimizing revenue. It is instructive to view $R'(\cdot)$ as the marginal increase in revenue we get for selling to Alice with incrementally more probability. For our goal of optimizing expected profit, it suggests selecting x to maximize this *marginal revenue*. Informally: revenue is maximized by optimizing marginal revenue. This principle is standard in microeconomics.

The standard terminology in mechanism design for this marginal revenue is *virtual value*.

Definition 3.14. *The* virtual value *of an agent with quantile q and revenue curve $R(\cdot)$ is the marginal revenue at $q \in (0,1)$:*[6]

$$\phi(q) = R'(q).$$

The virtual surplus *of outcome $\mathbf{x}$ and profile of agent quantiles $\mathbf{q}$ is:*

$$\text{Surplus}(\boldsymbol{\phi}(\mathbf{q}), \mathbf{x}) = \sum\nolimits_i \phi_i(q_i)x_i - c(\mathbf{x}),$$

where $\boldsymbol{\phi}(\mathbf{q}) = (\phi_1(q_1), \ldots, \phi_n(q_n))$.

Often it is useful to write the virtual value in terms of the agent's value, v, and the distribution, F, from which the value is drawn. Evaluating, in value space, the derivative (with respect to quantile) of the revenue curve we obtain:

$$\phi(v) = v - \tfrac{1-F(v)}{f(v)}. \tag{3.3}$$

The following theorem is an immediate consequence of Lemma 3.12 and linearity of expectation.

Theorem 3.15. *A mechanism's expected revenue is equal to its expected virtual surplus, i.e., with allocation rule $\mathbf{x}(\cdot)$ on agents with virtual value functions $\boldsymbol{\phi}(\cdot)$ the expected revenue is:*

$$\mathbf{E}_{\mathbf{q}}\left[\sum\nolimits_i \phi_i(q_i)x_i(\mathbf{q}) - c(\mathbf{x}(\mathbf{q}))\right].$$

It should be noted that the distributional properties of an agent's value can be given equivalently by specifying the distribution F, the value function $v(\cdot)$, the revenue curve $R(\cdot)$, or the virtual value function $\phi(\cdot)$.

3.3.4 Optimal Mechanisms and Regular Distributions

We now derive the optimal mechanism for profit. To do this we again walk through a standard approach in mechanism design. We completely relax the incentive constraints and solve the remaining non-game-theoretic optimization problem. Since expected profit equals expected virtual surplus, this non-game-theoretic optimization problem is to optimize virtual surplus. We then verify that this solution does not violate the incentive constraints (under some conditions). We conclude that (under the same conditions) the resulting mechanism is optimal.

[6] The virtual value for $q \in \{0,1\}$ is irrelevant for optimization for continuous distributions as these realizations of quantile is a measure zero event.

The non-game-theoretic optimization problem of maximizing virtual surplus is that of finding $\mathbf{x}$ to maximize $\text{Surplus}(\boldsymbol{\phi}(\mathbf{v}), \mathbf{x}) = \sum_i \phi_i(v_i)x_i - c(\mathbf{x})$. Let OPT again be the surplus maximizing algorithm. We will care about both the allocation that $\text{OPT}(\boldsymbol{\phi}(\mathbf{v}))$ selects, i.e., $\text{argmax}_{\mathbf{x}} \text{Surplus}(\boldsymbol{\phi}(\mathbf{v}), \mathbf{x})$ and its virtual surplus $\max_{\mathbf{x}} \text{Surplus}(\boldsymbol{\phi}(\mathbf{v}), \mathbf{x})$. Where it is unambiguous we will use notation $\text{OPT}(\boldsymbol{\phi}(\mathbf{v}))$ to denote either of these quantities. Note that this formulation of OPT has no mention of the incentive constraints.

We know from the BIC characterization (Corollary 2.16) that incentive constraints require that the allocation rule be monotone. Thus, the mechanism design problem of finding a BIC mechanism to maximize virtual surplus has an added monotonicity constraint. Yet, even though we did not impose a monotonicity constraint on OPT, if the virtual valuation functions $\phi_i(\cdot)$ are monotone, $\text{OPT}(\boldsymbol{\phi}(\cdot))$ is monotone.

Definition 3.16. *Distribution F is* regular *if its associated revenue curve $R(q)$ is a concave function of q (equivalently: $\phi(\cdot)$ is monotone).*

Many distributions are regular, e.g., uniform, normal, exponential. On the other hand many relevant distributions are irregular, e.g., bimodal.

Lemma 3.17. *For each agent i and any values of other agents $\mathbf{v}_{-i}$, if F_i is regular then i's allocation rule from $\text{OPT}(\boldsymbol{\phi}(\cdot))$ on virtual values is monotone in i's value v_i.*

Proof. Recall from Lemma 3.6 that maximizing surplus is monotone. Meaning, if we find $\mathbf{x}$ to maximize $\text{Surplus}(\mathbf{v}, \mathbf{x})$ then $x_i(\mathbf{v}_{-i}, v_i)$ is monotone in v_i. Therefore $x_i(\boldsymbol{\phi}_{-i}(\mathbf{v}_{-i}), \phi_i(v_i))$ is monotone in $\phi_i(v_i)$, i.e., increasing $\phi_i(v_i)$ does decrease x_i. By the regularity assumption on F_i, $\phi_i(v_i)$ is monotone in v_i. Therefore, increasing v_i cannot decrease $\phi_i(v_i)$ which cannot decrease $x_i(\boldsymbol{\phi}_{-i}(\mathbf{v}_{-i}), \phi_i(v_i))$. □

Since OPT on virtual values is monotone for each agent and any values of other agents it satisfies our strongest incentive constraint. With the appropriate payments (i.e., the "critical values") truthtelling is a dominant strategy equilibrium (recall Corollary 2.18). One way to view the suggested virtual surplus maximization mechanism is as a reduction to surplus maximization, which is solved by the SM mechanism (Mechanism 3.1).

Mechanism 3.2. *The* virtual surplus maximization (VSM) *mechanism with virtual value functions $\boldsymbol{\phi}(\cdot)$ is:*

1. *Solicit and accept sealed bids* $\mathbf{b}$,
2. $(\mathbf{x}, \mathbf{p}') \leftarrow \text{SM}(\boldsymbol{\phi}(\mathbf{b}))$, *and*
3. *for each i, $p_i \leftarrow \phi_i^{-1}(p'_i)$.*

Notice that the payments $\mathbf{p}$ calculated can be viewed as follows. SM on virtual values outputs virtual prices $\mathbf{p}'$. These correspond to the minimum virtual value an agent must have to win. The price an agent pays is the minimum value it must have to win, this can be calculated from the virtual prices via the inverse virtual valuation function.[7]

[7] Assuming virtual valuations are strictly non-decreasing then the inverse virtual valuations are well defined. We defer discussion of the non-strict case to the subsequent section on irregular distributions.

Theorem 3.18. *For regular distributions, the virtual value maximization mechanism is dominant strategy incentive compatible.*

Corollary 3.19. *For regular distributions, the virtual surplus maximization mechanism optimizes expected profit in dominant strategy equilibrium.*

3.3.5 Single-item Auctions

The above description of profit-optimal mechanisms does not offer much in the way of intuition. To get a clearer picture, we consider optimal mechanisms the special case of single-item auctions, i.e., environments where feasible outcomes serve at most one agent. What is the mechanism that optimizes virtual surplus for single-item environments?

First notice that virtual values can be negative. Consider the uniform distribution $U[0, 1]$ where $F(z) = z$ and $f(z) = 1$. From equation (3.3), $\phi(v) = v - \frac{1-F(z)}{f(z)} = 2v - 1$. Thus, $\phi(0) = -1$. If our goal is to optimize virtual surplus we clearly do not want to allocate to any agent with negative virtual value. Recall that virtual values are the derivative of the revenue curve and our analysis of single-agent environments already suggested that we should not allocate to an agent for whom this quantity is negative.

Second notice that among the agents with positive virtual values the virtual surplus is maximized by allocating to the one with the highest virtual value. Conclude the following corollary.

Corollary 3.20. *For regular, single-item environments, the auction that allocates to the agent with the highest non-negative virtual value optimizes expected revenue.*

As virtual valuations are the derivative of the revenue curve, the optimal auction allocates to the agent whose revenue curve is the steepest at her value.

The case where the agents are independent and identically distributed is of special interest. For i.i.d. and regular distributions, the agent with the highest positive virtual value is also the one with the highest value (as the virtual valuation functions are identical). An agent's virtual value is non-negative when her value is at least $\phi^{-1}(0)$. What auction allocates to the agent with the highest value that is at least $\phi^{-1}(0)$? It is the second-price auction with reserve $\phi^{-1}(0)$!

Definition 3.21 (Second-price Auction with reservation price r)**.** *The* second-price auction with reservation price r*, sells the item if any agent bids above* r*. The price the winning agent pays the maximum of the second highest bid and* r*.*

Corollary 3.22. *For i.i.d., regular, single-item environments, the second-price auction with reserve* $\phi^{-1}(0)$ *optimizes expected revenue.*

Notice that the optimal reserve price is not a function of the number of agents. Furthermore, the result can easily be extended to single-item multi-unit auctions where the optimal reserve price is also not a function of the number of units that are for sale. As we will see from Theorem 4.24 in Chapter 4 the same result extends beyond single-item and multi-unit feasibility constraints to those that are downward-closed and satisfy a natural "augmentation" property that is related to substitutability, a.k.a., *matroids*.

While this auction is optimal among all BIC auctions, which is the class of mechanisms we restricted our attention to, (a) the revelation principle implies that no auction has a BNE with higher expected revenue, and (b) it actually satisfies the stronger dominant strategy incentive compatibility constraint. Therefore, we conclude that in a very strong sense, that the second-price auction with reserve price maximizes expected revenue.

We conclude by returning to the two agent $U[0, 1]$ example. As we have calculated, $\phi(v) = 2v - 1$; therefore, $\phi^{-1}(0) = 1/2$. The second-price auction with reserve price 1/2 has the optimal expected revenue. Our calculation at the introduction of this chapter showed this optimal revenue to be 5/12.

3.4 Irregular Distributions

We now turn our attention to the case where the non-game-theoretic optimization problem is not itself inherently monotone. An *irregular* distribution is one for which the the revenue curve is non-concave (in quantile). The virtual valuation functions are non-monotone, therefore, a higher value might result in a lower virtual value. Clearly $\mathrm{OPT}(\phi(\cdot))$ is non-monotone for such a distribution; therefore, there is no payment rule for which it is incentive compatible.

3.4.1 Ironed Revenue Curves

Consider again the problem of selling an item to Alice with ex ante probability $\hat{q}$. We could offer Alice price $v(\hat{q})$ to obtain revenue $R(\hat{q}) = \hat{q} \cdot v_i(\hat{q})$; however, when $R(\cdot)$ is not concave, this approach may not optimize expected revenue.

To see what is going wrong, notice that if we treat Alice the same, regardless of her value, when her quantile is on some interval $[a, b]$ then we can replace her exact virtual valuation with her average virtual valuation on this interval. Figure 3.1(a) depicts a hypothetical non-concave revenue curve; Figure 3.1(c) depicts the corresponding virtual value function. Figure 3.1(d) shows Alice's virtual value averaged on $[a, b]$. Finally, Figure 3.1(b) shows the resulting revenue curve. Notice that the constant virtual valuation over $[a, b]$ results in a linear revenue curve, specifically, the line segment connecting $(a, R(a))$ to $(b, R(b))$. Since $R(\cdot)$ is non-concave this line segment at $\hat{q}$ can be strictly higher than $R(\hat{q})$, as pictured. This process of treating Alice the same on an interval to flatten the virtual valuation function is known as *ironing*.

To sell to Alice with ex ante probability $\hat{q}$ we can pick some interval $[a, b]$ with $a < \hat{q} < b$ and apply the allocation rule

$$x^{\hat{q}}(q) = \begin{cases} 1 & \text{if } q < a \\ \frac{\hat{q}-a}{b-a} & \text{if } q \in [a, b] \\ 0 & \text{if } b < q. \end{cases}$$

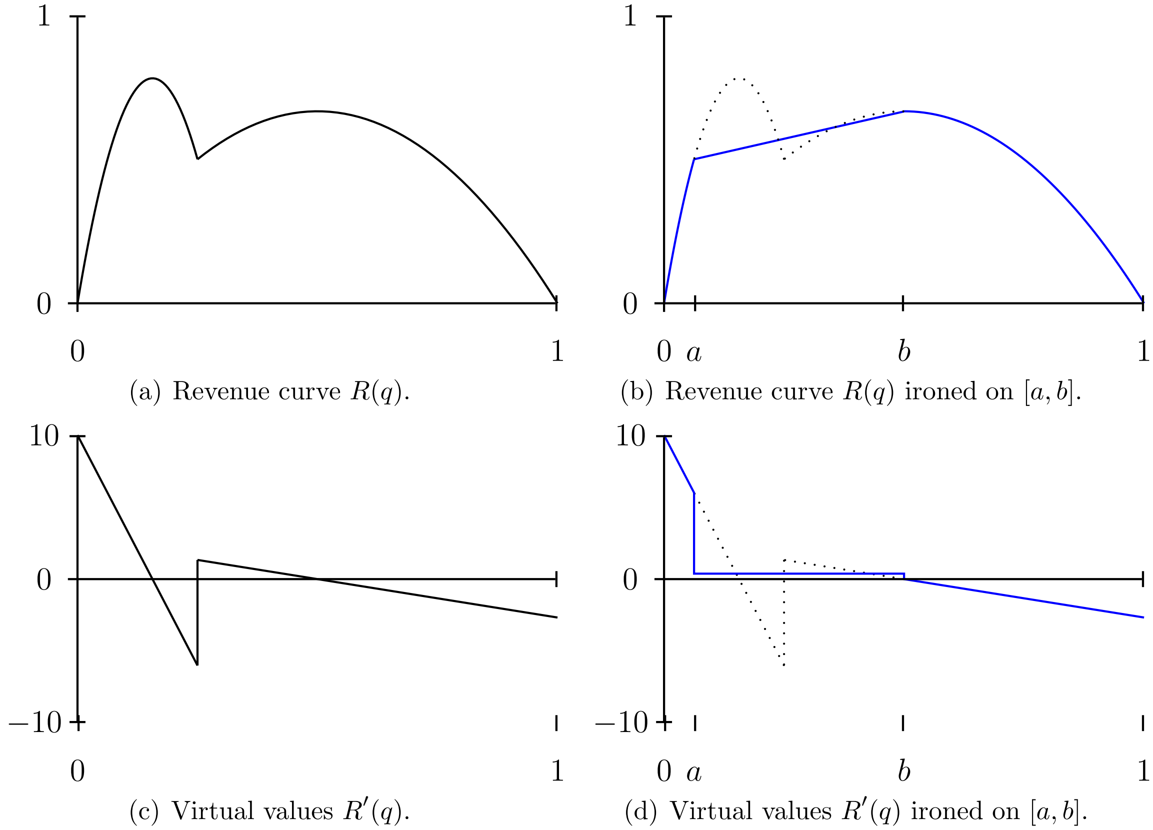


Figure 3.1: On the left is the revenue curve $R(q)$ and virtual valuations $R'(q)$ in quantile space. On the right is the effective revenue curve and virtual valuations when ironed on $[a, b]$. Though it is not necessary for understanding this example, this $R(\cdot)$ comes from bimodal distribution that is $U[0, 2]$ with probability $3/4$ and $U[2, 8]$ with probability $1/4$.

Notice that when Alice's quantile q is realized (i.e., drawn from the uniform distribution) then the probability that Alice is served by $x^{\hat{q}}(\cdot)$ is $1 \times a + \frac{\hat{q}-a}{b-a} \times (b-a) = \hat{q}$. The revenue from such an allocation rule follows directly from Theorem 3.10. It is $R(a) + \frac{\hat{q}-a}{b-a}(R(b) - R(a))$. Notice that this revenue is exactly the value at $\hat{q}$ on the line segment connecting $(a, R(a))$ to $(b, R(b))$, e.g., see Figure 3.1(b). Again, where $R(\cdot)$ is non-concave, the revenue obtained from this randomized rule can be higher than $R(\hat{q})$.

It should be intuitively clear that if we restrict ourselves to allocation rules that treat Alice the same on appropriate subintervals of quantile space we can construct an effective revenue curve $\bar{R}(\cdot)$ equal smallest concave function that upper-bounds the actual revenue curve $R(\cdot)$. This revenue curve is known as the *ironed revenue curve* and its derivative is the *ironed virtual valuation* function.

Definition 3.23. *For $v \sim F$, the* ironed revenue curve*, $\bar{R}(\cdot)$, is smallest concave function that upper-bounds $R(\cdot)$ and the* ironed virtual valuation *function is $\bar{\phi}(q) = \bar{R}'(q)$.*

Ironed intervals of the ironed revenue curve are those with $\bar{R}(q) > R(q)$. The usage of ironed virtual values in place of virtual values as a proxy for an agent's (Alice) expected payment is valid only for mechanisms treat her the same way regardless of where in the interval her quantile lies. Meaning: Alice with quantile $q \in [a, b]$ that is ironed will be served with the same probability as she would have been with any other quantile $q' \in [a, b]$. The following lemma formally states that ironed virtual surplus gives an upper bound on virtual surplus that is tight for mechanisms that respect the ironed intervals.

Lemma 3.24. *An agent's expected payment is upper-bounded by their expected ironed virtual surplus, i.e.,*

$$\mathbf{E}_v[p(v)] \leq \mathbf{E}_q\left[\bar{\phi}(q)x(q)\right].$$

Furthermore, this inequality holds with equality when $\forall q,\ \bar{R}(q) > R(q) \Rightarrow x'(q) = 0$.

Proof. We will start by showing a more precise statement.

$$\begin{aligned} \mathbf{E}_q[p(q)] &= \mathbf{E}_q\left[R'(q) \cdot x(q)\right] + \mathbf{E}_q\left[\bar{R}'(q) \cdot x(q)\right] - \mathbf{E}_q\left[\bar{R}'(q) \cdot x(q)\right] \\ &= \mathbf{E}_q\left[\bar{R}'(q) \cdot x(q)\right] - \mathbf{E}_q\left[\left(\bar{R}'(q) - R'(q)\right) \cdot x(q)\right] \\ &= \mathbf{E}_q\left[\bar{R}'(q) \cdot x(q)\right] + \mathbf{E}_q\left[\left(\bar{R}(q) - R(q)\right) \cdot x'(q)\right]. \end{aligned} \tag{3.4}$$

The last line above follows from writing the expectation as an integral and integration by parts.

Inspecting the second term of equation (3.4) more closely, notice that the difference in the revenue curves is non-negative, as $\bar{R}(\cdot)$ is defined to be an upper-bound on $R(\cdot)$; and the derivative of the allocation rule is non-positive, as the allocation rule is monotone non-increasing in quantile. Therefore, the second term is non-positive and the inequality of the lemma is proven.

Of course, the assumption that $\bar{R}(q) > R(q) \Rightarrow x'(q) = 0$ implies that the second term of (3.4) is identically zero: whenever the first multiplicand is non-zero, the second multiplicand is zero. □

Notice the advantage of $\bar{R}(\cdot)$ over $R(\cdot)$ is two-fold. First, Corollary 3.13 suggests that we can get more revenue from $\bar{R}(\cdot)$ than from $R(\cdot)$. Second, $\bar{R}(\cdot)$ is concave by definition, so ironed virtual valuations are monotone, so ironed virtual surplus maximization results in a monotone allocation rule, so with the appropriate payment rule it is incentive compatible.

In retrospect it should be obvious that the optimal revenue as a function of ex ante sale probability is concave. Given any two IC mechanisms the convex combination of the two mechanisms is IC and its revenue is a convex combination of the two mechanisms revenue.

3.4.2 Optimal Mechanisms

We will now show that for any distribution, the mechanism that maximizes ironed virtual surplus obtains the optimal expected profit. Again we view this mechanism as a reduction to surplus maximization which is solved, e.g., by mechanism SM (Mechanism 3.1). The resulting mechanism is sometimes referred to as the Myerson auction (for single-item environments) or the Myerson mechanism (for general single-dimensional environments) after Roger Myerson.

Mechanism 3.3. *The* ironed virtual surplus maximization (IVSM) *mechanism for distributions with ironed virtual value functions $\bar{\boldsymbol{\phi}}(\cdot)$ is:*

1. *Solicit and accept sealed bids* $\mathbf{b}$*,*
2. $(\mathbf{x}, \mathbf{p}') \leftarrow \mathrm{SM}(\bar{\boldsymbol{\phi}}(\mathbf{b}))$*, and*
3. *calculate payments for each agent from the payment identity.*

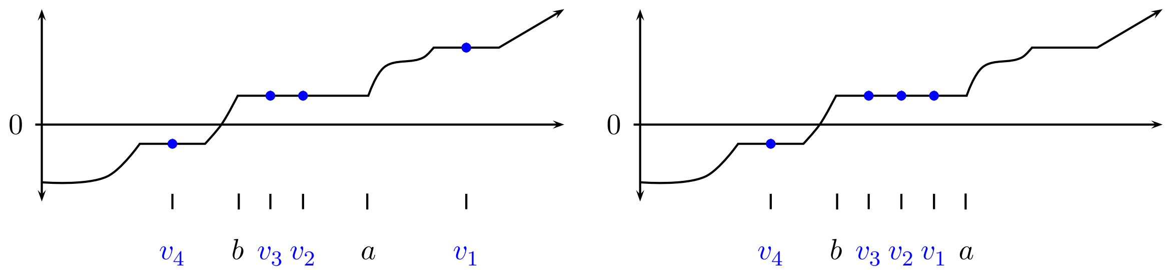
By monotonicity of $\bar{\boldsymbol{\phi}}(\cdot)$ and $\mathrm{OPT}(\cdot)$, $\mathrm{OPT}(\bar{\boldsymbol{\phi}}(\cdot))$ is monotone for each agent and all values of other agents. Therefore, ironed virtual surplus maximization satisfies our strongest incentive constraint. With the appropriate payments (i.e., the "critical values") truthtelling is a dominant strategy equilibrium (recall Corollary 2.18).

Theorem 3.25. *The ironed virtual surplus maximization mechanism is dominant strategy incentive compatible.*

To show that ironed virtual surplus mechanism is optimal we need to argue that it respects the ironed intervals of the ironed revenue curve, i.e., any agent with value within an ironed interval receives the same outcome regardless of where in the interval her value lies.

Lemma 3.26. *For agents with revenue curves $\mathbf{R}(\cdot)$, the allocation rule $\mathbf{x}(\cdot)$ of the ironed virtual surplus maximization mechanism satisfies $\bar{R}_i(q_i) > R_i(q_i) \Rightarrow x_i'(q_i) = 0$ for all i and q_i.*

Proof. Observe that on ironed intervals, i.e., where $\bar{R}(q) > R(q)$, the ironed revenue curve, $\bar{R}(\cdot)$, is linear. This follows from the definition of the ironed revenue curve as the smallest concave function that upper-bounds the revenue curve. Since $\bar{R}(\cdot)$ is linear on this ironed interval, its derivative and, consequently, the ironed virtual function is constant on the interval. The allocation probability of the ironed virtual surplus maximization mechanism



(a) Unique highest ironed virtual value.

(b) Non-unique highest ironed virtual value.

Figure 3.2: The ironed virtual valuation function $\bar{\phi}(v)$ under two realizations of agent values depicting both the case where the highest ironed virtual value is unique and the case where it is not unique.

is determined by the optimization $\mathrm{OPT}(\bar{\boldsymbol{\phi}}(\cdot))$ which is a function only of the ironed virtual values. Since an agent with any quantile within an ironed interval has the same ironed virtual valuation, this optimization must produce an outcome that is constant on the interval. On ironed intervals, therefore, the derivative of the allocation rule is zero. □

Corollary 3.27. *The ironed virtual surplus maximization mechanism optimizes expected profit in dominant strategy equilibrium.*

Like in the regular case, it is quite useful to view this result as a reduction from the problem of profit maximization to the problem of surplus maximization.

Note that unlike the surplus maximization mechanism and the virtual surplus maximization mechanism (for regular distributions) where the continuity assumption on the distribution implies that there is never a tie, the ironed virtual surplus maximization mechanism for irregular distributions may require a tie-breaking policy, for instance, when two agents with distinct values have the same ironed virtual value. Tie breaking can be implemented arbitrarily (as long as it is not a function of the agents' values). Common tie-breaking rules are *lexicographical* and *random*. Lexicographical tie breaking will favor sets of agents with higher indices. Random tie breaking takes the lexicographical ordering on a random permutation of the agent indices. The randomized tie-breaking rule is often desired because it is symmetric.

3.4.3 Single-item Auctions

We consider the special case of single-item auctions to get a clearer picture of exactly what the optimal mechanism is in the case of i.i.d., irregular distributions. Figure 3.2 depicts hypothetical ironed virtual valuation function. Instantiating the agents' values corresponds to picking points on the horizontal axis. The agents' ironed virtual valuations can then be read off the plot. The optimal auction assigns the item to the agent with the highest ironed virtual value. If there is a tie, it picks a random tied agent to win.

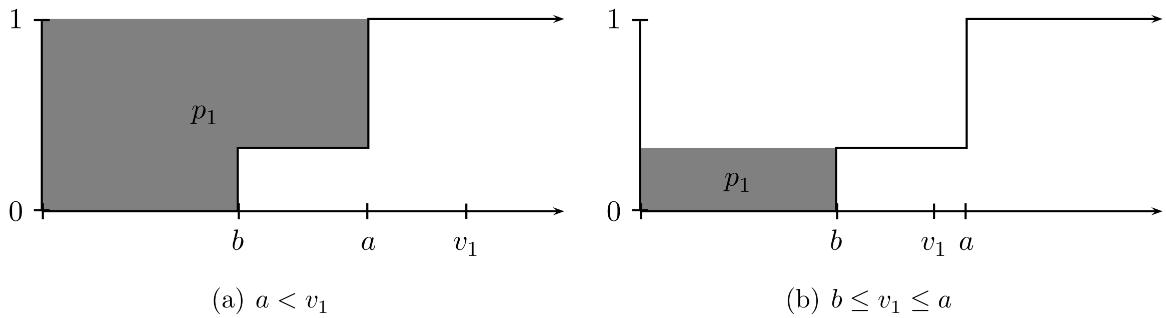


Figure 3.3: The allocation (black line) and payment rule (gray region) for agent 1 given $\mathbf{v}_{-i}$ and the ironed virtual valuation function from Figure 3.2.

Figure 3.2(a) depicts a realization of values for $n = 4$ agents where the highest ironed virtual valuation is unique. What does the ironed virtual surplus maximization do here? It allocates the item to this agent, i.e., agent 1 in the figure. Figure 3.2(b) depicts a second realization of values where the highest ironed virtual valuation is not unique. In this setting the mechanism, we will assume, breaks ties by picking a random tied agent as the winner, i.e., one of agents 1, 2, and 3 in the figure. In general when there is a k-agent tie for the highest ironed virtual valuation then each tied agent wins with probability $1/k$.

It is instructive to calculate the payment an agent must make in expectation over the random tie-breaking rule. Consider the case where there is a unique highest ironed virtual value. The agent with this ironed virtual value wins. To calculate her DSIC payment we need to consider agent i's allocation rule for fixed values $\mathbf{v}_{-i}$ of the other agents. Consider again the example in Figure 3.2(a) and imagine the probability we allocate to agent 1 as a function of v_1. This is

$$x_i(\mathbf{v}_{-i}, z) = \begin{cases} 1 & \text{if } z > a \\ 1/k & \text{if } z \in [b, a] \\ 0 & \text{if } z < b. \end{cases}$$

when $\mathbf{v}_{-i}$ has a $k - 1$ agents in $[b, a]$ tied for the highest ironed virtual valuation. The $1/k$ probability of winning for $z \in [b, a]$ arises from our analysis of what happens when in a k-agent tie. Figure 3.3(a) depicts the allocation and rule payment of this agent. When agent 1 has the unique highest ironed virtual value, i.e., $v_1 > a$, then $p_1 = a - (a - b)/k$.

When agent 1 is tied for the highest ironed virtual value with $k - 1$ other agents, as depicted in Figure 3.3(b), her expected payment is $p_1 = b/k$. Of course, $x_1 = 1/k$ so such a payment can be implemented by charging b to the tied agent that wins and zero to the losers.

Exercises

3.1 In computer networks such as the Internet it is often not possible to use monetary payments to ensure the allocation of resources to those who value them the most.

Computational payments, e.g., in the form of "proofs of work", however, are often possible. One important difference between monetary payments and computational payments is that computational payments can be used to align incentives but do not transfer utility from the agents to the seller. I.e., the seller has no direct value from an agent performing a proof-of-work computation. Define the *residual surplus* as the social surplus less the payments, i.e., $\sum_i (v_i \cdot x_i - p_i) - c(\mathbf{x})$. (For more details, see the discussion of non-monetary payments in Chapter 1.)

Describe the mechanism that maximizes residual surplus when the distribution on agents' values satisfy the *monotone hazard rate* assumption, i.e., $f(v)/(1 - F(v))$ is monotone non-decreasing. Your description should first include a description in terms of virtual values and then you should interpret the implication of the monotone hazard rate assumption to give a simple description of the optimal mechanism. In particular, consider monotone hazard rate distributions in the following environments:

(a) a single-item auction with i.i.d. values,

(b) a single-item auction with non-identical values, and

(c) an environment with general costs specified by $c(\cdot)$ and non-identical values.

3.2 Give a mechanism with first-price payment semantics that implements the social surplus maximizing outcome in equilibrium for any single-dimensional agent environment. Hint: Your mechanism may be parameterized by the distribution.

3.3 Prove from first principles that BNE implies the payment identity of Theorem 3.10. You may assume that $\mathbf{x}(\cdot)$ and $\mathbf{p}(\cdot)$ are continuously differentiable with respect to quantile.

3.4 Consider the non-downward closed environment of *public projects*: either every agent can be served or none of them. I.e., the cost structure satisfies:

$$c(\mathbf{x}) = \begin{cases} 0 & \text{if } \sum_i x_i = 0, \\ 0 & \text{if } \sum_i x_i = n, \text{ and} \\ \infty & \text{otherwise.} \end{cases}$$

(a) Describe the revenue maximizing mechanism for general distributions.

(b) Describe the revenue maximizing mechanism when agents' values are i.i.d. from $U[0, 1]$.

(c) Give an asymptotic, in terms of the number n of agents, analysis of the expected revenue of the optimal public project mechanism when agents' values are i.i.d. from $U[0, 1]$.

Chapter Notes

The surplus-optimal Vickrey-Clarke-Groves (VCG) mechanism is credited to Vickrey (1961), Clarke (1971), and Groves (1973).

The revenue-optimal single-item auction was derived by Roger Myerson (1981). Its generalization to single-dimensional agent environments is an obvious extension. The relationship between revenue-optimal auctions, revenue curves, and *marginal revenue* (equivalent to virtual values) is due to Bulow and Roberts (1989).

Chapter 4

Bayesian Approximation

One of the most intriguing conclusions from the preceding chapter is that for i.i.d., regular, single-item environments the second-price auction with a reservation price is revenue optimal. This result is compelling as the solution it proposes is quite simple; therefore, making it easy to prescribe. Furthermore, reserve-price-based auctions are often employed in practice so this theory of optimal auctions is also descriptive. Unfortunately, i.i.d., regular, single-item environments are hardly representative of the scenarios in which we would like to design good mechanisms. Furthermore, if any of the assumptions are relaxed, reserve-price-based mechanisms are not optimal.

In this chapter we address this deficiency by showing that while reserve-price-based mechanisms are not optimal, they are approximately optimal in a wide range of environments. Furthermore, these approximately optimal mechanisms are more robust, less dependent on the details of the distribution, and provide more conceptual understanding than their optimal counterparts. The approximation factor obtained by most of these reserve-pricing mechanisms is two. Meaning, for the worst distributional assumption, their performance is within a factor two of the optimal mechanism. Of course, in any particular environment these mechanisms may perform better than their worst-case guarantee.

Distributional regularity, as implied by the concavity of the revenue curve, and independence will be instrumental in many of the approximation results obtained, as will two additional structural properties. First, the *monotone hazard rate* condition, a further restriction of regularity, is a property of a distribution that, intuitively, restricts how heavy the tails of the distribution are. An important consequence of the monotone hazard rate assumption is that the optimal revenue and optimal social surplus are within a factor of $e \approx 2.718$ of each other. Second, a *matroid set system* is one that is downward-closed and satisfies an additional "augmentation property." An important consequence of the matroid property is that the surplus maximizing allocation (subject to feasibility) is given by the *greedy-by-value* algorithm: sort the agents by value, then consider each agent in-turn and serve the agent if doing so is feasible.

4.1 Single-item Auctions

We start with single-item auctions and show that the second-price auction with suitably chosen agent-specific reserve prices is always a good approximation to the optimal mechanism.

Mechanism 4.1. *The* second-price auction with reserves $\mathbf{r} = (r_1, \ldots, r_n)$ *is:*

1. *reject each agent i with $v_i < r_i$,*
2. *allocate the item to the highest valued agent remaining (or none if none exists), and*
3. *charge the winner her critical price.*

4.1.1 Regular Distributions

Recall from Chapter 3 that when the agents values are i.i.d. from a regular distribution F (Definition 3.16) then the optimal auction is identically the second-price auction with reserves $\mathbf{r} = (\phi^{-1}(0), \ldots, \phi^{-1}(0))$ where $\phi(\cdot)$ is the virtual value function (Definition 3.14) for F. Further, if we just had a single agent with value $v \sim F$ we would offer her $\phi^{-1}(0)$ to maximize our revenue. This price is often referred to as the *monopoly price.*

Definition 4.1 (monopoly price)**.** *The* monopoly price*, denoted η, for a distribution F is the revenue-optimal price to offer an agent with value drawn from F, i.e., $\eta = \phi^{-1}(0)$.*

Notice that for asymmetric distributions, i.e., where $F_i \neq F_{i'}$, monopoly prices may differ for different agents. Furthermore, the second-price auction with monopoly reserves is not equivalent to the optimal auction when agent values are non-identically distributed. Instead the optimal auction carefully optimizes agents' virtual values at all points of their respective distributions. Therefore, the second-price auction with monopoly reserves has suboptimal revenue.

As an example consider a 2-agent single-item environment with agent 1's value from $U[0, 1]$ and agent 2's value from $U[0, 2]$. The virtual value functions are $\phi_1(v_1) = 2v_1 - 1$ and $\phi_2(v_2) = 2v_2 - 2$. We serve agent 1 whenever $\phi_1(v_1) > \max(\phi_2(v_2), 0)$, i.e., when $v_1 > \max(v_2 - 1/2, 1/2)$. This auction is not the second-price auction with reserves.

The main result of this section is enabled by the following consequence of distributional regularity. The virtual valuation function is monotone in value, therefore, the monopoly price is the boundary between positive virtual values and negative virtual values.

Fact 4.2. *Any agent whose value exceeds the monopoly price has non-negative virtual value.*

We will shortly show that the expected revenue of the second-price auction with monopoly reserves is close to the optimal revenue when the distributions are regular; however, before doing so, consider the following intuition. Either the monopoly-reserves auction and the optimal auction have the same winner or different winners. If they have the same winner then they have the same virtual surplus. By Fact 4.2, the monopoly-reserve auction always has non-negative virtual surplus, so the virtual surplus when both auctions have the same winner is a lower bound on its total virtual surplus and, thus, its revenue. If the two auctions have different winners then the optimal auction's winner is not the agent with the highest value. Of course this winner can pay at most her value, but the monopoly-reserves auction's winner pays at least the second highest value which must be least the value of the optimal auction's winner. Therefore, in this case the payment in the monopoly-reserves auction is higher than the payment in the optimal auction. We conclude that the revenue of the monopoly-reserves auction bounds the optimal revenue in both cases, therefore, it is a 2-approximation. Importantly, this analysis is driven by regularity and Fact 4.2.

Theorem 4.3. *For any regular, single-item environment the second-price auction with monopoly reserves gives a 2-approximation to the optimal expected revenue.*

Proof. Let REF denote the optimal auction and its expected revenue and APX denote the second-price auction with monopoly reserves and its expected revenue; our goal is to show that $\mathrm{REF} \leq 2\,\mathrm{APX}$. Let I be the winner of the optimal auction and J be the winner of the monopoly reserves auction. Notice that both auctions do not sell the item if and only if all virtual values are negative; in this situation define $I = J = 0$. I and J are random variables. With these definitions, $\mathrm{REF} = \mathbf{E}[\phi_I(v_I)]$ and $\mathrm{APX} = \mathbf{E}[\phi_J(v_J)]$.

We start by simply writing out the expected revenue of the optimal auction as its expected virtual surplus conditioned on $I = J$ and $I \neq J$.

$$\mathrm{REF} = \underbrace{\mathbf{E}[\phi_I(v_I) \mid I = J]\,\mathbf{Pr}[I = J]}_{\mathrm{REF}_=} + \underbrace{\mathbf{E}[\phi_I(v_I) \mid I \neq J]\,\mathbf{Pr}[I \neq J]}_{\mathrm{REF}_{\neq}}.$$

We will prove the theorem by showing that both the terms on the right-hand side are bounded from above by APX. For the first term:

$$\begin{aligned}
\mathrm{REF}_= &= \mathbf{E}[\phi_I(v_I) \mid I = J]\,\mathbf{Pr}[I = J] \\
&= \mathbf{E}[\phi_J(v_J) \mid I = J]\,\mathbf{Pr}[I = J] \\
&\leq \mathbf{E}[\phi_J(v_J) \mid I = J]\,\mathbf{Pr}[I = J] + \mathbf{E}[\phi_J(v_J) \mid I \neq J]\,\mathbf{Pr}[I \neq J] \\
&= \mathrm{APX}\,.
\end{aligned}$$

The inequality in the above calculation follows Fact 4.2 (i.e., regularity) as even when $I \neq J$ the virtual value of J must be non-negative. Therefore, the term added is non-negative. For the second term:

$$\begin{aligned}
\mathrm{REF}_{\neq} &= \mathbf{E}[\phi_I(v_I) \mid I \neq J]\,\mathbf{Pr}[I \neq J] \\
&\leq \mathbf{E}[v_I \mid I \neq J]\,\mathbf{Pr}[I \neq J] \\
&\leq \mathbf{E}[p_J \mid I \neq J]\,\mathbf{Pr}[I \neq J] \\
&\leq \mathbf{E}[p_J \mid I \neq J]\,\mathbf{Pr}[I \neq J] + \mathbf{E}[p_J \mid I = J]\,\mathbf{Pr}[I = J] \\
&= \mathrm{APX}\,.
\end{aligned}$$

The first inequality in the above calculation follows because $\phi_i(v_i) = v_i - \frac{1-F_i(v_i)}{f_i(v_i)} \leq v_i$ (since $\frac{1-F_i(v_i)}{f_i(v_i)}$ is always non-negative). The second inequality follows because, among agents that meet their reserve, J is the highest valued agent and I is a lower valued agent and therefore, in a second-price auction J's price is at least I's value. The third inequality follows because payments are non-negative so the term added is non-negative. □

This 2-approximation theorem is tight. We will give a distribution and show that there is an auction with expected revenue $2-\epsilon$ for any $\epsilon > 0$ but the revenue of the monopoly reserves auction is precisely one. The example that shows this separation is easiest to intuit for a distribution that is partly discrete, i.e., one that does not satisfy the continuity assumptions of the preceding chapter. It is of course possible to obtain the same result with continuous distributions.

A distribution that arises in many examples is the *equal-revenue distribution*. The equal-revenue distribution lies on the boundary between regularity and irregularity, i.e., it has constant virtual value. It is called the equal-revenue distribution because the same expected revenue is obtained by offering the agent any price in the distribution's support.

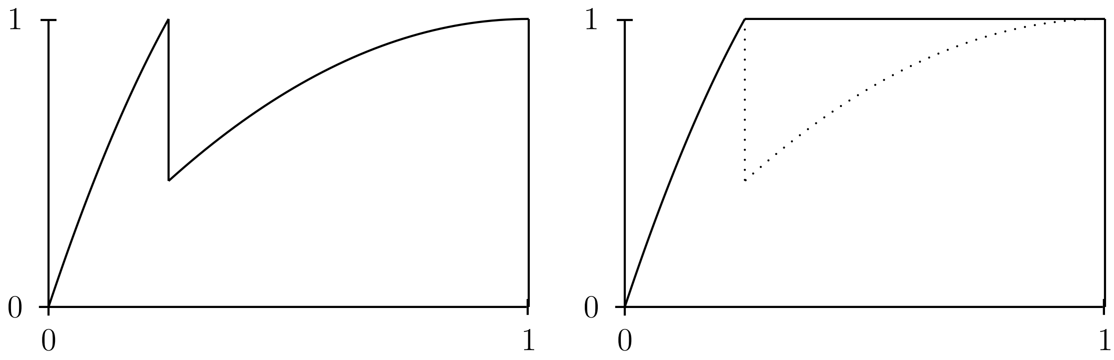
Definition 4.4. *The* equal-revenue distribution *has distribution function* $F(z) = 1 - 1/z$ *and density function* $f(z) = 1/z^2$. *Its support is* $[1, \infty)$.

Consider the asymmetric two-agent single-item auction setting where agent 1's value is deterministically 1 and agent 2's value is distributed according to a variant of the equal-revenue distribution. The monopoly price for the equal-revenue distribution is ill-defined because every price is optimal. Therefore, we slightly perturb the equal-revenue distribution for agent 2 so that her monopoly price is $\eta_2 = 1$. Clearly then, $\boldsymbol{\eta} = (1, 1)$ and the expected revenue of the second-price auction with monopoly reserve is one.

Of course, for this distribution it is easy to see how we can do much better. Offer agent 2 a high price h. If agent 2 rejects this price then offer agent 1 a price of 1. Notice that by the definition of the equal-revenue distribution, agent 2's expected payment is one, but still agent 2 rejects the offer with probability $1 - 1/h$ and the item can be sold to agent 1. The expected revenue of the mechanism is $h \cdot \frac{1}{h} + 1 \cdot (1 - \frac{1}{h}) = 2 - 1/h$. Choosing $\epsilon = 1/h$ gives the claimed result.

4.1.2 Irregular Distributions

Irregular distributions pose a challenge as where virtual valuations are not monotone, an agent whose value is above the monopoly price may yet have a negative virtual value. We first show that the regularity property is crucial to Theorem 4.3; without it the approximation factor of the second-price auction with monopoly reserves can be linear. We next make an aside to discuss *prophet inequalities* from *optimal stopping theory*. Finally, we use prophet inequalities to succinctly describe reserve prices for which the second-price auction is a 2-approximation. These approximation results rely critically on the assumption that the agents' values are independently distributed.



(a) Revenue curve $R(q)$.

(b) Ironed Revenue curve $\bar{R}(q)$.

Figure 4.1: The revenue curve and ironed revenue curve for the Sydney opera house distribution for $n = 2$.

Lower-bound for Monopoly Reserves

The second-price auction with monopoly reserve prices is only a two approximation for regular distributions. The proof of Theorem 4.3 relied on regularity crucially when it assumed that the virtual valuation of the winning agent is always non-negative. We start our exploration of approximately optimal auctions for the irregular case with an example that shows that the second-price auction with monopoly reserves can be a linear factor from optimal even when the agents' values are identically distributed.

Definition 4.5. *The* Sydney opera house distribution *arises from drawing a random variable* $1+U[0, 1-1/n^2]$ *with probability* $1-1/n^2$ *and* $n^2+U[0, 1]$ *with probability* $1/n^2$. *Its revenue curve resembles the Sydney opera house (Figure 4.1).*

The Sydney opera house distribution is bimodal with $R(\cdot)$ maximized at $q = 1$ ($v = 1$) and $q = 1/n^2$ ($v = n^2$). Both give expected revenue of 1. For the purpose of discussion, consider the distribution perturbed sightly so that $\eta = 1$ is the unique monopoly price. The key property of this distribution is that, if there are n agents, the probability of exactly one high-valued agent (i.e., with value at least n^2) is about $1/n$ while the probability of two or more high-valued agents is about $1/(2n^2)$.

The expected revenue of the second-price auction with monopoly reserves is simply the expected second highest value (since the reserve price is never binding). If there is one or fewer high-valued agents then the second highest agent value at most 2. If there are two or more high-valued agents then the second highest agent value is about n^2. The expected revenue is thus about 2.5 (for large n).

To calculate the expected revenue of the optimal auction notice that low-valued agents are completely ironed (Figure 4.1(b)). Suppose there is one high-valued agent, say Alice, and the rest are low valued. If Alice bids a high value she wins. If she bids a low value she is placed in a lottery with all the other agents for a $1/n$ chance of winning. (Of course if she bids below 1 she always loses.) This allocation rule is depicted in Figure 4.2. Alice's payment

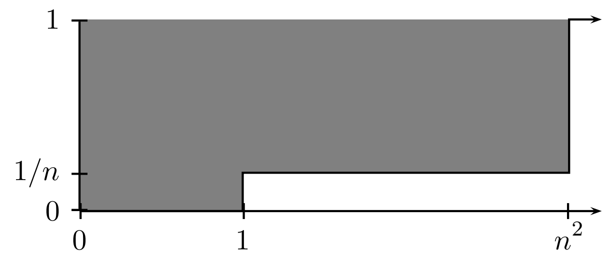


Figure 4.2: The optimal auction allocation rule (black line) and payment (area of gray region) for high-valued Alice when all other agents have low values.

(the area of the gray region in Figure 4.2) in this situation is $n^2 - (n^2 - 1)/n \approx n^2 - n$. There is one such high-valued agent with probability $1/n$ so the total expected revenue is about n.

The conclusion from this rough calculation is that the optimal auction's revenue can be a linear factor more than the second-price auction with monopoly reserves.

Theorem 4.6. *There is an i.i.d., irregular distribution for which the second-price auction with monopoly reserves is a linear approximation to the optimal auction revenue.*

Prophet Inequalities

Consider the following scenario. A gambler faces a series of n games on each of n days. Game i has prize distributed according to F_i. The order of the games and distribution of the game prizes is fully known in advance to the gambler. On day i the gambler *realizes* the value $v_i \sim F_i$ of game i and must decide whether to keep this prize and *stop* or to return the prize and *continue* playing. In other words, the gambler is only allowed to keep one prize and must decide which prize to keep immediately on realizing the prize and before any other prizes are realized.

The gambler's optimal strategy can be calculated by *backwards induction.* On day n the gambler should stop with whatever prize is realized. This results in expected value $\mathbf{E}[v_n]$. On day $n-1$ the gambler should stop if the prize has greater value than $t_{n-1} = \mathbf{E}[v_n]$, the expected value of the prize from the last day. On day $n-2$ the gambler should stop with if the prize has greater value than t_{n-2}, the expected value of the strategy for the last two days. Proceeding in this manner the gambler can calculate a threshold t_i for each day where the optimal strategy is to stop with prize i if and only if $v_i \geq t_i$.

Of course, this optimal strategy suffers from many of the drawbacks of optimal strategies. It is complicated: it takes n numbers to describe it. It is not robust to small changes in the game, e.g., changing of the order of the games or making small changes to distribution i strictly above t_i. It does not allow for any intuitive understanding of the properties of good strategies. Finally, it does not generalize well to give solutions to other similar kinds of games.

Therefore, as we are predisposed to do in this text, we turn to approximation to give a crisper picture. A *threshold strategy* is given by a single t for acceptable prizes and an implicit tie-breaking rule which specifies which prize should be selected if there are multiple prizes above t. The implicit tie-breaking rule in the gambler's game is lexicographical: the gambler takes the first prize with value at least t. Threshold strategies are clearly suboptimal as even on day n if prize $v_n < t$ the gambler will not stop and will, therefore, receive no prize.

Theorem 4.7 (Prophet Inequality)**.** *There exists a threshold strategy such that the expected prize of the gambler is at least half the expected value of the maximum prize. Moreover, one such threshold strategy is the one where the probability that the gambler receives no prize is exactly 1/2. Moreover, this bound is invariant to the tie-breaking rule.*

The *prophet inequality* theorem is suggesting something quite strong. Most importantly it is saying that even though the gambler does not know the realizations of the prizes in advance, he can still do as well in comparison to a "prophet" who does. While this result implies that the optimal (backwards induction) strategy satisfies the same condition, such a implication was not at all clear from the original formulation of the optimal strategy. We can also observe that the result is driven by trading off the probability of not stopping and receiving no prize with the probability of stopping early with a suboptimal prize. The suggested threshold strategy is also quite robust. Notice that the order of the games makes no difference in the determination of the threshold. Furthermore, if the distribution above the threshold changes, nothing on the bound or suggested strategy is affected.

Implicit in definition of a threshold strategy is a tie-breaking rule for resolving which acceptable prize is selected when there is a tie, i.e., more than one prize above the threshold. In fact, the prophet inequality theorem, as is stated, is invariant to the tie-breaking rule. While some tie-breaking rules may bring more or less value to the gambler, the 2-approximation result still holds. This invariance of with respect to the tie-breaking rule means that the prophet inequality theorem has broad implications to other similar settings and in particular to auction design and posted pricing, as we will see later in this section.

Proof of Theorem 4.7. Let REF denote prophet and her expected prize, i.e., the expected maximum prize, $\mathbf{E}[\max_i v_i]$, and APX denote a gambler with threshold strategy t and her expected prize. Define $q_i = 1 - F_i(t)$ as the probability that $v_i \geq t$. Let $\chi = \prod_i (1 - q_i)$ be the probability that the gambler receives no prize. The proof follows in three steps. In terms of t and χ, we get an upper bound on the prophet, REF. Likewise, we get a lower bound on the gambler, APX. Finally, we plug in $\chi = 1/2$ to obtain the bound. If there is no t with $\chi = 1/2$, which is possible if the distributions F_i are not continuous, one of the t that corresponds to the smallest $\chi > 1/2$ or largest $\chi < 1/2$ suffices.

In the analysis below, the notation "$(v_i - t)^+$" is short-hand for "$\max(v_i - t, 0)$."

1. An upper bound on $\text{REF} = \mathbf{E}[\max_i v_i]$:

   Notice that regardless of whether there exists a $v_i \geq t$ or not, REF is at most $t + \max_i (v_i - t)^+$. Therefore,

$$\begin{aligned}\text{REF} &\leq t + \mathbf{E}\big[\max_i (v_i - t)^+\big] \\ &\leq t + \sum\nolimits_i \mathbf{E}\big[(v_i - t)^+\big] .\end{aligned}$$

2. A lower bound on $\text{APX} = \mathbf{E}[\text{prize of gambler with threshold } t]$:

   Clearly, we get t with probability $1-\chi$. Depending on which prize i is the earliest one that is greater than t we also get an additional $v_i - t$. It is easy to reason about the expectation of this quantity when there is exactly one such prize and much more difficult to do so when there are more than one. We will ignore the additional prize we get from the latter case and get a lower bound.

$$\begin{aligned}\text{APX} &\geq (1-\chi)t + \sum\nolimits_i \mathbf{E}\big[(v_i - t)^+ \mid \text{other } v_j < t\big]\, \mathbf{Pr}\big[\text{other } v_j < t\big] \\ &\geq (1-\chi)t + \chi \sum\nolimits_i \mathbf{E}\big[(v_i - t)^+ \mid \text{other } v_j < t\big] \\ &= (1-\chi)t + \chi \sum\nolimits_i \mathbf{E}\big[(v_i - t)^+\big]\,.\end{aligned}$$

   The second inequality follows because $\mathbf{Pr}\big[\text{other } v_j < t\big] = \prod_{j \neq i}(1-q_i) \geq \chi$. The final equality follows because the random variable v_i is independent of random variables v_j for $j \neq i$.

3. Plug in $\chi = 1/2$.

   From the upper and lower bounds calculated, if we can find a t such that $\chi = 1/2$ then $\text{APX} \geq \text{REF}/2$. Incidentally, as t increases $\sum_i \mathbf{E}\big[(v_i - t)^+\big]$ decreases; therefore, we can also solve for $t = \sum_i \mathbf{E}\big[(v_i - t)^+\big]$ to obtain same approximation result.

   Consider χ as a function of t denoted $\chi(t)$. For discontinuous distributions, e.g., ones with point-masses, $\chi(t)$ may be discontinuous. Therefore, there may be no t with $\chi(t) = 1/2$. Let $\chi_1 = \sup\{\chi(t) < 1/2\}$ and $\chi_2 = \inf\{\chi(t) > 1/2\}$. Notice that an arbitrarily small increase in threshold causes the jump from χ_1 to χ_2; let t^* be the limiting threshold for both these χs. Therefore, for $\chi \in \{\chi_1, \chi_2\}$ the lower-bound formula $\text{LB}(\chi) \geq (1-\chi)t^* + \chi \sum_i \mathbf{E}\big[(v_i - t^*)^+\big]$ which is linear in χ.

   We know that this function evaluated at $\chi = 1/2$ (which is not possible to implement) satisfies $\text{LB}(1/2) \geq \text{REF}/2$. Of course it is a linear function of χ so it is maximized on the end-points on which it is valid, namely χ_1 or χ_2. Therefore, one of $\chi \in \{\chi_1, \chi_2\}$ satisfies $\text{LB}(\chi) \geq \text{REF}/2$. If it is optimized at χ_1 then the threshold is exclusive, i.e., the gambler should accept the first prize in (t^*, ∞); if it is optimize at χ_2 then the threshold is exclusive, i.e., the gambler should accept the first prize in $[t^*, \infty)$.

□

Again the independence of the distributions of prizes is fundamental to the prophet inequality.

Uniform Virtual Prices

We return to our discussion of single-item auctions. Our goal in single-item auctions is to select the winner with the highest (positive) ironed virtual value. To draw a connection between the auction problem and the gambler's problem, we note that the gambler's problem in prize space is similar to the auctioneer's problem in ironed-virtual-value space. The gambler aims to maximize expected prize while the auctioneer aims to maximize expected virtual value. A uniform threshold in the gambler's prize space corresponds to a *uniform ironed virtual price* in ironed-virtual-value space. This strongly suggests that a uniform ironed virtual price would make good reserve prices in the second-price auction.

Definition 4.8. *A* uniform ironed virtual price *is* $\mathbf{p} = (p_1, \ldots, p_n)$ *such that* $\bar{\phi}_i(p_i) = \bar{\phi}_{i'}(p_{i'})$ *for all* i *and* i'.

Now compare the second-price auction with a uniform ironed virtual reserve price to the gambler's threshold strategy in the stopping game. The difference is the tie-breaking rule. The second-price auction breaks ties by value whereas the gambler's threshold strategy breaks ties by the ordering assumption on the games (i.e., lexicographically). Recall, though, that the tie-breaking rule was irrelevant for our analysis of the prophet inequality. We conclude the following theorem and corollary where, as in the prophet inequality, the uniform virtual price is selected so that the probability that the item remains sold is about 1/2.

Theorem 4.9. *For any independent, single-item environment the second-price auction with a uniform ironed virtual reserve price is a 2-approximation to the optimal auction revenue.*

It should be clear that what is driving this result is the specific choice of reserve prices and not explicit competition in the auction. Instead of running an auction imagine the agents arrived in any, perhaps worst-case, order and we made each in turn a take-it-or-leave-it offer of her reserve price? Such a *sequential posted pricing* mechanism is also a 2-approximation.

Theorem 4.10. *For any independent, single-item environment a sequential posted pricing of uniform ironed virtual prices is a 2-approximation to the optimal auction revenue.*

Proof. There may be several agents with values at least their posted price. Suppose that in such a situation the agent with the lowest price arrives first. The revenue under this assumption is certainly a lower bound on the revenue of any other ordering. Furthermore, the prophet inequality on virtual values with tie-breaking by "$-p_i$" guarantees a virtual surplus and, therefore, expected revenue that is a 2-approximation to the optimal expected revenue. □

In fact we already saw in Chapter 1 that posted pricing can be a $\frac{e}{e-1} \approx 1.58$ approximation to the optimal mechanism for social surplus for i.i.d. distributions (Theorem 1.6). This approximation factor also holds for revenue and i.i.d., regular distributions.

Corollary 4.11. *For any i.i.d., regular, single-item environment posting a uniform price is an* $\frac{e}{e-1}$ *approximation to the optimal revenue.*

We can also apply the prophet inequality in value space to argue, similarly to Theorem 4.10 that when the values are non-identically distributed posting a uniform price is is a 2-approximation to the optimal social surplus.

4.1.3 Anonymous Reserves

Thus far we have shown that simple reserve-price-based auctions approximate the optimal auction. Unfortunately, agent-specific reserve prices may be impractical for many scenarios, especially ones where agents could reasonably expect some degree of fairness of the auction protocol. Undoubtedly eBay faces such a constraint for the design of their auction. We therefore consider the extent to which an auction with an *anonymous reserve price*, i.e., the same for each agent, can approximate the revenue of the optimal, perhaps non-anonymous, auction.

We start by considering i.i.d., irregular distributions. For i.i.d., irregular distributions, the optimal auction is anonymous, but it is not a reserve-price-based auction. An immediate corollary of Theorem 4.9 is the following.

Corollary 4.12. *For any i.i.d., single-item environment, the second-price auction with an anonymous reserve is a 2-approximation to the optimal auction revenue.*

We now turn to the more challenging question of whether an anonymous reserve price will give a good revenue when the agents' values are not identically distributed. For instance, in the eBay auction the buyers are not identical. Some buyers have higher *ratings* and these ratings are public knowledge. The value distributions for agents with different ratings may generally be distinct. Therefore, the eBay auction may be suboptimal. Surely though, if the eBay auction was very far from optimal, eBay would have switched to a better auction. The theorem below justifies eBay sticking with the second-price auction with anonymous reserve.

Theorem 4.13. *For any independent, regular, single-item environment the second-price auction with an anonymous reserve is a 4-approximation to the optimal auction revenue.*

Proof. This proof can be obtained by extending the proof of Theorem 4.3 or by following a similar approach to the proof of the prophet inequality. We leave the details to Exercise 4.2. □

Note first that it is possible to prove Theorem 4.13 without considering the effect of competition between agents. Therefore, an anonymous price that satisfies the conditions of the theorem is the monopoly price for the distribution of the maximum value. Note second that the bound given in this theorem is not known to be tight. The two agent example with F_1, a point mass at one, and F_2, the equal-revenue distribution, shows that there is a distributional setting where approximation an factor of anonymous reserve pricing is least two.

We now turn to non-identical, irregular distributions. Here we show that anonymous reserve pricing cannot be better than a logarithmic approximation to the optimal (asymmetric) mechanism.

Theorem 4.14. *There is an n-agent, non-identical, irregular, single-item environment for which the second-price auction with an anonymous reserve is an $\Omega(\log n)$-approximation to the optimal auction revenue.*

Proof. The proof follows from analyzing the optimal revenue and the revenue of the second-price auction with any anonymous reserve on the following discrete distribution (which can, of course, be approximated by a continuous distribution). Agent i's value is drawn as:

$$v_i = \begin{cases} n^2/i & \text{w.p. } 1/n^2\text{, and} \\ 0 & \text{otherwise.} \end{cases}$$

The details of this analysis are left to Exercise 4.3. □

4.2 General Feasibility Settings

We now return to more general single-dimensional mechanism design problems, namely, those where the seller faces a combinatorial feasibility constraint. Feasibility constraints that are not downward closed will turn out to be exceptionally difficult and we will give no general approximation mechanisms for them. On the other hand, for regular, downward-closed environments, we show that the surplus maximization mechanism with monopoly reserves is often a 2-approximation. In particular, this result holds if we further restrict the distribution to those satisfying a "monotone hazard rate" condition. It also also holds if we instead restrict the feasible sets to those satisfying a natural augmentation property. These two results are driven by completely different phenomena.

Definition 4.15. *The surplus maximization mechanism with reserves* $\mathbf{r}$ *is:*

1. $\mathbf{v}' \leftarrow \{$*agents with* $v_i \geq r_i\}$.
2. $(\mathbf{x}, \mathbf{p}') \leftarrow \mathrm{SM}(\mathbf{v}')$.
3. *for all* i: $p_i \leftarrow \begin{cases} \max(r_i, p_i') & \text{if } x_i = 1\text{, and} \\ 0 & \text{otherwise,} \end{cases}$

where SM *is the surplus maximization mechanism with no reserves.*

4.2.1 Monotone-hazard-rate Distributions (and Downward-closed Feasibility)

An important property of electronic devices, such as light bulbs or computer chips, is how long they will operate before failing. If we model the lifetime of such a device as a random variable then the failure rate, a.k.a., *hazard rate*, for the distribution at a certain point in time is the conditional probability (actually: density) that the device will fail in the next instant given that it has survived thus far. Device failure is naturally modeled by a distribution with a monotone hazard rate, i.e., the longer the device has been running the more likely it is to fail in the next instant. The uniform, normal, and exponential distributions all have monotone hazard rate. The equal-revenue distribution (Definition 4.4) does not.

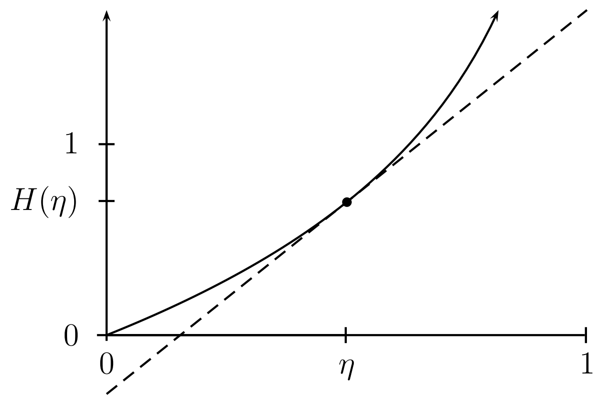


Figure 4.3: The cumulative hazard rate function (solid) for the uniform distribution is $H(v) = -\ln(1 - v)$ and it is lower bounded by its tangent (dashed) at $v = \eta = 1/2$.

Definition 4.16. *The* hazard rate *of distribution F (with density f) is $h(z) = \frac{f(z)}{1-F(z)}$. The distribution has* monotone hazard rate (MHR) *if $h(z)$ is monotone non-decreasing.*

Intuitively distributions with monotone hazard rate are not *heavy tailed*. In fact, the exponential distribution, with $F(z) = 1 - e^{-z}$, is the boundary between monotone hazard rate and non; its hazard rate is constant. Hazard rates are clearly important for optimal auctions as the definition of virtual valuations, expressed in terms of the hazard rate, is $\phi(v) = v - 1/h(v)$. An important property of monotone hazard rate distributions that will enable approximation by the surplus maximization mechanism with monopoly reserves is that, for MHR distributions, the optimal revenue is within a factor of $e \approx 2.718$ of the optimal surplus. We illustrate this with an example, then prove it for the case of a single agent. The proof of the general theorem, we will leave for Exercise 4.4.

Theorem 4.17. *For any downward-closed, monotone-hazard-rate environment, the optimal expected revenue is an e-approximation to the optimal expected surplus.*

To gain some intuition we will look at the exponential distribution. The expected value the exponential distribution (with rate one) is one. This can be calculated from the formula $\mathbf{E}[v] = \int_0^\infty (1 - F(z))\, dz$ with $F(z) = 1 - e^{-z}$. Since the exponential distribution has hazard rate $h(z) = 1$, the virtual valuation formula for the exponential distribution is $\phi(v) = v - 1$. The monopoly price is $\eta = \phi^{-1}(0) = 1$. The probability that the agent accepts the monopoly price is $1/e$ so its expected revenue is $1/e$. The ratio of the expected surplus to expected revenue is e as claimed.

Lemma 4.18. *For any monotone-hazard-rate distribution its expected value is at most e times more than the expected monopoly revenue.*

Proof. Let REF $= \mathbf{E}[v]$ be the expected value and APX $= \eta \cdot (1 - F(\eta))$ be the expected monopoly revenue. Let $H(v) = \int_0^v h(z)dz$ be the *cumulative hazard rate* of the distribution

F. We can write

$$1 - F(v) = e^{-H(v)}, \tag{4.1}$$

an identity that can be easily verified by differentiating $\log(1 - F(z))$. Recall of course that the expectation of $v \sim \mathbf{F}$ is $\int_0^\infty (1 - F(z))\,dz$. To get an upper bound on this expectation we need to upper bound $e^{-H(v)}$ or equivalently lower bound $H(v)$.

The main difficulty is that the lower bound must be tight for the exponential distribution where optimal expected value is exactly e times more than the expected monopoly revenue. Notice that for the exponential distribution the hazard rate is constant; therefore, the cumulative hazard rate is linear. This observation suggests that perhaps we can get a good lower bound on the cumulative hazard rate with a linear function.

Let $\eta = \phi^{-1}(0)$ be the monopoly price. Since $H(v)$ is a convex function (it is the integral of a monotone function). We can get a lower bound $H(v)$ by the line tangent to it at η. See Figure 4.3. I.e.,

$$\begin{aligned} H(v) &\geq H(\eta) + h(\eta)(v - \eta) \\ &= H(\eta) + \tfrac{v-\eta}{\eta}. \end{aligned} \tag{4.2}$$

The second part follows because $\eta = 1/h(\eta)$ by definition. Now we use this bound to calculate a bound on the expectation.

$$\begin{aligned} \text{REF} &= \int_0^\infty (1 - F(z))\,dz = \int_0^\infty e^{-H(z)}\,dz \\ &\leq \int_0^\infty e^{-H(\eta) - \frac{z-\eta}{\eta}}\,dz = e \cdot e^{-H(\eta)} \int_0^\infty e^{-\frac{z}{\eta}}\,dz \\ &= e \cdot e^{-H(\eta)} \cdot \eta = e \cdot (1 - F(\eta)) \cdot \eta = e \cdot \text{APX}\,. \end{aligned}$$

The first and last lines follow from (4.1); the inequality follows from (4.2). □

Importantly, it is not generally the case that the optimal surplus and revenue are within a constant of each other. In fact, for non-monotone-hazard-rate distributions the separation between the optimal revenue and the optimal surplus can be arbitrarily large. To see this consider the equal-revenue distribution with $F(z) = 1 - 1/z$. The expected surplus is given by $\mathbf{E}[v] = 1 + \int_1^\infty \frac{1}{z}\,dz = 1 + [\log z]_1^\infty = \infty$. The expected monopoly revenue, of course, is one.

Shortly we will show that the surplus maximization mechanism with monopoly reserve prices is a 2-approximation to the optimal mechanism for MHR, downward-closed environments. This result derives from the intuition that revenue and surplus are close. The following lemma reformulates this intuition.

Lemma 4.19. *For any monotone-hazard-rate distribution F and $v \geq \eta$, $\phi(v) + \eta \geq v$.*

Proof. Since $\eta = \phi^{-1}(0)$ it solves $\eta = 1/h(\eta)$. By MHR, $v \geq \eta$ implies $h(v) \geq h(\eta)$. Therefore,

$$\phi(v) + \eta = v - 1/h(v) + 1/h(\eta) \geq v.$$

□

Theorem 4.20. *For any independent, monotone hazard rate, downward-closed environment the revenue of the surplus maximization mechanism with monopoly reserves is a 2-approximation to the optimal mechanism revenue.*

Proof. Let APX denote the surplus maximization mechanism with monopoly reserves (and its expected revenue) and let REF denote the revenue-optimal mechanism (and its expected revenue). We start with two bounds on APX and then add them.

$$\begin{aligned} \text{APX} &= \mathbf{E}[\text{APX's virtual surplus}]\,, \text{ and} \\ \text{APX} &\geq \mathbf{E}[\text{APX's winners' reserve prices}]\,. \end{aligned}$$

So, summing these two equations and letting $\mathbf{x}(\mathbf{v})$ denote the allocation rule of APX,

$$\begin{aligned} 2 \cdot \text{APX} &\geq \mathbf{E}[\text{APX's winners' virtual values + reserve prices}] \\ &= \mathbf{E}\Big[\sum\nolimits_i (\phi_i(v_i) + \eta_i) x_i(\mathbf{v})\Big] \\ &\geq \mathbf{E}\Big[\sum\nolimits_i v_i x_i(\mathbf{v})\Big] = \mathbf{E}[\text{APX's surplus}] \\ &\geq \mathbf{E}[\text{REF's surplus}] \geq \mathbf{E}[\text{REF's revenue}] = \text{REF}\,. \end{aligned}$$

The second inequality follows from Lemma 4.19. By downward closure, neither REF nor APX sells to agents with negative virtual values. Of course, APX maximizes the surplus subject to not selling to agents with negative virtual values. Hence, the third inequality. The final inequality follows because the revenue of any mechanism is never more than its surplus. □

4.2.2 Matroid Feasibility (and Regular Distributions)

In Chapter 3 we saw that the second-price auction with the monopoly reserve was optimal for i.i.d., regular, single-item environments. In the first section of this chapter we showed that the second-price auction with monopoly reserves is a 2-approximation for regular, single-item environments. A very natural question to ask at this point is to what extent we can relax the single-item feasibility constraint and still preserve these results. Often the answer to such questions is *matroids*.

Definition 4.21. *A set system is $(E, \mathcal{I})$ where E is the* ground set *of elements and $\mathcal{I}$ is a set of feasible (a.k.a.,* independent*) subsets of E. A set system is a* matroid *if it satisfies:*

- downward closure*: subsets of independent sets are independent.*
- augmentation*: given two independent sets, there is always an element from the larger whose union with the smaller is independent.*

$$\forall I, J \in \mathcal{I},\ |J| < |I| \Rightarrow \exists e \in I \setminus J,\ \{e\} \cup J \in \mathcal{I}.$$

The augmentation property trivially implies that all maximal independent sets of a matroid have the same cardinality. This cardinality is known as the *rank* of the matroid. The most important theorem about matroids is that the *greedy-by-value* algorithm optimizes surplus. In fact, the most succinct proofs of many mechanism design results in matroid environments are obtained as consequences of the optimality of the greedy-by-value algorithm.

Algorithm 4.1. *A* greedy-by-value algorithm *is*

1. *Sort the agents in decreasing order by value.*

2. $\mathbf{x} \leftarrow \mathbf{0}$ *(the null assignment).*

3. *For each agent i (in this sorted order),*

   if $(\mathbf{x}_{-i}, 1)$ is feasible, $x_i \leftarrow 1$.

   (I.e., serve i if i can be served along side previously served agents.)

4. *Output* $\mathbf{x}$.

Theorem 4.22. *The greedy-by-value algorithm is selects the independent set with largest surplus for all valuation profiles if and only if feasible sets are a matroid.*

Proof. The "only if" direction follows from showing, by counter example, (a) downward-closure is necessary and (b) if the set system is downward-closed then the augmentation property is necessary; the "if" direction is as follows.

Let r be the *rank* of the matroid. Let $I = \{i_1, \ldots, i_r\}$ be the set of elements selected in the surplus maximizing assignment, and let $J = \{j_1, \ldots, j_r\}$ be the set of elements selected by greedy-by-value. The surplus from serving a subset S of the agents is $\sum_{i \in S} v_i$.

Assume for a contradiction that the surplus of set I is strictly more than the surplus of set J, i.e., greedy-by-value is not optimal. Assume the items of I and J are indexed in decreasing order. Therefore, there must exist a first index k such that $v_{i_k} > v_{j_k}$. Let $I_k = \{i_1, \ldots, i_k\}$ and let $J_{k-1} = \{j_1, \ldots, j_{k-1}\}$. Applying the augmentation property to sets I_k and J_{k-1} we see that there must exist some element $i \in I_k \setminus J_{k-1}$ such that $J_{k-1} \cup \{i\}$ is feasible. Of course, $v_i \geq v_{i_k} > v_{j_k}$ which means agent i was considered by greedy-by-value before it selected j_k. By downward-closure and feasibility of $J_{k-1} \cup \{i\}$, when i was considered by greedy-by-value it was feasible. By definition of the algorithm, i should have been added; this is a contradiction. □

The following matroids will be of interest.

- In a *k-uniform matroid* all subsets of cardinality at most k are independent. The 1-uniform matroid corresponds to a single-item auction; the k-uniform matroid corresponds to a k-unit auctions.

- In a *transversal matroid* the ground set is the set of vertices of part A of the bipartite graph $G = (A, B, E)$ (where vertices A are adjacent to vertices B via edges E) and independent sets are the subsets of A that can be simultaneously matched. E.g., if A is people, B is houses, and an edge from $a \in A$ to $b \in B$ suggests that b is acceptable to a; then the independent sets are subsets of people that can simultaneously be assigned acceptable houses with no two people assigned the same house. Notice that k-uniform matroids are the special case where $|B| = k$ and all houses are acceptable. Therefore, transversal matroids represent a generalization of k-unit auctions to a market environment where not all items are acceptable to every agent.

- In a *graphical matroid* the ground set is the set of edges in graph $G = (V, E)$ and independent sets are acyclic subgraphs (i.e., a *forest*). Maximal independent sets in a connected graph are spanning trees. The greedy-by-value algorithm for graphical matroids is known as *Kruskal's algorithm* and is studied in every introductory algorithms text.

It is important to be able to argue that a set system satisfies the augmentation property to verify that it is a matroid. As an example we show that acyclic subgraphs are indeed a matroid. For graph $G = (V, E)$ with vertex set $V = \{1, \ldots, n\}$ and edge set $E \subseteq V \times V$ the subgraph induced by edge set $E' \subseteq E$ is $G' = (V, E')$.

Lemma 4.23. *For graph $G = (V, E)$ with $\mathcal{I}$ the set of sets of edges for induced subgraphs that are acyclic, set system $(E, \mathcal{I})$ is a matroid.*

Proof. Downward closure is easy to argue: given an acyclic subgraph, removing edges cannot create cycles.

To show the augmentation property, consider the number of connected components of an acyclic subgraph $G' = (V, E')$ with $m' = \left|E'\right|$ edges. By induction the number of connected components is $n - m'$: when $m' = 0$ each vertex is its own connected component; the addition of any edge that does not create a cycle must connect two connected components thereby reducing the number of connected components by one.

Now consider two acyclic subgraphs given by edge sets $I, J \subseteq E$ satisfying the assumption of the augmentation property, i.e., that $|J| < |I|$. We conclude that the number of connected components of graph (V, J) is strictly more than that of (V, I) which is at least that of connected components of the graph $(V, I \cup J)$.

Consider adding edges $I \setminus J$ one at a time to J and let e be the first such edge that decreases the number of connected components. Then $(V, J \cup \{e\})$ is acyclic, as e connects two connected components of (V, J) and therefore does not create a cycle; and the augmentation property is satisfied. □

Since greedy by value is the optimal algorithm for matroid environments; the revenue-optimal mechanism for matroid environments is *greedy by ironed virtual value*. Of course, for i.i.d., regular distributions greedy by ironed virtual value is simply greedy by value with a reserve price of $\phi^{-1}(0) = \eta$. This is exactly the surplus maximization mechanism with reserve price η. This argument is implicitly taking advantage of the fact that the greedy-by-value algorithm is ordinal, i.e., only the relative order of values matters in determining the optimal feasible allocation.

Theorem 4.24. *For any i.i.d., regular, matroid environment, the surplus maximization mechanism with monopoly reserve price optimizes expected revenue.*

Proof. The optimal algorithm for maximizing virtual surplus (hence: the optimal mechanism) is greedy by virtual value with agents with negative virtual value discarded. In the regular case, i.e., when virtual values are monotone and identical, sorting by virtual values is the same as sorting by values and discarding negative virtual values is the same as discarding values less than the monopoly price. □

Of course, in matroid environments that are inherently asymmetric, the i.i.d. assumption is overly restrictive. It turns out that the surplus maximization mechanism with (agent-specific) monopoly reserves continues to be a good approximation even when the agents' values are non-identically distributed.

Theorem 4.25. *In regular, matroid environments the revenue of the surplus maximization mechanism with monopoly reserves is a 2-approximation to the optimal mechanism revenue.*

There are two very useful facts about the surplus maximization mechanism in matroid environments that enable the proof of Theorem 4.25. The first shows that the critical value (which determine agent payments) for an agent is the value of the agent's "best replacement." The second shows that the surplus maximization mechanism is point-wise revenue monotone, i.e., if the values of any subset of agents increases the revenue of the mechanism does not decrease. These properties are summarized by Lemma 4.29 and Lemma 4.28, below. The formal proofs of Theorem 4.25 and Lemma 4.28 are left for Exercises 4.5 and 4.6, respectively.

Definition 4.26. *If $I \cup \{i\} \in \mathcal{I}$ is surplus maximizing set containing i then the* best replacement *for i is $j = \operatorname{argmax}_{\{k\,:\,I\cup\{k\}\in\mathcal{I}\}} v_k$.*

Definition 4.27. *A mechanism is* revenue monotone *if for all valuation profiles $\mathbf{v} \geq \mathbf{v}'$ (i.e., for all i, $v_i \geq v_i'$), the revenue of the mechanism on $\mathbf{v}$ is no worse than its revenue on $\mathbf{v}'$.*

Lemma 4.28. *In matroid environments, the surplus maximization mechanism is revenue monotone.*

Lemma 4.29. *In matroid environments, the surplus maximization mechanism on valuation profile $\mathbf{v}$ has the critical values $\boldsymbol{\tau}$ satisfying, for each agent i, $\tau_i = v_j$ where j is the* best replacement *for i.*

Proof. The greedy-by-value algorithm is ordinal, therefore we can assume without loss of generality that the cumulative values of all subsets of agents are distinct. E.g., add a $U[0, \epsilon]$ random perturbation to each agent value, the event where two subsets sum to the same value has measure zero, and as $\epsilon \to 0$ the critical values for the perturbation approach the critical values for the original valuation profile, i.e., from equation (4.3).

To proceed with the proof, consider two alternative calculations of the critical value for player i. The first is from the proof of Lemma 3.6 where $\mathrm{OPT}(\mathbf{v}_{-i})$ and $\mathrm{OPT}_{-i}(\mathbf{v})$ are optimal surplus from agents other than i with i is not served and served, respectively.

$$\tau_i = \mathrm{OPT}(\mathbf{v}_{-i}) - \mathrm{OPT}_{-i}(\mathbf{v}). \tag{4.3}$$

The second is from the greedy algorithm. Sort all agents except i by value, then consider placing agent i at any position in this ordering. Clearly, when placed first i is served. Let j be the first agent after which i would not be served. Then,

$$\tau_i = v_j. \tag{4.4}$$

Now we compare these the two formulations of critical values given by equations (4.3) and (4.4). Notice that if i is ordered after j and this causes i to not be served, then j must be served as this is the only possible difference between i coming before or after j. Therefore, agent j must be served in the calculation of $\mathrm{OPT}(\mathbf{v}_{-i})$. Let $J \cup \{j\}$ be the agents served in $\mathrm{OPT}(\mathbf{v}_{-i})$ and let I be the agents served in $\mathrm{OPT}_{-i}(\mathbf{v})$ (which does not include i). We can deduce (denoting by $v(S) = \sum_{k \in S} v_k$):

$$\begin{aligned} v_j &= \tau_i \\ &= \mathrm{OPT}(\mathbf{v}_{-i}) - \mathrm{OPT}_{-i}(\mathbf{v}) \\ &= v_j + v(J) - v(I). \end{aligned}$$

We conclude that $v(I) = v(J)$ which, by the assumption that the cumulative values of distinct subsets are distinct, implies that $I = J$. Meaning: j is a replacement for i; furthermore, by optimality of $J \cup \{j\}$ for $\mathrm{OPT}(\mathbf{v}_{-i})$, j must be the best, i.e., highest valued, replacement. □

Exercises

4.1 Show that there exists an i.i.d. matroid environment for which the surplus maximization mechanism with anonymous reserve is no better than an $\Omega(\log n / \log \log n)$-approximation to the Bayesian optimal mechanism.

4.2 Show that for any non-identical, regular distribution of agents, there exists a reserve price such that second-price auction with an anonymous reserve price obtains 4-approximation to the optimal single-item auction revenue.

4.3 Prove Theorem 4.14 by analyzing the revenue of the optimal auction and the second-price auction with any anonymous reserve when the agents values distributed as:

$$v_i = \begin{cases} n^2/i & \text{w.p. } 1/n^2, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

(a) Show that the expected revenue of the optimal auction is $\Omega(\log n)$.

(b) Show that for any anonymous reserve, the expected revenue of the second-price auction conditioned on exactly one agent having a non-zero value is $O(n)$.

(c) Show that for any anonymous reserve, the expected revenue of the second-price auction is $O(1)$.

(d) Combine the above three steps to prove the theorem.

4.4 Consider the following *surplus maximization mechanism with lazy monopoly reserves* where, intuitively, we run the the surplus maximization mechanism SM and then reject any winner i whose value is below her monopoly price η_i:

1. $(\mathbf{x}', \mathbf{p}') \leftarrow \mathrm{SM}(\mathbf{v})$,
2. $x_i = \begin{cases} x_i' & \text{if } v_i \geq \eta_i \\ 0 & \text{otherwise, and} \end{cases}$
3. $p_i = \max(\eta_i, p_i')$.

Prove that the revenue of this mechanism is an e-approximation to the optimal social surplus in any downward-closed, monotone-hazard-rate environment. Conclude Theorem 4.17 as a corollary.

4.5 Show that in regular, matroid environments the surplus maximization mechanism with monopoly reserves gives a 2-approximation to the optimal mechanism revenue, i.e., prove Theorem 4.25. Hint: This result can be proved using Lemmas 4.29 and 4.28 and a similar argument to the proof of Theorem 4.3.

4.6 A mechanism $\mathcal{M}$ is *revenue monotone* if for all pairs of valuation profiles $\mathbf{v}$ and $\mathbf{v}'$ such that for all i, $v_i \geq v_i'$, the revenue of $\mathcal{M}$ on $\mathbf{v}$ is at least its revenue on $\mathbf{v}'$. It is easy to see that the second-price auction is revenue monotone.

1. Give a single-parameter agent environment for which the surplus maximization mechanism (Mechanism 3.1) is not revenue monotone.
2. Prove that the surplus maximization mechanism is revenue monotone in matroid environments.

Chapter Notes

For non-identical, regular, single-item environments, the proof that the second-price auction with monopoly reserves is a 2-approximation is from Chawla et al. (2007). For the same environment, the second-price auction with anonymous reserve was shown to be a 4-approximation by Hartline and Roughgarden (2009).

The prophet inequality theorem was proven by Samuel-Cahn (1984) and the connection between prophet inequalities and mechanism design was first made by Taghi-Hajiaghayi et al. (2007). For irregular distributions and single-item auctions, the 2-approximation for the second-price auction with constant virtual reserves (and the related sequential posted pricing mechanism) was given by Chawla et al. (2010a).

Beyond single-item environments, Hartline and Roughgarden (2009) show that the surplus maximization mechanism with monopoly reserves is a 2-approximation to the optimal mechanism both for regular, matroid environments (generalizing the single-item auction proof of Chawla et al., 2007) and for monotone-hazard-rate, downward-closed environments.

The structural comparison between optimal surplus and optimal revenue for downward-closed, monotone-hazard-rate environments was given by Dhangwatnotai et al. (2010). The analysis of greedy-by-value under matroid feasibility was initiated by Joseph Kruskal (1956) and there are books written solely on the structural properties of matroids, see e.g., Welsh (2010) or Oxley (2006). Mechanisms based on the greedy algorithm were first studied by Lehmann et al. (2002) where it was shown that even when these algorithms are not optimal, mechanisms derived from them are incentive compatible.

The first comprehensive study of the revenue of the surplus maximizing mechanism in matroid environments was given by Talwar (2003); for instance, he proved critical values for matroid environments are given by the best replacement. The revenue monotonicity for matroid environments and non-monotonicity for non-matroids is discussed by Dughmi et al. (2009), Ausubel and Milgrom (2006), and Day and Milgrom (2007).

Chapter 5

Prior-independent Approximation

In the last two chapters we discussed mechanism that performed well for a given Bayesian prior distribution. Assuming such a Bayesian prior is natural when deriving mechanisms for games of incomplete information as Bayes-Nash equilibrium requires the prior distribution to be common knowledge. For reasons to be discussed, it is desirable to relax this known prior assumption. The objective of prior-independent mechanism design is to identify a single mechanism that always has good performance, e.g., under any distributional assumption. A slightly relaxed objective would be to constrain the distributions to fall within some broad, natural class, e.g., i.i.d., regular distributions.

As is evident from our analysis of Bayesian optimal auctions, e.g., for profit maximization, for any auction that one might consider good, there is a value distribution for which another auction performs strictly better. This is obvious because optimal auctions for distinct distributions are generally distinct. While no auction is optimal for all value distributions, there may be a single auction that is approximately optimal across a wide range of distributions.

In this chapter we will take two approaches to prior-independent mechanisms. The first is a "resource" augmentation, a.k.a., bicriteria, approach. We will show that increasing competition, e.g., by recruiting more agents, and running the (prior-independent) surplus maximization mechanism sometimes earns more revenue more than the revenue-optimal mechanism would have without the increased competition. The second approach is to design mechanisms that do a little market analysis on the fly. We will show that for a large class of environments there is a single mechanism that approximates the revenue of the optimal mechanism.

5.1 Motivation

Since prior-independence is not without loss it is important to consider the motivation behind going from Bayesian optimal to prior-independent approximation mechanisms.

Remember why we adopted Bayesian optimality in the first place: we are considering a game of incomplete information and in games of incomplete information, in order to argue about strategic choice, we needed to formalize how players deal with uncertainty. In a

Stackelberg game, instead of moving simultaniously, players make actions in a prespecified order. We can view the mechanism designer as a player who moves first and the agents as players who (simultaniously) move second. To analize the Bayes-Nash equilibrium in such a Stackelberg game, the designer bases her strategy on the common prior. Without such prior knowledge, prediction the designer's strategy is ill posed.

Now consider from where the designer may have learned the prior-distributions. There are two most logical candidates. The first is from the designer's history in interacting with these or similar agents. The problem with this point of view is that the earlier agents may strategize so that information about their preferences is not exploited by the designer later. In fact, if a monopolist cannot commit not exploit the agents using information from prior interaction then the socially efficient (i.e., surplus maximizing) outcome is the only equilibrium. Its revenue can be far from the optimal revenue. This phenomenon is referred to as the *Coase Conjecture* (a theorem).

The second candidate is *market analysis*. The designer can hire a marketing firm to survey the market and provide distributional estimates of agent preferences. This mode of operation is quite reasonable in large markets. However, in large markets mechanism design is not such an interesting topic; each agent will have little impact on the others and therefore the designer may as well stick to posted-pricing mechanisms. Indeed, for commodity markets posted prices are standard in practice. Mechanisms on the other hand are most interesting in small, a.k.a., *thin*, markets. Contrast the large market for personal computers to the thin market for super computers. There may be five organizations in the world in the market for super computers. How would a designer optimize a mechanism for selling super computers? First, even if the agents' values do come from a distribution, the only way to sample the distribution is to interview the agents themselves. Second, even if we did interview the agents, the most data points we could obtain is five. This is hardly enough for statistical approaches to be able to estimate the distribution of agent values. This strongly motivates a question (which we will also answer in this chapter) that is closely related to prior-independent mechanism design: How many samples from a distribution are necessary to design a mechanism that can approximate the optimal mechanism for the distribution?

There are other reasons to consider prior-independent mechanism design besides the questionable origin of prior information. The most striking of which is the frequent inability of a designer to redesign a new mechanism for each scenario in which she wishes to run a mechanism. This is not just a concern, in many settings it is a principle. Consider the standard Internet routing protocol TCP/IP. This is the protocol responsible for sending emails, browsing web pages, steaming video, etc. Notice that the workloads for each of these tasks is quite different. Emails are small and can be delivered with several minutes delay without issue. Web pages are small, but must be delivered immediately. Comparably, video streaming requires a high responsiveness and a large bandwidth. There is not the flexibility to install new protocols in Internet routers each time a new network usage pattern arises. Instead, a good protocol, such as TCP/IP, should work pretty well in any setting, perhaps ones well beyond the imaginations of the original designers of the Internet.

The final motivation we will discuss for prior-independent mechanism design is that the solution of Bayesian optimal (or approximate) mechanisms is incomplete. It solves the problem of what a designer should do who knows the prior-distribution, but in many real situations a designer may not have such knowledge. Requiring the designer to acquire distribution information from "outside the system", therefore, does not completely solve the designer's problem.

5.2 "Resource" Augmentation

In this section we describe a classical result from auction theory which shows that a little more competition in a surplus maximizing mechanism revenue-dominates the profit maximizing mechanism without the increased competition. From an economic point of view this result questions the *exogenous-participation* assumption, i.e., that there a certain number of agents, say n, that will participate in the mechanism. If, for instance, agents only participate in the mechanism if their utility from doing so is large enough, i.e., with *endogenous participation*, then running an optimal mechanism may decrease participation and then result in a lower revenue than the surplus maximizing mechanism.

On the other hand, the suggestion of this result, that a little increasing competition can ensure good revenue, is inherently prior-independent. The designer does not need to know the prior distribution to market her service so as to attract more agent participation.

5.2.1 Single-item Auctions

The following theorem is due to Jeremy Bulow and Peter Klemperer and is known as the Bulow-Klemperer Theorem.

Theorem 5.1. *For i.i.d., regular, single-item environments, the expected revenue of the second-price auction on $n + 1$ agents is at least the expected revenue of the optimal auction on n agents.*

Proof. First consider the following question. What is the optimal single-item auction for $n + 1$ agents that always sells the item? The requirement to always sell the item means that, even if all virtual values are less than zero, a winner must still be selected. Clearly the optimal such auction is the one that assigns the item to the agent with the highest virtual value. Since the distribution is i.i.d. and regular, the agent with the highest virtual value is the agent with the highest value. Therefore, this optimal auction that always sells the item is the second-price auction.

Now consider an $n + 1$ agent mechanism that we will call "mechanism B". Mechanism B runs the optimal auction on the first n agents and if this auction fails to sell the item, it gives the item away for free to the last agent. By definition, B's expected revenue is equal to the expected revenue of the optimal n-agent auction. It is, however, an $n + 1$-agent auction that always sells. Therefore, its revenue is at most that of the optimal $n + 1$-agent auction that always sells.

We conclude that the expected revenue of the second-price auction with $n + 1$ agents is at least that of mechanism B which is equal to that of the optimal auction for n agents. □

This resource augmentation result provides the beginning of a prior-independent theory for mechanism design. For instance, we can easily obtain a prior-independent approximation result as a corollary to Theorem 5.1 and Theorem 5.2, below.

Theorem 5.2. *For i.i.d., regular, single-item environments the optimal* $(n-1)$*-agent auction is an* $\frac{n}{n-1}$*-approximation to the optimal* n*-agent auction revenue.*

Proof. See Exercise 5.1. □

Corollary 5.3. *For i.i.d., regular, single-item environments with* n *agents, the second-price auction is an* $\frac{n}{n-1}$*-approximation to the optimal auction revenue.*

5.2.2 Matroid Environments

Unfortunately, the "just add a single agent" result fails to generalize beyond single-item environments. Suppose instead that there are k identical units for sale. Is the $k + 1$st-price auction (i.e., the one that sells to the k highest-valued agents at the $k + 1$st value) revenue on $n + 1$ agents at least that of the optimal k-unit auction on n agents? It is certainly not.

Consider the special case where $k = n$ and the values are distributed uniformly on $[0, 1]$. The expected revenue of the $n + 1$st-price auction on $n + 1$ agents is about one as there are n winners any the $n + 1$st value is about $1/n$ in expectation. Of course the optimal auction will offer a price of $1/2$ and achieve an expected revenue of $n/4$.

It turns out that the resource augmentation result does extend, and in a very natural way, but we will have to recruit more than a single agent. For k-unit auctions we will have to recruit k additional agents. Notice that to extend the proof of Theorem 5.1 to the k-unit setting we can define the auction B to allocate optimally to the first n agents and then any remaining items can be given to the k additional agents. The desired conclusion results. In fact, this argument can be extended to matroids. Of course matroid set systems are generally asymmetric so we have to specific what kind of agents we are adding. This is formalized by the definition and theorem below.

Definition 5.4. *A* base *of a matroid is an independent set of maximal cardinality.*

Theorem 5.5. *For any i.i.d., regular, matroid environment the expected revenue of the surplus maximization mechanism is at least that of the optimal mechanism in the environment induced by removing the agents corresponding to any base of the matroid.*

Notice that by the augmentation property of matroids, all bases are the same size. Notice that the theorem implies the aforementioned result for k-unit auctions as any set of k agents forms a base.

5.3 Single-sample Mechanisms

While the assumption that it is possible to recruit an additional agent seems not to be too severe; once we have to recruit k new agents in k-unit auctions or a new base for matroid environments, the approach provided by Bulow-Klemperer Theorem seems less relevant. In this section we will show that a single additional agent is enough to obtain a good approximation to the optimal auction revenue. We will not, however, just add this agent to the market, instead we will use this agent for statistical purposes.

In the opening of this chapter we discussed the need to connect the size of the sample for market analysis with the size of the actual market. In this context, the assumption that the prior distribution is known is tantamount to assuming that an infinitely large sample is available for market analysis. In this section we show that this impossibly large sample market can be approximated by a single sample from the distribution.

Mechanism 5.1. *The* lazy single-sample mechanism *is the following:*

1. $(\mathbf{x}', \mathbf{p}') \leftarrow \mathrm{SM}(\mathbf{v})$,
2. *draw a single sample from the distribution* $r \sim F$,
3. $x_i = \begin{cases} x'_i & \textit{if } v_i \geq r \\ 0 & \textit{otherwise, and} \end{cases}$
4. $p_i = \max(r, p'_i)$,

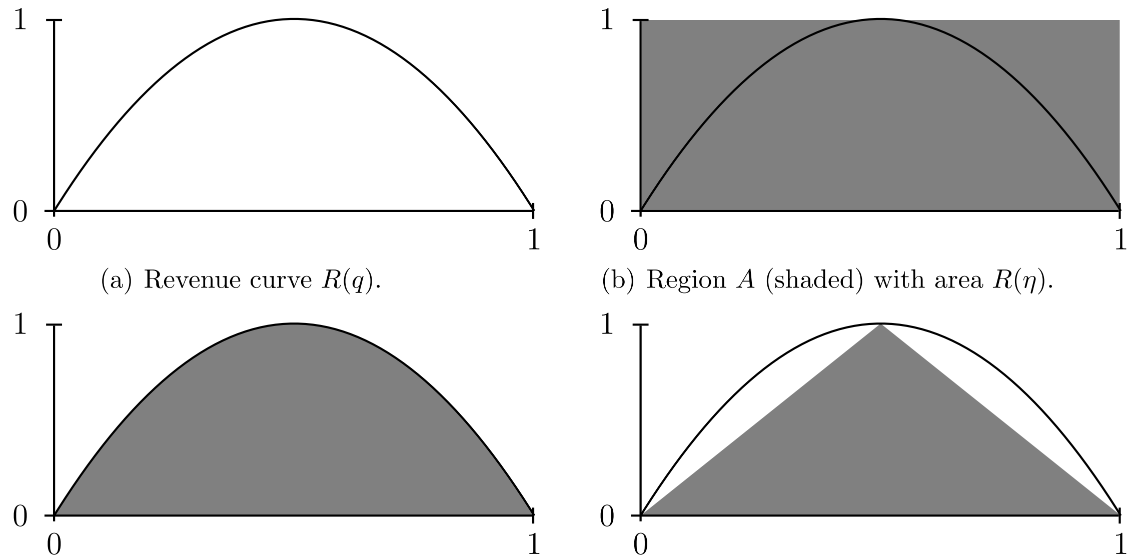
where SM *denotes the surplus maximization mechanism.*

In comparison to the surplus maximization mechanism with reserve prices discussed in Chapter 4, where the reserve prices are used filter out low-valued agents out before finding the surplus maximizing set, in the lazy single-sample mechanism the reserve price filters out low-valued agents after finding the surplus maximizing set. In matroid environments, which include single- and k-unit auctions, the order in which the reserve price is imposed is irrelevant (i.e., the same outcome results), therefore, in such environments we will refer to the lazy single-sample mechanism as the *single-sample mechanism*.

5.3.1 The Geometric Interpretation

Consider a single-agent, single-item environment. The optimal auction in such an environment is simply to post the monopoly price as a take-it-or-leave-it offer. The single-sample mechanism in this context posts a random, from the distribution, price as a take-it-or-leave-it offer. We will give a geometric proof that shows that for regular distributions, the revenue from this random price is within a factor of two of the revenue from the (optimal) monopoly price.

This result can be viewed as the $n = 1$ special case of the Bulow-Klemperer Theorem, i.e., that the two-agent second-price auction obtains at least the (one-agent) monopoly revenue.



(a) Revenue curve $R(q)$.

(b) Region A (shaded) with area $R(\eta)$.

(c) Region B (shaded) with area $\mathbf{E}[R(q)]$.

(d) Region C (shaded) with area $A/2$.

Figure 5.1: In the geometric proof of the that a random reserve is a 2-approximation to the optimal reserve, the areas of the shaded regions satisfy $A \geq B \geq C = A/2$.

In a two-agent second-price auction each agent is offered the a price equal to the value of the other, i.e., a random price from the distribution. Therefore, the two-agent second-price auction obtains twice the revenue of the single sample. The result showing that the single-sample revenue is at least half of the monopoly revenue then implies that the two-agent second-price auction obtains at least the (one-agent) monopoly revenue.

Lemma 5.6. *For a single-agent with value drawn from regular distribution F, the revenue from a random take-it-or-leave-it offer $r \sim F$ is at least half the revenue of the (optimal) monopoly offer.*

Proof. Let $R(q)$ be the revenue curve for F in quantile space. Let η be the quantile corresponding to the monopoly price, i.e., $\eta = \operatorname{argmax}_q R(q)$. The expected revenue from such a price is $R(\eta)$. Recall that drawing a random value from the distribution F is equivalent to drawing a uniform quantile $q \sim U[0, 1]$. The expected revenue from such a random price is $\mathbf{E}_q[R(q)] = \int_0^1 R(q)\, dq$. In the Figure 5.1 the area of region A is $R(\eta)$. The area of region B is $\mathbf{E}_q[R(q)]$. Of course, the area of C is less than the area of B, by concavity of $R(\cdot)$, but at least half the area of A, by geometry. The lemma follows. □

5.3.2 Random versus Monopoly Reserves

The geometric interpretation above is almost all that is necessary to show that the lazy single-sample mechanism is a good approximation to the optimal mechanism. We will show this in two pieces. First we will show that the lazy single-sample mechanism is a good approximation to the revenue of the surplus maximization mechanism with a lazy monopoly reserve. Then we argue that this lazy monopoly reserve mechanism is optimal or approximately optimal.

Theorem 5.7. *For any i.i.d., regular distribution, downward-closed environment, the revenue of the lazy single-sample mechanism is a 2-approximation to that of the lazy monopoly reserve mechanism.*

Proof. Let REF denote the lazy monopoly reserve mechanism and its revenue, and let APX denote the lazy single-sample mechanism and its revenue.

For $\mathbf{v}_{-i}$. We argue that the expected revenue from agent i in APX is at least half of that in REF. REF and APX are deterministic and dominant strategy IC, therefore in each mechanism agent i faces a critical value for winning. It will be useful to consider this critical value in quantile space, henceforth, "critical quantile". Let τ_i be the critical quantile of the surplus maximization mechanism (with no reserve). In APX, i's critical quantile is $\min(\tau_i, q)$ for $q \sim U[0, 1]$. In REF, i's critical quantile $\min(\tau_i, \eta)$, where η is the quantile of the monopoly price.

Now consider the induced revenue curve in APX from agent i with value $v_i \sim F$ as a function of q in the case where $\tau_i \leq \eta$ and where $\tau_i > \eta$ (Figure 5.2). We show, via the geometric interpretation, that in each case APX is a 2-approximation. Notice that if $q \leq \tau_i$ then APX's revenue from i is $R(q)$, otherwise it is $R(\tau_i)$. The REF revenue from i is $R(\min(\tau_i, \eta))$. By concavity of $R(\cdot)$ and geometry (Figure 5.2) the theorem follows. □

5.3.3 Single-sample versus Optimal

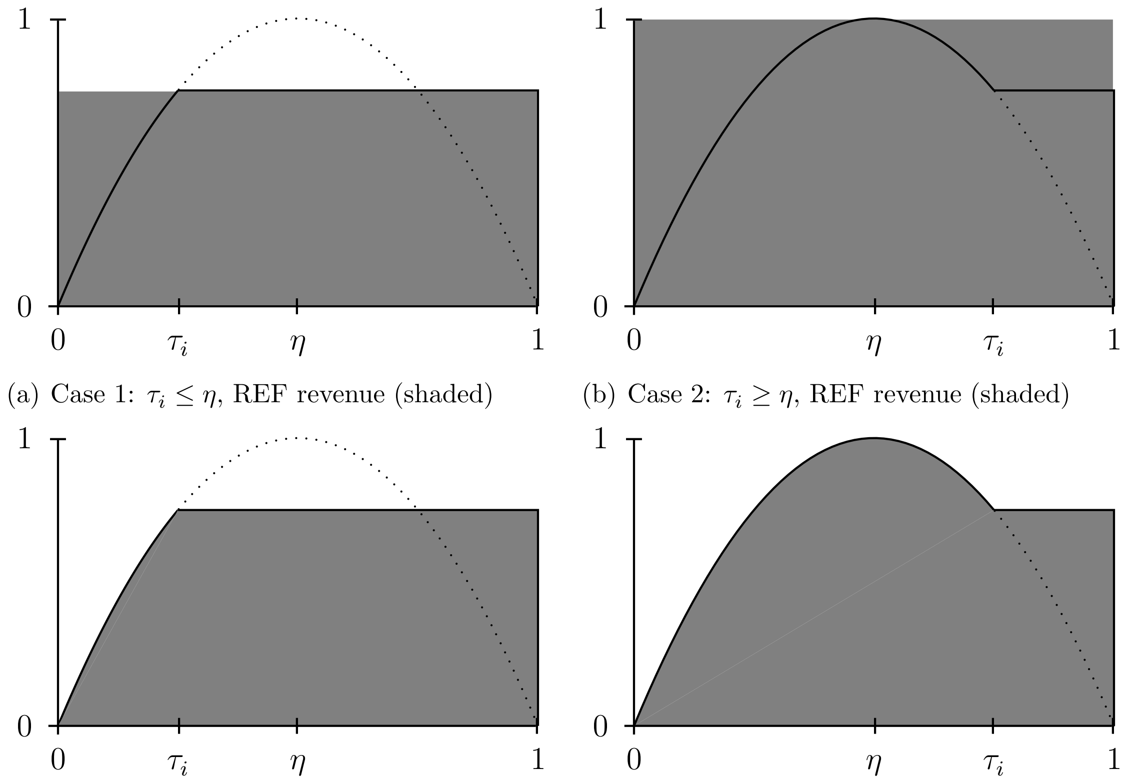
We have shown that lazy random reserve pricing is almost as good as lazy monopoly reserve pricing. We now connect lazy monopoly reserve pricing to the optimal mechanism to show that the lazy single-sample mechanism is a good approximation to the optimal mechanism.

As discussed above lazy monopoly reserve pricing is identical to (eager) monopoly reserve pricing in matroid settings. Also, Theorem 4.24 showed that monopoly reserve pricing is optimal. We conclude the following corollary. (Of course, k-unit auctions are a special case of matroid environments.)

Corollary 5.8. *For any i.i.d., regular, matroid environment, the single-sample mechanism is a 2-approximation to the optimal mechanism revenue.*

For downward-closed environments we have slightly more work to do. Theorem 4.20 showed that for monotone-hazard-rate distributions surplus maximization with (eager) monopoly reserves is a 2-approximation to the optimal mechanism. For downward-closed environments, eager and lazy reserve pricing are not identical. However, as suggested by Exercise 4.4 (for the eventual proof of Theorem 4.17), the revenue of lazy monopoly reserve pricing is an e-approximation to the optimal surplus. Clearly, then its revenue is an e-approximation to the optimal revenue. We can conclude the following.

Corollary 5.9. *For any i.i.d., monotone-hazard-rate, downward-closed environment, the lazy single-sample mechanism is a $2e$-approximation to the optimal mechanism revenue.*



(a) Case 1: $\tau_i \leq \eta$, REF revenue (shaded)

(b) Case 2: $\tau_i \geq \eta$, REF revenue (shaded)

(c) Case 1: $\tau_i \leq \eta$, APX revenue (shaded)

(d) Case 2: $\tau_i \geq \eta$, APX revenue (shaded)

Figure 5.2: On the top is a geometric depiction of the payment (shaded area) of agent i in the lazy monopoly reserve (REF) mechanism; on the bottom is the same for the lazy single sample (APX) mechanism. On the left is the case where the critical quantile τ_i is less than the monopoly quantile η; on the right is the opposite case. The revenue curve is depicted with a dotted line, and the induced revenue curve given the critical quantile is depicted with a solid line.

The bound in the above corollary can be improved to a factor of four, but we will not discuss the details here.

5.4 Prior-independent Mechanisms

We now turn to mechanisms that are completely prior-independent. Unlike the mechanisms of the preceding section, these mechanisms will not require any distributional information in advance, not even a single sample from the distribution. We will, however, still assume that there is a distribution. We search for a single mechanism that has good expected performance for any distribution from a large class of distributions.

Definition 5.10. *A mechanism* APX *is a* prior-independent β-approximation *if*

$$\forall \mathbf{F}, \quad \mathbf{E}_{\mathbf{v}\sim\mathbf{F}}[\mathrm{APX}(\mathbf{v})] \geq \tfrac{1}{\beta}\mathbf{E}_{\mathbf{v}\sim\mathbf{F}}[\mathrm{REF}_{\mathbf{F}}(\mathbf{v})]$$

where $\mathrm{REF}_{\mathbf{F}}$ *is the optimal mechanism for distribution* $\mathbf{F}$*.*

The central idea behind the design of prior-independent mechanisms is that a small amount of market analysis can be done on-the-fly as the mechanism is being run; bids of some agents can be used for market analysis for other agents.

Consider the following k-unit auction:

0. Solicit bids.
1. Randomly reject an agent i^*.
2. Run the $k+1$st-price auction with reserve v_{i^*} on $\mathbf{v}_{-i^*}$.

This auction is clearly incentive compatible. Furthermore, it is easy to see that it is a $\frac{2n}{n-1}$-approximation for n agents with values drawn i.i.d. from a regular distribution. This follows from the fact that rejecting a random agent loses at most a $1/n$ faction of the optimal revenue and the previous single-sample result (Corollary 5.8). For $n \geq 2$ this mechanism guarantees a 4-approximation. The same approach can be applied to matroid and downward-closed environments as well; however, we will focus instead on a slightly more sophisticated approach.

5.4.1 Digital Good Environments

An important single-dimensional agent environment is that of a *digital good*. A digital good is one where there is little or no cost for duplication. The cost function for digital goods is $c(\mathbf{x}) = 0$ for all $\mathbf{x}$, or equivalently, all outcomes are feasible. Digital goods are the special case of k-unit auctions where $k = n$. Therefore the mechanism above obtains a $2n/(n-1)$-approximation.

There are a number of ways to improve this mechanism to remove the $n/(n-1)$ from the approximation factor. Two of the most natural are the following.

Definition 5.11.

- *The* pairing auction *arbitrarily pairs agents and runs the second-price auction on each pair (assuming n is even).*
- *The* circuit auction *orders the agents arbitrarily (e.g., lexicographically) and offers each agent a price equal to the value of the preceding agent in the order (the first agent is offered the last agent's value).*

The random pairing auction *and the* random circuit auction *are the variants where the implicit pairing or circuit is selected randomly.*

Theorem 5.12. *For i.i.d., regular, digital-good environments, any auction wherein each agent is offered the price of another random or arbitrary (but not value dependent) agent is a 2-approximation to the optimal auction revenue.*

The proof of this theorem follows directly from the geometric interpretation for the single-sample mechanism. Clearly, the pairing and circuit auctions satisfy the conditions of the above theorem. In conclusion, it is relatively easy, within a mechanism, to get samples from the distribution.

5.4.2 General Environments

We now adapt the results for digital goods to general environments. The main idea here is to replace the lazy single-sample reserve with a lazy circuit or pairing mechanism. Notice that in downward-closed environments we can view the lazy reserve pricing used with a surplus maximizing mechanism as a digital good auction. The surplus maximizing mechanism outputs a feasible outcome. Since all subsets of feasible outcomes are feasible, the induced environment is essentially one of a digital good.

Two deterministic DSIC mechanisms, $\mathcal{M}'$ and $\mathcal{M}''$, can be composed in many ways, perhaps the most natural is the following. Consider the critical values an agent i in each mechanism, τ_i' and τ_i'', respectively. Consider the composite mechanism $\mathcal{M}$ in which i's critical value is $\tau_i = \max(\tau_i', \tau_i'')$. Notice that the set of agents served by $\mathcal{M}$ is the intersection of those served by $\mathcal{M}'$ and $\mathcal{M}''$. This outcome is feasible by downward-closure and DSIC by its definition via critical values. Notice that the surplus maximization mechanism with lazy reserves is the composition, in this manner, of the surplus maximization mechanism with the mechanism that simply makes a take-it-or-leave-it offer of the reserve price to each agent.

Instead of composing the surplus maximization mechanism with the reserve pricing, we can compose it with either the pairing or circuit auctions. Both of the theorems below follow from analyses similar to that of the single-sample mechanism.

Definition 5.13.

- *The* pairing mechanism *is the composition of the surplus maximization mechanism with the (digital goods) pairing auction.*

- *The* circuit mechanism *is the composition of the surplus maximization mechanism with the (digital goods) circuit auction.*

Theorem 5.14. *For i.i.d., regular, matroid environments, the pairing and circuit mechanisms are 2-approximations to the optimal mechanism revenue.*

Theorem 5.15. *For i.i.d., monotone-hazard-rate, downward-closed environments, the pairing and circuit mechanisms are* 4*-approximations to the optimal mechanism revenue.*

Two issues remain undiscussed. First, our prior-independent mechanisms were derived from single-sample mechanisms. Clearly more on-the-fly samples can be used to obtain revenue that more closely approximates the lazy monopoly reserve pricing, and therefore, the optimal auction. Second, similarly more on-the-fly samples can be used to obtain prior-independent approximation mechanisms when distributions may be irregular. Both of these directions will be taken up during our discussion of prior-free mechanisms in Chapter 6.

Exercises

5.1 Prove Theorem 5.2: For i.i.d., regular, single-item environments the optimal $(n-1)$-agent auction is an $\frac{n}{n-1}$-approximation to the optimal n-agent auction revenue.

5.2 Suppose we are in a non-identical environment, i.e., agent i's value is drawn from independently from distribution F_i, and suppose the mechanism can draw one sample from each agent's distribution.

(a) Give a constant approximation mechanism for regular, matroid environments (and give the constant).

(b) Give a constant approximation mechanism for monotone-hazard-rate, downward-closed environments (and give the constant).

(c) Conclude with a two Bulow-Klemperer style theorems. Suppose you can recruit a competitor for each agent (i.e., from the same distribution but where the set system only allows an agent or her competitor to be served), then the surplus maximization mechanism with competitors obtains a constant fraction of the optimal revenue in the original environment. Give one theorem for regular, matroid environments and one for monotone-hazard-rate, downward-closed environments.

5.3 This chapter has been mostly concerned with the profit objective. Suppose we wished to have a single mechanism that obtained good surplus and good profit.

(a) Show that surplus maximization with monopoly reserves is not generally a constant approximation to the optimal social surplus in regular, single-item environments.

(b) Show that the lazy single sample mechanism is a constant approximation to the optimal social surplus in i.i.d., regular, matroid environments.

(c) Investigate the Pareto fontier between prior-independent approximation of surplus and revenue. I.e., if a mechanism is an α approximation to the optimal surplus and a β-approximation to the optimal revenue, plot it as point $(1/\alpha, 1/\beta)$ in the positive quadrant.

5.4 Suppose the agents are divided into k markets where the value of agents in the same market are identically distributed. Assume that the partitioning of agents into markets is known, but not the distributions of the markets. Assume there are at least two agents in each market. Unrelated to the markets, assume the environment has a downward-closed feasibility constraint.

(a) Give a prior-independent constant approximation to the revenue-optimal mechanism for regular, matroid environments.

(b) Give a prior-independent constant approximation to the revenue-optimal mechanism for monotone-hazard-rate, downward-closed environments.

Chapter Notes

The Coase conjecture, which states that a monopolist cannot sell early at a high price to high-valued consumers and late at a low price to low-valued consumers as the late price will compete with the high price, is due to Ronald Coase (1972).

The resource augmentation result that shows that recruiting one more agent to a single-item auction raises more revenue than setting the optimal reserve price is due to Bulow and Klemperer (1996). The proof of the Bulow-Klemperer Theorem that was presented in this text is due to René Kirkegaard (2006).

The single-sample mechanism and the geometric proof of the Bulow-Klemperer theorem is due to Dhangwatnotai et al. (2010). The pairing auction for digital good environments was proposed by Goldberg et al. (2001); however, in their, potentially irregular, environment it does not have good revenue guarantees.

Chapter 6

Prior-free Mechanisms

The big challenge that separates mechanism design from (non-game-theoretic) optimization is that the incentive constraints in mechanism design bind across all valuation profiles. E.g., the payment of an agent depends on the what the mechanism does when the agent has a lower value (Theorem 2.7). Therefore, where optimization gives an outcome that is good point-wise (i.e., for any input), mechanism design gives a mechanism for all of type-space that must trade-off performance on one input for another.

In the last chapter we gave mechanisms that made this trade-off obliviously to the actual distribution. The resulting mechanisms were prior-independent and approximated the optimal mechanism for the implicit distribution. Furthermore, the described mechanisms were dominant-strategy incentive compatible, meaning, agent also need not know the distribution to act. This lack of distributional requirement for both the agent and the designer suggests that there must be a completely prior-free theory of mechanism design.

Intuitively, the class of good prior-free mechanisms should be smaller than the class of good prior-independent mechanisms. The prior-independent mechanism can rely on there being a distribution where as the prior-free mechanism cannot. Therefore, we demand from our prior-free design and analysis framework, that prior-free approximation implies prior-independent approximation. Indeed, up to constance factors, the results of this chapter subsume the results of the previous chapter.

A main challenge in considering a formal framework in which to design and analyze prior-free mechanisms is in identifying a meaningful benchmark against which to evaluate a mechanism's performance. For instance, it was natural to compare the prior-independent mechanisms of the previous chapter to the (Bayesian) optimal mechanism for the implicit, and unspecified, distribution. We define the meaningfulness of a benchmark by the implications of its approximation. A mechanism is a prior-free approximation to a given benchmark if the mechanisms performance on any valuation profile always approximates the benchmark performance. The benchmark is economically meaningful if, as desired by the previous paragraph, its approximation implies prior-independent approximation.

In this chapter we introduce the envy-free optimal revenue benchmark. An outcome, i.e., allocation and payments $(\mathbf{x}, \mathbf{p})$, is envy-free if no agent prefers to swap outcome (allocation and payment) with another agent. Notice that the envy-freedom constraint binds point-wise on valuation profiles; therefore, for any objective and valuation profile there is an envy-free outcome that is optimal. Envy-freedom can be viewed as a relaxation incentive compatibility, a view point that can be made precise in many environments, e.g., as the envy-free optimal revenue dominates the revenue of any (Bayesian) optimal mechanism. Thus, the focus of the chapter is on designing prior-free approximation mechanisms for this benchmark in general downward-closed environments.

6.1 The Digital Good Environment

Recall the digital good environment wherein all allocations are feasible. Given an i.i.d. distribution, the optimal mechanism would post the monopoly price as a take-it-or-leave-it offer to each agent. Of course, agents with values above the monopoly price would choose to purchase the item, while, agents with values below the monopoly price would not. This outcome is inherently envy-free as each agent was permitted to choose from among the two possible outcomes: either take item at the monopoly price, or take nothing and pay nothing.

Without a prior the monopoly price is not well defined; however, on inspection of the valuation profile $\mathbf{v} = (v_1, \ldots, v_n)$ it is easy to obtain an upper bound on the revenue of any monopoly pricing as $\max_i iv_{(i)}$. While it is not incentive compatible to inspect the valuation profile and offer the revenue maximizing price to each agent, it is envy free. Furthermore, though we do not argue it here, it gives the envy-free optimal revenue, denoted $\mathrm{EFO}(\mathbf{v})$. Clearly, any mechanism that approximates this envy-free optimal revenue would also approximate the (Bayesian) optimal auction for any i.i.d. distribution. Therefore, this envy-free benchmark is economically meaningful.

Unfortunately, there is no prior-free constant approximation to this benchmark. In particular, when there is $n = 1$ agent the optimal envy-free revenue is the surplus, while we know that, even if the distribution on values is known (cf., Chapter 4, Section 4.2.1), the optimal surplus and revenue can be separated by more than a constant. For instance, if the agent's value is known to fall within the range $[1, h]$ then the best approximation factor is $1 + \ln h$ (See Exercise 6.1). Clearly, if nothing is known about the range of values then no finite approximation is possible.

In fact the only thing preventing $\max iv_{(i)}$ from being a good benchmark is the case where the maximization is obtained at $i = 1$ by selling to the highest value agent at her value. We therefore slightly alter the benchmark to exclude this scenario. The *envy-free (optimal) benchmark* for digital goods is $\mathrm{EFO}^{(2)}(\mathbf{v}) = \max_{i \geq 2} iv_{(i)}$.

We now consider approximating this benchmark. In the remainder of this section we will show that deterministic auctions cannot give good prior-free approximation. We will then describe two approaches for designing prior-free auctions for digital goods. The first auction is based on a straightforward market analysis metaphor: use a random sample of the agents to estimate the distribution of values, run the optimal auction for the estimated distribution on the remaining agents. The resulting auction is known to be a 4.68-approximation. The second auction is based on a standard algorithmic design paradigm: reduction to the a "decision version" of the problem. The resulting auction is known to be a 4-approximation.

Finally, we describe a method for proving lower bounds on the approximation factor of any prior-free auction; no auction is better than a 2.42-approximation.

6.1.1 Deterministic Auctions

The main idea that enables approximation of the envy-free benchmark is that when figuring out a price to offer agent i we can use statistics from the values of all other agents $\mathbf{v}_{-i}$. This motivates the following mechanism.

Mechanism 6.1. *The* deterministic optimal price *auction offers each agent i the take-it-or-leave-it price of τ_i equal to the monopoly price for $\mathbf{v}_{-i}$.*

It is possible to show that the deterministic optimal price auction is a prior-independent constant approximation; however it is not a prior-free approximation. For example, consider the valuation profile with ten high-valued agents, with value ten, and 90 low-valued agents, with value one. What does the auction do on such a valuation profile? The offer to a high-valued agent is $\tau_h = 1$, as $\mathbf{v}_{-h}$ consists of 90 low-valued agents and 9 high-valued agents. The revenue from the high price is 90; while the revenue from the low price is 99. The offer to a low-valued agent is $\tau_1 = 10$, as $\mathbf{v}_{-1}$ consists of 89 low-valued agents and 10 high-valued agents. The revenue from the high price is 100; while the revenue from the low price is 99. Clearly with these offers all high-valued agents will win and pay one, while all low-valued agents will lose. The total revenue is ten, a far cry from the envy-free benchmark revenue of $\mathrm{EFO}^{(2)}(\mathbf{v}) = \mathrm{EFO}(\mathbf{v}) = 100$. In fact, this deficiency of the deterministic optimal price auction is one that is fundamental to all *anonymous* (a.k.a., symmetric) deterministic auctions.

Theorem 6.1. *No anonymous, deterministic digital good auction is better than an n-approximation to the envy-free benchmark.*

Proof. We consider only valuation profiles with values $v_i \in \{1, h\}$. Let $n_h(\mathbf{v})$ and $n_1(\mathbf{v})$ represent the number of h values and 1 values in $\mathbf{v}$, respectively. That an auction $\mathcal{A}$ is anonymous implies that the critical value for agent i as a function of the reports of other agents is independent of i and only a function of $n_h(\mathbf{v}_{-i})$ and $n_1(\mathbf{v}_{-i})$. Thus, we can let $\tau(n_h, n_1)$ represent the offer price of $\mathcal{A}$ for any agent i when we plug in $n_h = n_h(\mathbf{v}_{-i})$ and $n_1 = n_1(\mathbf{v}_{-i})$. Finally we assume that $\tau(n_h, n_1) \in \{1, h\}$ as this restriction cannot hurt the auction profit on the valuation profiles we are considering.

We assume for a contradiction that the auction is a good approximation and proceed in three steps.

1. Observe that for any auction that is a good approximation, it must be that for all m, $\tau(m, 0) = h$. Otherwise, on the all h's input, the auction only achieves profit n while the envy-free benchmark is hn. Thus, the auction would be at most an h-approximation which is not constant.

2. Likewise, observe that for any auction that is a good approximation, it must be that for all m, $\tau(0, m) = 1$. Otherwise, on the all 1's input, the auction achieves no profit and is clearly not an approximation of the envy-free benchmark n.

3. For the final argument, consider taking m sufficiently large and looking at $\tau(k, m-k)$. As we have argued for $k = 0$, $\tau(k, m-k) = 1$. Consider increasing k until $\tau(k, m-k) = h$. This must occur since $\tau(k, m-k) = h$ when $k = m$. Let $k^* = \min\{k : \tau(k, m-k) = h\} \geq 1$ be this transition point. Now consider an $n = m+1$ agent valuation profile with $n_h(\mathbf{v}) = k^*$ and $n_1(\mathbf{v}) = m - k^* + 1$. Consider separately the offer prices to high- and low-valued agents:

   - For low-valued agents: $\tau(n_h(\mathbf{v}_{-1}), n_1(\mathbf{v}_{-1})) = \tau(k^*, m - k^*) = h$. Thus, all low-valued agents are rejected and contribute nothing to the auction profit.
   - For high-valued agents: $\tau(n_h(\mathbf{v}_{-h}), n_1(\mathbf{v}_{-h})) = \tau(k^* - 1, m - k^* + 1) = 1$. Thus, all high-valued agents are are offered a price of one which they accept. Thus, the contribution to the auction profit from such agents is $1 \times n_h(\mathbf{v}) = k^*$.

Set $h = n$. If $k^* = 1$ then the benchmark is n (from selling to all agents at price 1); of course, for $k^* = 1$ then $n = nk^*$. If $k^* > 1$ the benchmark is also nk^* (from selling to the k^* high-valued agents at price n). Therefore, the auction profit k^* is at-best an n-approximation. □

6.1.2 Random Sampling

The conclusion from the preceding discussion is that either randomization or asymmetry is necessary to obtain prior-free approximations. While either approach will permit the design of good mechanisms, all deterministic asymmetric auctions known to date are based on derandomizations of randomized auctions. In this text we will discuss only these randomized mechanisms.

Notice that the problem with the deterministic optimal price auction is that it sometimes offers high-valued agents a low price and low-valued agents a high price. Either of these prices would have been good if only it offered consistently to all agents. The first idea to combat this lack of coordination is to coordinate using random sampling. The idea is roughly to partition the agents into a market and sample and then use the sample to estimate a good price and then offer that price to the agents in the market. With a random partition we expect a fair share of high- and low-valued agents to be in both the market and the sample; therefore, a price that is good for the sample should also be good for the market.

Mechanism 6.2. *The* random sampling (optimal price)

1. *randomly partitions the agents into S' and S'' (by flipping a fair coin for each agent),*
2. *computes (empirical) monopoly prices η' and η'' for S' and S'' respectively, and*
3. *offers η' to S'' and η'' to S'.*

As a warm-up exercise for analyzing this random sampling auction we observe that its not better than a 4-approximation to the envy-free benchmark. Consider the 2-agent input $\mathbf{v} = (1.1, 1)$ for which the envy-free benchmark is $\text{EFO}^{(2)}(\mathbf{v}) = 2$. To calculate the auction's revenue on this input, notice that these two agents are in the same partition with probability 1/2 and in different partitions with probability 1/2. In the former case, the auction's revenue is zero. In the latter case it is the lower value, i.e., one. The auction's expected profit is therefor 1/2, which is a 4-approximation to the benchmark.

Theorem 6.2. *For digital good environments and all valuation profiles, the random sampling auction is at least a 4.68-approximation to the envy-free benchmark.*

This theorem is involved and it is generally believed that the bound it provides is loose and the random sampling auction is in fact a worst-case 4-approximation. Below we will prove the weaker claim that it is at worst at 15-approximation. This weaker claim highlights the main techniques involved in proving that variants and generalizations of the random sampling auction are constant approximations.

Lemma 6.3. *For all valuation profiles, the random sampling auction is at least a 15-approximation to the envy-free benchmark.*

Proof. Assume without loss of generality that $v_{(1)} \in S'$ and call S' the market; call S'' the sample. This terminology comes from the fact that if $v_{(1)}$ is much bigger than all other agent values then all agents in S'' will be rejected; the role of S'' is then only as a sample for statistical analysis. There are two main steps in the proof. Step 1 is to show that $\text{EFO}(\mathbf{v}_{S''})$ is close to $\text{EFO}^{(2)}(\mathbf{v})$. Step 2 is to show that the revenue from price η'' on S' is close to $\text{EFO}(\mathbf{v}_{S''})$, i.e., the revenue from price η'' on S''.

We will use the following definitions. First sort the agents by value so that v_i is the ith largest valued agent. Define X_i is an indicator variable for the event that $i \in S''$ (the sample). Notice that $\mathbf{E}[X_i] = 1/2$ except for $i = 1$; $X_1 = 0$ by our assumption that the highest valued agent is in the market. Define $S_i = \sum_{j<i} X_j$. Let k be the number of winners in $\text{EFO}(\mathbf{v})$, i.e., $k = \text{argmax}_i \, i v_i$.

1. With good probability, the optimal revenue for the sample, $\text{EFO}(\mathbf{v}_{S''})$, is close to the benchmark, $\text{EFO}^{(2)}(\mathbf{v})$.

   Define the event $\mathcal{B}$ that $S_k \geq k/2$. Of course $\text{EFO}(\mathbf{v}_{S''}) \geq S_k v_k$ as the former is the optimal single price revenue on S'' and the latter is the revenue from S'' with price v_k. Event $\mathcal{B}$ implies that $S_k v_k \geq k v_k/2 = \text{EFO}^{(2)}(\mathbf{v})/2$, and thus, $\text{EFO}(\mathbf{v}_{S''}) \geq \text{EFO}^{(2)}(\mathbf{v})/2$.

   We now show that $\mathbf{Pr}[\mathcal{B}] = 1/2$ when k is even. Recall that the highest valued agent is always in the market. Therefore there are $k-1$ (an odd number) of agents which we partition between the market and the sample. One partition receives at least $k/2$ of these and half the time it is the sample; therefore, $\mathbf{Pr}[\mathcal{B}] = 1/2$. When k is odd $\mathbf{Pr}[\mathcal{B}] < 1/2$, and a slightly more complicated argument is needed to complete the proof. We omit the details.

2. With good probability, the revenue from price η'' on S' is close to $\mathrm{EFO}(\mathbf{v}_{S''})$.

   Define the event $\mathcal{E}$ that "$\forall i,\ (i - S_i) \geq S_i/3$." Notice that the left hand side of this equation is the number of agents with value at least v_i in the market, while the right hand side is a third of the number of such agents in the sample. I.e., this event implies that the partitioning of agents is not too imbalanced in favor of the sample. We refer to this event as the *balanced sample* event; though, note that it is only a one-directional balanced condition.

   Let k'' be index of the agent whose value is the monopoly price for the sample, i.e., $v_{k''} = \eta''$ and $\mathrm{EFO}(\mathbf{v}_{S''}) = S_{k''} v_{k''}$. The profit of the random sampling auction is equal to $(k'' - S_{k''}) v_{k''}$. Under the balanced sample condition this is lower bounded by $S_{k''} v_{k''}/3 = \mathrm{EFO}(\mathbf{v}_{S''})/3$.

   We defer to later the proof of a *balanced sampling lemma* (Lemma 6.4) that shows that $\mathbf{Pr}[\mathcal{E}] \geq .9$.

Finally, we combine these two pieces. If both good events $\mathcal{E}$ and $\mathcal{B}$ hold, then the expected revenue of random sampling auction is at least $\mathrm{EFO}^{(2)}(\mathbf{v})/6$. By the union bound, the probability of this good fortune is $\mathbf{Pr}[\mathcal{E} \wedge \mathcal{B}] = 1 - \mathbf{Pr}[\neg\mathcal{E}] - \mathbf{Pr}[\neg\mathcal{B}] \geq 0.4$. We conclude that the random sampling auction is a 15-approximation to the envy-free benchmark. ☐

Lemma 6.4 (Balanced Sampling)**.** *For $X_1 = 0$, X_i for $i \geq 1$ an indicator variable for a independent fair coin flipping to heads, and sum $S_i = \sum_{j \leq i} X_i$,*

$$\mathbf{Pr}[\forall i,\ (i - S_i) \geq S_i/3] \geq 0.9.$$

Proof. We relate the condition to the *probability of ruin* in a *random walk* on the integers. Notice that $(i - S_i) \geq S_i/3$ if and only if, for integers i and S_i, $3i - 4S_i + 1 > 0$. So let $Z_i = 3i - 4S_i + 1$ and view Z_i as the position, in step i, of a random walk on the integers. Since $S_1 = 0$ this random walk starts at $Z_1 = 4$. Notice that at step i in the random walk with $Z_i = k$ then at step $i + 1$ we have

$$Z_{i+1} = \begin{cases} k - 1 & \text{if } X_i = 1, \text{ and} \\ k + 3 & \text{if } X_i = 0; \end{cases}$$

i.e., the random walk either takes three steps forward or one step back. We wish to calculate the probability that this random walk never touches zero. This type of calculation is known as the *probability of ruin* in analogy to a gambler's fate when playing a game with such a payoff structure.

Let r_k denote the probability of ruin from position k. This is the probability that the random walk eventually takes k steps backwards. Clearly $r_0 = 1$ (at $k = 0$ we are already ruined) and $r_k = r_1^k$ (taking k steps back is equal to stepping back k times). By the definition of the random walk, we have the recurrence

$$r_k = \tfrac{1}{2}(r_{k-1} + r_{k+3}).$$

Plugging in the above identities,

$$r_1 = \tfrac{1}{2}(1 + r_1^4).$$

This is a quartic equation that can be solved, e.g., by *Ferarri's formula*. Since our random walk starts at $Z_1 = 4$ we calculate $r_4 = r_1^4 \leq 0.1$, meaning that the success probability for the random walk satisfying the balanced sampling condition is at least 0.9. □

6.1.3 Decision Problems

Decision problems play a central role in computational complexity and algorithm design. Where as an optimization problem is to find the optimal solution to a problem, a decision problem is to decide whether or not there exists a solution that meets a given objective criterion. While it is clear that decision problems are no harder to solve than optimization problems, often times the opposite is also true. For instance, with binary search and repeated calls to an algorithm that solves the decision problem, the optimal solution can be found. In this section we develop a similar theory for mechanism design.

Profit extraction

For profit maximization in mechanism design, recall, there is no absolutely optimal mechanism. Therefore, we define the mechanism design decision problem in terms of the aforementioned profit benchmark EFO. The decision problem for EFO and profit target R to design a mechanism that obtains profit at least R on any input $\mathbf{v}$ with $\text{EFO}(\mathbf{v}) \geq R$. We call the mechanism that solves the decision problem a *profit extractor*.

Definition 6.5. *The* digital good profit extractor *for target R and valuation profile $\mathbf{v}$ finds the largest k such that $v_{(k)} \geq R/k$, sells to the top k agents at price R/k, and rejects all other agents. If no such set exists, it rejects all agents.*

Lemma 6.6. *The digital good profit extractor is dominant strategy incentive compatible.*

Proof. Consider the following indirect mechanism. See if all agents can evenly split the target R. If some agents cannot afford to pay their fair share, reject them. Repeat with the remaining agents. Notice that as the number of agents in this process is decreasing, the fair share that each agent faces is increasing. Therefore, any agent rejected for inability to pay their fair share could not afford any of the future prices considered in the mechanism either. Thus, the incentives are identical to that of the English auction. An agent wishes to drop out when the increasing price surpasses her value.

The digital good profit extractor is obtained by applying the revelation principle to the above ascending price mechanism. □

Lemma 6.7. *For all valuation profiles $\mathbf{v}$, the digital good profit extractor for target R obtains revenue R if $R \leq \text{EFO}(\mathbf{v})$ and zero otherwise.*

Proof. EFO($\mathbf{v}$) $= kv_{(k)}$ for some k. If $R \leq$ EFO($\mathbf{v}$) then $R/k \leq v_{(k)}$. The digital good profit extractor may yet find a larger k that satisfies the same property, however, it can certainly find some k. On the other hand, if $R >$ EFO($\mathbf{v}$) $= \max_k kv_{(k)}$ then there is no such k for which $R/k \leq v_{(k)}$ and the mechanism has no winners and no revenue. □

Approximate Reduction to Decision Problem

We now use random sampling to approximately reduce the mechanism design problem of optimizing profit to the decision problem. The key observation in this reduction is an analogy. Notice that given a single agent with value v, if we offer this agent a threshold t the agent buys and pays t if and only if $v \geq t$. Analogously a profit extractor with target R obtains revenue R on $\mathbf{v}$ if and only if EFO($\mathbf{v}$) $\geq R$. The idea then is to randomly partition the agents and use profit extraction to run the second-price auction on the benchmark profit from each partition.

Definition 6.8. *The* random sampling profit extraction *auction works as follows:*

1. *Randomly partition the agents by flipping a fair coin for each agents and assigning her to S' or S''.*
2. *Calculate $R' = \text{EFO}(\mathbf{v}_{S'})$ and $R'' = \text{EFO}(\mathbf{v}_{S''})$, the benchmark profit for each part.*
3. *Profit extract R'' from S' and R' from S''.*

Notice that the intuition from the analogy to the second-price auction implies that the revenue of the random sampling profit extraction auction is exactly the minimum of R' and R''. Since the profit extractor is dominant strategy incentive compatible, so is the random sampling profit extraction auction.

Lemma 6.9. *The random sampling profit extraction auction is dominant strategy incentive compatible.*

Before we prove that the auction is a 4-approximation to to the envy-free benchmark, we give a simple proof of a lemma that will be important in the analysis.

Lemma 6.10. *Flip $k \geq 2$ fair coins, then*

$$\mathbf{E}[\min\{\#\textit{heads}, \#\textit{tails}\}] \geq \tfrac{k}{4}.$$

Proof. Let M_i be a random variable for the min{#heads, #tails} after only i coin flips. We make the following basic calculations (verify these as an exercise):

- $\mathbf{E}[M_1] = 0$.
- $\mathbf{E}[M_2] = 1/2$.
- $\mathbf{E}[M_3] = 3/4$.

We now obtain a general bound on $\mathbf{E}[M_i]$ for $i > 3$. Let $X_i = M_i - M_{i-1}$ representing the change to $\min\{\#\text{heads}, \#\text{tails}\}$ after flipping one more coin. Notice that linearity of expectation implies that $\mathbf{E}[M_k] = \sum_{i=1}^{k} \mathbf{E}[X_i]$. Thus, it would be enough to calculate $\mathbf{E}[X_i]$ for all i. We consider this in two cases:

Case 1 (i even): This implies that $i-1$ is odd, and prior to flipping the ith coin it was not the case that there was a tie, i.e, #heads $\neq$ #tails. Assume without loss of generality that #heads $<$ #tails. Now when we flip the ith coin, there is probability 1/2 that it is heads and we increase the minimum by one; otherwise, we get tails have no increase to the minimum. Thus, $\mathbf{E}[X_i] = 1/2$.

Case 2 (i odd): Here we use the crude bound that $\mathbf{E}[X_i] \geq 0$. Note that this is actually the best we can claim in worst case since $i-1$ is even and it could have been that #heads = #tails in the previous round. If this were the case then regardless of the ith coin flip, $X_i = 0$ and the minimum of #heads and #tails would be unchanged.

Case 3 ($i = 3$): This is a special case of Case 2; however we can get a better bound using the calculations of $\mathbf{E}[M_2] = 1/2$ and $\mathbf{E}[M_3] = 3/4$ above to deduce that $\mathbf{E}[X_3] = \mathbf{E}[M_3] - \mathbf{E}[M_2] = 1/4$.

Finally we are ready to calculate a lower bound on $\mathbf{E}[M_k]$.

$$
\begin{aligned}
\mathbf{E}[M_k] &= \sum\nolimits_{i=1}^{k} \mathbf{E}[X_k] \\
&\geq 0 + \tfrac{1}{2} + \tfrac{1}{4} + \tfrac{1}{2} + 0 + \tfrac{1}{2} + 0 + \tfrac{1}{2} \dots \\
&= \tfrac{1}{4} + \tfrac{\lfloor k/2 \rfloor}{2} \\
&\geq \tfrac{k}{4}.
\end{aligned}
$$

□

Theorem 6.11. *For digital good environments and all valuation profiles, the revenue of the random sampling profit extraction auction is a 4-approximation to the envy-free benchmark.*

Proof. For valuation profile $\mathbf{v}$, let REF be the envy-free benchmark and its revenue and APX be the random sampling profit extraction auction and its expected revenue. From the aforementioned analogy, the expected revenue of the auction is $\text{APX} = \mathbf{E}\big[\min(R', R'')\big]$ (where the expectation is taken over the randomized of the partitioning of agents).

Assume that envy-free benchmark sells to $k \geq 2$ agents at price p, i.e., REF $= kp$. Of the k winners in REF, let k' be the number of them that are in S' and k'' the number that are in S''. Since there are k' agents in S' at price p, then $R' \geq k'p$. Likewise, $R'' \geq k''p$.

$$
\begin{aligned}
\tfrac{\text{APX}}{\text{REF}} &= \tfrac{\mathbf{E}\left[\min(R', R'')\right]}{kp} \\
&\geq \tfrac{\mathbf{E}\left[\min(k'p, k''p)\right]}{kp} \\
&= \tfrac{\mathbf{E}\left[\min(k', k'')\right]}{k} \\
&\geq \tfrac{1}{4}.
\end{aligned}
$$

The last inequality follows from applying Lemma 6.10 when we consider $k \geq 2$ coins and heads as putting an agent in S' and a tails as putting the agent in S''.

This bound is tight as is evident from the same example from which we concluded that the random sampling optimal price auction is at best a 4-approximation. ☐

One question that should seem pertinent at this point is whether partitioning into two groups is optimal. We could alternatively partition into three parts and run a three-agent auction on the benchmark revenue of these parts . Of course, the same could be said for partitioning into k parts for any k. In fact, the optimal partitioning comes from $k = 3$, though we omit the proof and full definition of the mechanism.

Theorem 6.12. *For digital good environments and all valuation profiles, the random three-partitioning profit extraction auction is a 3.25-approximation to the envy-free benchmark.*

6.1.4 Lower bounds

We have discussed three auctions with known approximation factors 4.62, 4, and 3.25. What is the best approximation factor possible? This question, of course, turns our framework of approximation into one of optimality.

Definition 6.13. *The* prior-free optimal auction *for a envy-free benchmark* $\mathrm{EFO}^{(2)}$ *is*

$$\mathrm{argmin}_{\mathcal{A}} \max_{\mathbf{v}} \frac{\mathrm{EFO}^{(2)}(\mathbf{v})}{\mathcal{A}(\mathbf{v})}.$$

Unfortunately, this optimal auction suffers from the main drawback of optimal mechanisms. In general it is quite complicated. The auctions described heretofore can be viewed as simple approximations to this potentially complex optimal auction.

For the special case of $n = 2$, however, the prior-free optimal auction is simple. In this case, the envy-free benchmark is $\mathrm{EFO}^{(2)}(\mathbf{v}) = 2v_{(2)}$. Recall that the revenue of the second-price auction is $v_{(2)}$. Therefore, in this special case, the second-price auction is a 2-approximation to the benchmark. Is this the best possible or is there some better approximation factor possible by a more complicated auction? In fact, it is the best possible.

Lemma 6.14. *For any auction, there is a $n = 2$ agent valuation profile such that the auction is at best a 2-approximation to the envy-free benchmark.*

Proof. The proof follows a simple structure that is useful for proving lower bounds for this type of problem. First, we consider values drawn from a random distribution. Second, we argue that for any auction $\mathcal{A}$ and $\mathbf{v}$ i.i.d. from F, $\mathbf{E}_{\mathbf{v}}[\mathcal{A}(\mathbf{v})] \leq \mathbf{E}_{\mathbf{v}}\left[\mathrm{EFO}^{(2)}(\mathbf{v})\right]/2$. By the definition of expectation this implies that there exists a valuation profile $\mathbf{v}^*$ such that $\mathcal{A}(\mathbf{v}^*) \leq \mathrm{EFO}^{(2)}(\mathbf{v}^*)/2$ (as otherwise the expected values could not satisfy this condition).

We choose a distribution to make the analysis of $\mathbf{E}_{\mathbf{v}}[\mathcal{A}(\mathbf{v})]$ simple. This is important because we have to analyze it for all auctions $\mathcal{A}$. The idea is to choose the distribution for $\mathbf{v}$ such that all auctions obtain the same expected profit. The distribution that satisfies this condition is the equal-revenue distribution (Definition 4.4), i.e., $F(z) = 1 - 1/z$. Note that whatever price $\tau_i \geq 1$ that $\mathcal{A}$ offers agent i, the expected payment made by agent

i is $\tau_i \times \mathbf{Pr}[v_i \geq \tau_i] = 1$. Thus, for $n = 2$ agents the expected profit of the auction is $\mathbf{E}_\mathbf{v}[\mathcal{A}(\mathbf{v})] = n = 2$.

We must now calculate $\mathbf{E}_\mathbf{v}\left[\text{EFO}^{(2)}(\mathbf{v})\right]$. $\text{EFO}^{(2)}(\mathbf{v}) = \max_{i \geq 2} i v_{(i)}$ where $v_{(i)}$ is the ith highest valuation. In the case that $n = 2$, this simplifies to $\text{EFO}^{(2)}(\mathbf{v}) = 2v_{(2)} = 2\min(v_1, v_2)$. We recall that a non-negative random variable X has $\mathbf{E}[X] = \int_0^\infty \mathbf{Pr}[X \geq z]\, dz$ and calculate $\mathbf{Pr}\left[\text{EFO}^{(2)}(\mathbf{v}) > z\right]$.

$$\begin{aligned}
\mathbf{Pr}_\mathbf{v}\left[\text{EFO}^{(2)}(\mathbf{v}) > z\right] &= \mathbf{Pr}_\mathbf{v}[v_1 \geq z/2 \wedge v_2 \geq z/2] \\
&= \mathbf{Pr}_\mathbf{v}[v_1 \geq z/2]\, \mathbf{Pr}_\mathbf{v}[v_2 \geq z/2] \\
&= 4/z^2.
\end{aligned}$$

Note that this equation is only valid for $z \geq 2$. Of course for $z < 2$, $\mathbf{Pr}\left[\text{EFO}^{(2)}(\mathbf{v}) \geq z\right] = 1$.

$$\begin{aligned}
\mathbf{E}_\mathbf{v}\left[\text{EFO}^{(2)}(\mathbf{v})\right] &= \int_0^\infty \mathbf{Pr}\left[\text{EFO}^{(2)}\, \mathbf{v} \geq z\right] dz \\
&= 2 + \int_2^\infty \tfrac{4}{z^2} dz = 4.
\end{aligned}$$

Thus we see that for this distribution and any auction $\mathcal{A}$, $\mathbf{E}_\mathbf{v}[\mathcal{A}(\mathbf{v})] = 2$ and $\mathbf{E}_\mathbf{v}\left[\text{EFO}^{(2)}(\mathbf{v})\right] = 4$. Thus, the inequality $\mathbf{E}_\mathbf{v}[\mathcal{A}(\mathbf{v})] \leq \mathbf{E}_\mathbf{v}\left[\text{EFO}^{(2)}(\mathbf{v})\right]/2$ holds and there must exist some input $\mathbf{v}^*$ such that $\mathcal{A}(\mathbf{v}^*) \leq \text{EFO}^{(2)}(\mathbf{v}^*)/2$. □

For $n > 2$ the same proof schema gives lower bounds on the approximation factor of the prior-free optimal auction. The main difficulty of the $n > 2$ case is in calculating the expectation of the benchmark. This is complicated because it becomes the maximum of many terms. E.g., for $n = 3$ agents, $\text{EFO}^{(2)}(\mathbf{v}) = \max(2v_{(2)}, 3v_{(3)})$. Nonetheless, its expectation can be calculated exactly.

For any auction, there is a $n = 2$ agent valuation profile such that the auction is at best a 2-approximation to the envy-free benchmark.

Theorem 6.15. *For any auction, there is a valuation profile such that the auction is at best a 2.42-approximation to the envy-free benchmark. Furthermore, for special cases of $n = 2$, 3, and 4 agents the lower bound on approximation factors are exactly 2, 13/6, and 96/215, respectively.*

It is known that there is a 13/6-approximation for $n = 3$ agents. It is not known whether the 96/215 bound is tight for $n = 4$.

6.2 The Envy-free Benchmark

To generalize beyond digital good environments we must be formal about the envy-free benchmark. First, is the envy-free benchmark meaningful in multi-unit, matroid, or downward-closed environments? For instance, informally we would like prior-free approximation of the benchmark to imply prior-independent approximation for any i.i.d. prior. Second, is it analytically tractable, i.e., is there an easy to interpret description of envy-free optimal pricings? Both of these issues are important.

Recall that in the digital goods example the envy-free benchmark is the revenue from the monopoly pricing of the empirical distribution given by the valuation profile. This seems like a reasonable benchmark as the Bayesian optimal auction for digital goods is the monopoly pricing (for the real distribution). Recall that for irregular multi-unit auction environments the optimal auction is not just the second-price auction with the monopoly reserve (in particular, it may iron). For these environments the envy-free benchmark is also more complex.

Up to this point, we have assumed that the environment is given deterministically, e.g., by a cost function or set system (Chapter 3, Section 3.1). A generalization of this model would be to allow randomized environments. We view a randomized environment as a probability distribution over deterministic environments, i.e., a convex combination. For the purpose of incentives and performance, we will view mechanism design in randomized environments as follows. First, the agents report their preferences; second, the designer's cost function (or feasibility constraint) is realized; and third, the mechanism for the realized cost function is run on the reported preferences. The performance in such probabilistic environment is measured in expectation over both the randomization in the mechanism and the environment. Agents act before the set system is realized and therefore from their perspective the game they are playing in is the composition of the randomized environment with the (potentially randomized) mechanism.

An example of such a probabilistic environment comes from "display advertising." Banner advertisements on web pages are often sold by auction. Of course the number of visitors to the web page is not precisely known at the time the advertisers bid; instead, this number can be reasonably modeled as a random variable. Therefore, the environment is a convex combination of multi-unit auctions where the supply is randomized.

Definition 6.16. *Given an environment, specified by cost function $c(\cdot)$, the* permutation environment *is the convex combination of the environment with the identities of the agents permuted. I.e., for permutation π drawn uniformly at random from all permutations, the permutation environment has cost function $c(\pi(\cdot))$.*

Our goal is a prior-free analysis framework for which approximation implies prior-independent approximation in i.i.d. environments. Of course the expected revenue of the optimal auction in an i.i.d. environment is unaffected by random permutations. Therefore, with respect to our goal, it is without loss to assume a permutation environment. Importantly, while a matroid or downward-closed environment may be asymmetric, a matroid permutation or downward-closed permutation environment is inherently symmetric. This symmetry permits a meaningful study of envy-freedom.

Definition 6.17. *For valuation profile* $\mathbf{v}$*, an outcome with allocation* $\mathbf{x}$ *and payments* $\mathbf{p}$ *is envy-free if no agent prefers the outcome of another agent to her own, i.e.,*

$$\forall i, j,\ v_i x_i - p_i \geq v_i x_j - p_j.$$

The definition of envy freedom should be contrasted to the Bayes-Nash equilibrium condition given by Fact 2.6. Importantly, Bayes-Nash equilibrium constrains the outcome an agent would receive upon a unilateral "misreport" where as envy freedom constrains the outcome she would receive upon swapping with another agent. However, unlike the incentive-compatibility constraints, no-envy constraints bind point-wise on the given valuation profile; therefore, there is always a point-wise optimal envy-free outcome. The similarity of envy freedom and incentive compatibility enables virtually identical characterization and optimization of envy free outcomes (cf. Theorem 2.7).

Theorem 6.18. *For valuation profile* $\mathbf{v}$ *(sorted with* $v_1 \geq v_2 \geq \ldots \geq v_n$*), an outcome* $(\mathbf{x}, \mathbf{p})$ *is envy free if and only if*

- *(monotonicity)* $x_1 \geq x_2 \geq \ldots \geq x_n$.
- *(payment correspondence) there exists a* p_0 *and monotone function* $y(\cdot)$ *with* $y(v_i) = x_i$ *such that for all* i

  $$p_i = v_i x_i - \int_0^{v_i} y(z)\, dz + p_0,$$

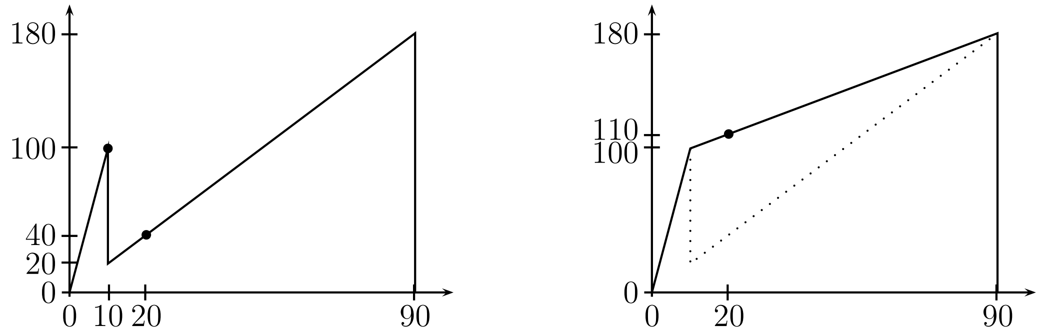
  where usually $p_0 = 0$.

Notice that the envy-free payments are not pinned down precisely by the allocation; instead, there is a range of appropriate payments. Given our objective of profit maximization, for any monotone allocation rule, we focus on the largest envy-free payments. As this payment can be interpreted as the "area above the curve $y(\cdot)$," the maximum payments are given when $y(\cdot)$ is the smallest monotone function consistent with the allocation. Formulaically this revenue can be calculated as:

$$p_i = \sum\nolimits_{j \geq i}^{n} v_j (x_j - x_{j+1}). \tag{6.1}$$

We can define the revenue curve, marginal revenue, virtual values, and their ironed equivalents that correspond to envy-free revenue (cf. Definitions 3.11, 3.14, and 3.23). In fact, these terms are exactly those that govern the Bayesian optimal revenue for the *empirical distribution.* The empirical distribution for a valuation profile is the distribution with mass i/n above value $v_{(i)}$.

For envy-free revenue, the index of an agent (in the sorted order plays the same role as quantile in the analogous definitions of Bayesian optimal mechanisms in Chapter 3 (cf.



(a) Revenue curve $R(\cdot)$ (b) Ironed revenue curve $\bar{R}(\cdot)$

Figure 6.1: The revenue curves corresponding to 10 high-valued agents and 80 low-valued agents. Depicted on $R(\cdot)$ are the revenues of the second-price auctions with reserves 10 and 2 with $k = 20$ units. Depicted on $\bar{R}(\cdot)$ is the envy-free optimal revenue with $k = 20$ units.

Definition 3.9). For these natural definitions, the optimal envy-free outcome in any symmetric environment is the ironed virtual surplus optimizer (cf. Corollary 3.27). In particular for permutation environments the desired allocation can be calculated as follows: first, calculate ironed virtual values from values; second, realize the random permutation; and third, serve the subset of agents to maximize the ironed virtual surplus.

Definition 6.19. *For the ith index, the* revenue *is* $R_i = iv_i$*; the* virtual value *is* $\phi_i = R_i - R_{i-1}$*; the* ironed revenue *is denoted by* $\bar{R}_i$ *and given by the evaluating at* i *the smallest concave function that upper bounds the point set given by* $\{0, 0\} \cup \{(i, R_i) : i \in [n]\}$*; and the* ironed virtual value *is* $\bar{\phi}_i = \bar{R}_i - \bar{R}_{i-1}$.

Theorem 6.20. *The maximal envy-free revenue for monotone allocation* $\mathbf{x}$ *is*

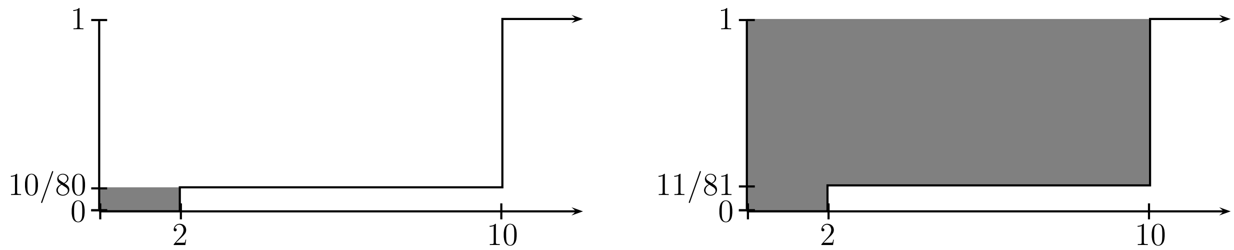
$$\sum\nolimits_i \phi_i x_i = \sum\nolimits_i R_i(x_i - x_{i+1}) \leq \sum\nolimits_i \bar{\phi}_i x_i = \sum\nolimits_i \bar{R}_i(x_i - x_{i+1})$$

with equality if and only if $\bar{R}_i \neq R_i \Rightarrow x_i = x_{i+1}$.

Theorem 6.21. *In symmetric environments, ironed virtual surplus maximization (with random tie-breaking) gives the envy-free outcome with the maximum profit.*

In a symmetric environment, ironed virtual surplus maximization gives an allocation that is monotone, i.e., $v_i > v_j \Rightarrow x_i \geq x_j$, as well as an allocation rule that is monotone, i.e., $z > z' \Rightarrow x_i(z) \geq x_i(z')$. The maximal envy-free payment of agent i for this allocation comes from equation (6.1) whereas the payment of the incentive compatible mechanism with this allocation rule comes from Corollary 2.17. These payments are related but distinct.

For example, consider a $k = 20$ unit environment and valuation profile $\mathbf{v}$ that consists of ten high-valued agents each with value ten and 80 low-valued agents each with value two. The revenue curve for this valuation profile is given in Figure 6.1(a). Both selling ten



(a) Allocation rule $x_{\text{low}}(v)$ and payment $p_{\text{low}}(2)$

(b) Allocation rule $x_{\text{high}}(v)$ and payment $p_{\text{high}}(10)$

Figure 6.2: The allocation rules for high- and low-valued agents induced by the mechanism with virtual values given in the text on the valuation profile given in the text. The payments are given by the area of the shaded region.

units at price ten and 20 units at price two are envy-free. The envy-free optimal revenue, however, is given by selling to the high-valued agents with probability one and price nine and selling to the low-valued agents with probability 1/8 and price two. It is easy to verify that this outcome is envy-free and that its total revenue is 110. The ironed revenue curve for this valuation profile is given in Figure 6.1(b). The ironed virtual values are given by the following function:

$$\bar{\phi}(v) = \begin{cases} -\infty & v < 2, \\ 1 & v \in [2, 10), \text{ and} \\ 10 & v \in [10, \infty). \end{cases}$$

We now calculate the revenue of the incentive compatible mechanism that serves the 20 agents with the highest ironed virtual value. In the virtual-surplus-maximizing auction, on the valuation profile $\mathbf{v}$ (with ten high-valued agents and 80 low-valued agents), the high-valued agents win with probability one and the low-valued agents win with probability 1/8 (as there are ten remaining units to be allocated randomly among 80 low-valued agents). To calculate payments we must calculate the allocation rule for both high- and low-valued agents. Low-valued agents, by misreporting a high value, win with probability one. The allocation rule for low-valued agents is depicted in Figure 6.2(a). High-valued agents, by misreporting a low value, on the other hand, win with probability 11/81. Such a misreport leaves only nine high-value-reporting agents and so there are 11 remaining units to allocate randomly to the 81 low-value-reporting agents. The allocation rule for high-valued agents is given in Figure 6.2(b). Payments can be read from the allocation rules: a high-valued agent pays about 8.9 and a low-valued agent (in expectation) pays 1/4. The total revenue from ten of each is about 109. Notice that this revenue is only slightly below the envy-free optimal revenue.

The revenue calculation above was complicated by the fact that when a high-valued agent reports truthfully there are ten remaining units to allocate to the 80 low-valued agents; whereas when misreporting a low value, there are 11 remaining units to allocate to 81 low-value reporting agents. Importantly: the allocation rule for high-valued agents and low-valued agents are not the same (compare Figures 6.2(a) and 6.2(b)). The envy-free optimal revenue can be viewed as an approximation of the incentive-compatible revenue that is more analytically tractable.

We now formalize the fact that the envy-free revenue is an economically meaningful benchmark. The theorem below implies that, in matroid permutation environments, prior-free approximation of the benchmark implies prior-independent approximation.

Theorem 6.22. *For any matroid permutation environment and any ironed virtual value function $\bar{\phi}(\cdot)$, ironed virtual surplus maximzation's envy-free revenue is at least its incentive-compatible revenue.*

Proof. We show that the envy-free payment of agent i is at least her incentive-compatible payment. In particular if we let $x_i(\mathbf{v})$ be the allocation rule of the ironed virtual surplus optimizer in the permutation environment, then for $z \leq v_i$, $x_i(\mathbf{v}_{-i}, z)$ (as a function of z) is at most the smallest $y(z)$ that satisfies the conditions of Theorem 6.18. Since the incentive-compatible and envy-free payments, respectively, correspond to the area "above the curve" this inequality implies the desired payment inequality.

Since $x_i(\mathbf{v}_{-i}, z)$ is monotone, we only evaluate it at $v_j \leq v_i$ and show that $x_i(\mathbf{v}_{-i}, v_j) \geq x_j(\mathbf{v})$. This can be seen by the following sequence of inequalities.

$$\begin{aligned} x_i(\mathbf{v}_{-i}, v_j) &= x_j(\mathbf{v}_{-i}, v_j) \\ &\geq x_j(\mathbf{v}). \end{aligned}$$

The equality above comes from the symmetry of the environment and the fact that agent i and j have the same value in profile $(\mathbf{v}_{-i}, v_j)$. The inequality comes from the matroid assumption and the fact that the greedy algorithm is optimal (Theorem 4.22): when agent i reduces her bid to v_j, agent j is less likely to be blocked by i. ☐

We are now ready to formally define the envy-free benchmark. Notice that the envy-free benchmark is well defined in all environments not just symmetric environments. For instance, when we wish to compare a mechanisms performance to the envy-free benchmark, it is not necessary for the environment to be symmetric.

Definition 6.23. *Given any environment, let* $\mathrm{EFO}(\mathbf{v})$ *denote the maximum revenue attained by an envy-free outcome in the corresponding permutation environment.*

Definition 6.24. *Let* $\mathbf{v}^{(2)} = (v_{(2)}, v_{(2)}, \ldots, v_{(n)})$ *be the valuation profile with* $v_{(1)}$ *replaced with a duplicate of* $v_{(2)}$ *and define* $\mathrm{EFO}^{(2)}(\mathbf{v}) = \mathrm{EFO}(\mathbf{v}^{(2)})$.

6.3 Multi-unit Environments

We will discuss two approaches for multi-unit environments. In the first, we will give an approximate reduction to digital good environments. This reduction will lose a factor of two in the approximation ratio, i.e., it will derive a 2β-approximation for multi-unit environments from any β-approximation for digital goods. The second approach will be to directly generalize the random sampling optimal price auction to multi-unit environments. This generalization randomly partitions the agents into two part, calculates ironed virtual valuation functions for the empirical distribution of each part, and then runs optimal $k/2$-unit auction on each part using the ironed virtual valuation function from the opposite part.

Our first approach is an approximate reduction. For i.i.d., irregular, single-item environments Corollary 4.12 shows that the second-price auction with anonymous reserve is a 2-approximation to the optimal auction. I.e., the loss in performance from not ironing when the distribution is irregular is at most a factor of two. In fact, this result extends to multi-unit environments (as does the prophet inequality from which it is proved) and the approximation factor only improves. Given the close connection between envy-free optimal outcomes and Bayesian optimal auctions, it should be unsurprising that this result translates between the two models.

Consider the revenue of the surplus maximization mechanism with the best (ex post) anonymous reserve price. For instance, for the k-unit environment and valuation profile $\mathbf{v}$, this revenue is $\max_{i \leq k} i v_{(i)}$. It is impossible to approximate this revenue with a prior-free mechanism so, as we did for the envy-free benchmark, we exclude the case that it sells to only the highest-valued agent at her value. Therefore, for k-unit environments the *anonymous-reserve benchmark* is $\max_{2 \leq i \leq k} i v_{(i)}$. Notice that for digital goods, i.e., $k = n$, the anonymous-reserve benchmark is equal to the envy-free benchmark. Of course, an anonymous reserve is envy free so the envy-free benchmark is at least the anonymous-reserve benchmark.

We now give an approximate reduction from multi-unit environments to digital-good environments in two steps. We first show that the envy-free benchmark is at most twice the anonymous-reserve benchmark in multi-unit environments. We then show an approximation preserving reduction from multi-unit to digital-good environments with respect to the anonymous-reserve benchmark.

Theorem 6.25. *For any valuation profile, in multi-unit environments, the envy-free benchmark is at most twice the anonymous-reserve benchmark.*

Proof. Assume without loss of generality that the envy-free optimal revenue is derived from selling all k units. In terms of revenue curves (Definition 6.19), the envy-free optimal revenue for $\mathbf{v}$ is $\text{REF} = \max_{i \leq k} \bar{R}_i$ whereas the anonymous-reserve revenue is $\text{APX} = \max_{i \leq k} R_i$.

Assume without loss of generality that the envy-free optimal revenue sells all k units and irons between index $i < k$ and $j > k$ as depicted in Figure 6.3. We will use a short hand notation and refer to the value of a point as the value of its y-coordinate. Accordingly, $C = \text{REF} = \bar{R}_k$, $A = R_i = i v_{(i)}$, $E = R_j = j v_{(j)}$, and $D = \frac{k}{j} R_j = k v_{(j)}$. Note that $v_{(k)} \geq v_{(j)}$ so $R_k \geq D$.

By definition the anonymous-reserve revenue satisfies $\text{APX} = \max_i R_i$ so $A \leq \text{APX}$ and $D \leq \text{APX}$ so $A + D \leq 2\,\text{APX}$. But, line segment AB is certainly longer than line segment CD so $\text{REF} = C \leq A + D \leq 2\,\text{APX}$.

Finally, this inequality holds for any $\mathbf{v}$ therefore it also holds for $\mathbf{v}^{(2)}$. ∎

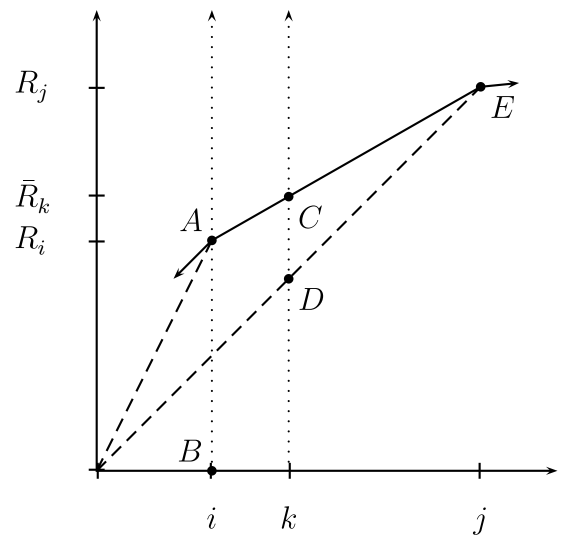


Figure 6.3: Depiction of ironed revenue curve $\bar{\mathbf{R}}$ for the pictorial proof of Theorem 6.25. The solid piece-wise linear curve is $\bar{\mathbf{R}}$, the convex hull of $\mathbf{R}$, and contains the line-segment connecting point $A = (i, R_i)$ and point $E = (j, R_j)$. The envy-free benchmark is achieved at point $D = (k, \bar{R}_k)$. The dashed lines have slope $v_{(i)}$ and $v_{(j)}$.

Theorem 6.25 reduces the problem of approximating the envy-free benchmark to that of approximating the anonymous-reserve benchmark. There is a general construction for converting a digital good auction $\mathcal{A}$ into a limited supply auction and if $\mathcal{A}$ is a β-approximation to the anonymous-reserve benchmark (which is identical to the envy-free benchmark for digital goods) then so is the resulting multi-unit auction.

Mechanism 6.3. *The k-unit variant $\mathcal{A}_k$ of digital good auction $\mathcal{A}$ is the following:*

1. *Simulate the $k+1$st-price auction (i.e., the k highest valued agents win and pay $v_{(k+1)}$).*
2. *Simulate $\mathcal{A}$ on the k winners $v_{(1)}, \ldots, v_{(k)}$.*
3. *Serve the winners from the second simulation and charge them the higher of their prices in the two simulations.*

Implicit in this definition is a new notion of mechanism composition (cf. Chapter 5, Section 5.4.2). It is easy to see that this mechanism composition is dominant strategy incentive compatible. In general such a composition is DSIC whenever no winner of the first mechanism can manipulate her value to change the set of winners while simultaneously remaining a winner (Exercise 6.3). The proof of the following theorem is immediate.

Theorem 6.26. *If $\mathcal{A}$ is a β-approximation in digital good environments then its multi-unit variant $\mathcal{A}_k$ is a 2β-approximation in multi-unit environments (with respect to the envy-free benchmark).*

We can of course apply this theorem to any digital good auction; for instance, from Theorem 6.12 we can conclude the following corollary.

Corollary 6.27. *There is an multi-unit auction that is a 6.5-approximation to the envy-free benchmark.*

An alternative approach to the multi-unit auction problem is to directly generalize the random sampling optimal price auction. Intuitively, the random sampling auction partitions the agents into two parts and then derives the optimal auction for each part and runs that auction on the opposite part. For digital goods the optimal auction for each part is just the to post the monopoly price for the valuation profile. Of course, multi-unit environments the optimal auction, e.g., for irregular distributions, may iron.

Mechanism 6.4. *The* random sampling (ironed virtual surplus maximization) *auction for the k-unit environment*

1. *randomly partitions the agents into S' and S'' (by flipping a fair coin for each agent),*
2. *computes ironed virtual valuation functions $\bar{\phi}'$ and $\bar{\phi}''$ for the empirical distributions of S' and S'' respectively, and*
3. *maximizes ironed virtual surplus on S'' with respect to $\bar{\phi}'$ and S' with respect to $\bar{\phi}''$ with $k/2$-units each.*

If k is odd the last unit is allocated with probability 1/2 to each part.

The proof of the following theorem can be derived similarly to the proof of Lemma 6.3; we omit the details.

Theorem 6.28. *For multi-unit environments and all valuation profiles, the random sampling auction is a constant approximation to the envy-free benchmark.*

The random sampling auction shares some good properties with optimal mechanisms. The first is that the mechanism on each part is an ironed-virtual-surplus optimization. I.e., in each part it sorts the agents by ironed virtual surplus and allocates to the agents greedily in that order. This property is useful for two reasons. First, in environments where the supply k of units is unknown in advance, the mechanism can be implemented *incrementally*. Each unit of supply is allocated to alternating partitions to the agent remaining with the highest ironed virtual valuation. Second, as we will see in the next section, it can be applied without specialization to matroid permutation and position environments.

6.4 Matroid Permutation and Position Environments

Position environments are important as they model auctions for selling advertisements on Internet search engines such as Google, Yahoo!, and Bing. In these auctions agents bid for positions with higher positions being better. The feasibility constraint imposed by position auctions is a priori symmetric.

Definition 6.29. *A* position environment *is one with n agents, m positions, each position j described by weight w_j. An auction assigns each position j to an agent i which corresponds to setting $x_i = w_j$. Positions are usually assumed to be ordered in non-increasing order, i.e., $w_j \geq w_{j+1}$. (Often w_1 is normalized to one.)*

Position auctions correspond to advertising on Internet search engines as follows. Upon each search to the search engine, *organic search results* appear on the left hand side and *sponsored search results*, a.k.a., advertisements, appear on the right hand side of the search results page. Advertiser i receives a revenue of v_i in expectation each time their ad is clicked (e.g., if the searcher buys the advertisers product) and if their ad is shown in position j it receives click-through rate w_j, i.e., the probability that the searcher clicks on the ad is w_j. If the ad is not clicked on the advertiser receives no revenue. Searchers are more likely to click on the top slots than the bottom slots, hence $w_j \geq w_{j+1}$. An advertiser i shown in slot j receives value $v_i w_j$. Though this model of Internet advertising leaves out many details of the environment, it captures many others.

We now show that mechanism design for matroid permutation environments can be reduced to position auctions which can be reduced to k-unit auctions. The main intuition that underlies this reduction is provided by the following definition.

Definition 6.30. *The* characteristic weights $\mathbf{w}$ *for a matroid are defined as follows: Set $v_i = n - i + 1$, for all i, and consider the surplus maximizing allocation when agents are assigned roles in the set system via random permutation and then the maximum feasible set is calculated, e.g., via the greedy algorithm. Let w_i be the probability of serving agent i, i.e., by definition, the ith highest-valued agent.*

To see why the characteristic weights are important, notice that since the greedy algorithm is optimal for matroids, the cardinal values of the agents do not matter, just the sorted order. Therefore, e.g., when maximizing ironed virtual value, w_i is the probability of serving the agent with the ith highest ironed virtual value.

Theorem 6.31. *The problem of revenue maximization (or approximation) in matroid permutation environments reduces to the problem of revenue maximization (or approximation) in position environments.*

Proof. We show two things. First, we show that for any matroid permutation environment with characteristic weights $\mathbf{w}$, the position environment with weights $\mathbf{w}$ has the same optimal expected revenue. Second, for any such environments any position auction can be converted into an matroid permutation auction auction that achieves the same approximation factor to the optimal mechanism. These two results imply that any Bayesian, prior-independent, or prior-free approximation results for position auctions extend to matroid permutation environments.

1. Revenue optimal auctions are ironed virtual surplus optimizers. Let $\mathbf{w}$ be the characteristic weights for the given matroid environment. By the definition of $\mathbf{w}$, the optimal auctions for both the matroid permutation and position environments serve the agent

with the jth highest positive ironed virtual value with probability w_j. (In both environments agents with negative ironed virtual values are discarded.) Expected revenue equals expected virtual surplus; therefore, the optimal expected revenues in the two environments are the same.

2. Consider the following matroid permutation mechanism which is based on the position auction with weights $\mathbf{w}$. The input is $\mathbf{v}$. First, simulate the position auction and let $\mathbf{j}$ be the assignment where j_i is the position assigned to agent i, or $j_i = \perp$ if i is not assigned a slot. Reject all agents i with $j_i = \perp$. Now run the greedy matroid algorithm in the matroid permutation environment on input $v'_i = n - j_i + 1$ and output its outcome.

   Notice that any agent i is allocated in the matroid permutation setting with probability equal to the expected weight of the position it is assigned in the position auction. Therefore the two mechanisms have the exact same allocation rule (and therefore, the exact same expected revenue). □

We are now going to reduce position auctions to single-item multi-unit auctions. This reduction implies that the approximation factor of a given multi-unit auction in an i.i.d. distributions can immediately be extended to matroid permutation and position environments. Furthermore, the mechanism that gives this approximation can be derived from the multi-unit auction.

Theorem 6.32. *The problem of revenue maximization (or approximation) in position auctions reduces to the problem of revenue maximization (or approximation) in k-unit auctions.*

Proof. This proof follows the same high-level argument as the proof of Theorem 6.31.

Let $d_j = w_j - w_{j+1}$ be the difference between successive weights. Recall that without loss of generality $w_1 = 1$ so $\mathbf{d}$ gives a probability measure over $[m]$.

1. The expected revenue of an optimal position auction is equal to the expected revenue of the convex combination of optimal j-unit auctions under measure $\mathbf{d}$. In the optimal position auction and the optimal auction for the above convex combination of multi-unit auctions the agent with the jth highest positive ironed virtual value is served with probability w_j. (In both settings agents with negative ironed virtual values are discarded.) Therefore, the expected revenues in the two environments are the same.

2. Now consider the following position auction which is based on a multi-unit auction. Simulate a j-unit auction on the input $\mathbf{v}$ for each $j \in [m]$ and let $x_i^{(j)}$ be the (potentially random) indicator for whether agent i is allocated in simulation j. Let $x_i = \sum_j x_i^{(j)} d_j$ be the expected allocation to j in the convex combination of multi-unit auctions given by measure $\mathbf{d}$. Reindex $\mathbf{x}$ in non-increasing order. Then $\mathbf{w}$ majorizes $\mathbf{x}$ in the sense that $\sum_i^k w_i \geq \sum_i^k x_i$ (and with equality for $k = m$). Therefore we can write $\mathbf{x} = S\mathbf{w}$ where S is a doubly stochastic matrix. Any doubly stochastic matrix is a convex combination of permutation matrices, so we can write $S = \sum_\ell \rho_\ell P_\ell$ where $\sum_\ell \rho_\ell = 1$ and each P_ℓ is

a permutation matrix (Birkhoff–von Neumann Theorem). Finally, we pick an ℓ with probability ρ_ℓ and assign the agents to positions in the permutation specified by P_ℓ. The resulting allocation is exactly the desired $\mathbf{x}$.

Let β be the worst case, over number of units k, approximation factor of the multi-unit auction in the Bayesian, prior-independent, or prior-free sense. The position auction constructed is at worst a β-approximation in the same sense. □

We conclude that matroid permutation auctions reduce to position auctions which reduce to multi-unit auctions. But multi-unit environments are the simplest of matroid permutation environments, i.e., the uniform matroid, where even the fact that the agents are permuted is irrelevant because uniform matroids are inherently symmetric. Therefore, from the perspective of optimization and approximation all of these problems are equivalent.

It is important to note, however, that this reduction may not preserve non-objective aspects of the mechanism. For instance, we have discussed that anonymous reserve pricing is a 2-approximation to ironed virtual surplus maximization in multi-unit environments (e.g., Corollary 4.12 and Theorem 6.25). The reduction from matroid permutation and position environments does not imply that surplus maximization with an anonymous reserve gives a 2-approximation in these more general environments. This is because in the multi-unit 2-approximation via an anonymous reserve, the reserve is tailored to k, the number of units. Therefore, constructing a position auction or matroid mechanism would require simulating the multi-unit auction with various supply constraints and reserve prices; the resulting mechanism would not be an anonymous reserve mechanism.

In fact, for i.i.d., irregular, position and matroid permutation environments the surplus maximization mechanism with anonymous reserve is not generally a constant approximation to the optimal mechanism. The approximation factor via the anonymous reserve in these environments is $\Omega(\log n / \log \log n)$, i.e., there exists matroid permutation and position environments, and distribution such that the anonymous reserve mechanism has expected revenue that is a $\Theta(\log n / \log \log n)$ multiplicative factor from the optimal mechanism revenue. We leave this result as an exercise with the hint that the distribution that gives this result is a generalization of the Sydney opera house distribution (Definition 4.5). The same inapproximation result holds with comparison between the anonymous-reserve and envy-free benchmarks.

Theorem 6.33. *There exists an i.i.d. distribution (resp. valuation profile), a matroid permutation environment, and position environment such that the (optimal) anonymous reserve mechanism (resp. benchmark) a* $\Theta(\log n / \log \log n)$*-approximation the Bayesian optimal mechanism (resp. envy-free benchmark).*

Implicit in the above discussion (and reductions) is the assumption that the characteristic weights for a matroid permutation setting can be calculated, or fundamentally, that the weights in the position auction are precisely known. Notice that in our application of position auctions to advertising on Internet search engines the position weights were the likelihood of a click for an advertisement in each position. These weights can be estimated but are not known exactly. The general reduction from matroid permutation and position auctions to multi-unit auctions requires foreknowledge of these weights.

Recall from the discussion of the multi-unit random sampling auction (Mechanism 6.4) that, as an ironed virtual surplus maximizer, it does not require foreknowledge of the supply k of units. Closer inspection of the reductions of Theorem 6.32 reveals that if the given multi-unit auction is an ironed virtual surplus maximizer then the weights do not need to be known to calculate the appropriate allocation. Simply maximize the ironed virtual surplus.

In the definition of permutation environments, it is assumed that the agents are unaware of their roles in the set system, i.e., the agents' incentives are taken in expectation over the random permutation. A mechanism that is incentive compatible in this permutation model may not generally be incentive compatible if agents do know their roles. Therefore, matroid permutation auctions that result from the above reductions are not generally incentive compatible without the permutation. Of course the random sampling auction is a ironed virtual surplus maximizer and ironed virtual surplus maximizers are dominant strategy incentive compatible (Theorem 3.25).

Corollary 6.34. *For any matroid environment and valuation profile, the random sampling auction is dominant strategy incentive compatible and when the values are randomly permuted, its expected revenue is a β-approximation to the envy-free benchmark where β is its approximation factor for multi-unit environments.*

6.5 Downward-closed Permutation Environments

In multi-unit, position, and matroid permutation environments, ironed virtual surplus maximization is ordinal, i.e., it depends on the relative order of the ironed virtual values and not their magnitudes. In contrast, the main difficulty of downward-closed environments is that ironed virtual surplus maximization is not ordinal. Nonetheless, for downward-closed environments variants of the random sampling (ironed virtual surplus maximization) and the random sampling profit extraction auctions give constant approximations to the envy-free benchmark. We will describe only the latter result.

Our approach to profit extraction in general downward-closed environments will be the following. The true (and unknown) valuation profile is $\mathbf{v}$. Suppose we knew a profile $\mathbf{v}'$ that was a coordinate-wise lower bound on $\mathbf{v}$, i.e., $v_{(i)} \geq v'_{(i)}$ for all i (short-hand notation: $\mathbf{v} \geq \mathbf{v}'$). A natural goal with this side-knowledge would be to obtain the optimal envy-free revenue for $\mathbf{v}'$. A mechanism that obtains this revenue (in expectation over the random permutation) whenever the coordinate-wise lower-bound assumption holds is a profit extractor.

Mechanism 6.5. *The* downward-closed profit extractor *for $\mathbf{v}'$ is the following:*

1. *Reject all agents if there exists an i with $v_{(i)} < v'_{(i)}$.*
2. *Calculate the ironed virtual values $\bar{\boldsymbol{\phi}}'$ for $\mathbf{v}'$.*
3. *For all i, assign the ith highest-valued agent the ith highest ironed virtual value $\bar{\phi}'_{(i)}$.*

4. Serve the agents to maximize the ironed virtual surplus.

Theorem 6.35. *For any downward-closed environment and valuation profiles* $\mathbf{v}$ *and* $\mathbf{v}'$*, the downward-closed profit extractor for* $\mathbf{v}'$ *is dominant strategy incentive compatible and if* $\mathbf{v} \geq \mathbf{v}'$ *then its expected revenue under a random permutation is at least the envy-free optimal revenue for* $\mathbf{v}'$*.*

Proof. See Exercise 6.5. □

To make use of this profit extractor we need to calculate a $\mathbf{v}'$ that satisfies the assumption of the theorem that is non-manipulable. The idea is to use biased random sampling. In particular, if we partition the agents into a sample with probability $p < 1/2$ and market with probability $1 - p$, then there is a high probability the valuation profile for the sample is a coordinate-wise lower bound on that for the sample. Furthermore, conditioned on this event, the expected optimal envy-free revenue of the sample approximates the envy-free benchmark.

Mechanism 6.6. *The biased (random) sampling profit extraction mechanism for downward-closed environments (with parameter* $p < 1/2$*) is:*

1. *Randomly partition the agents into* S *(with probability* p*) and* M *(with probability* $1-p$*).*

2. *Reject agents in* S*.*

3. *Run the downward-closed profit extractor for* $\mathbf{v}_S$ *on* M*.*

The main lemma that enables the proof that this biased sampling profit extraction mechanism performs well is very similar to Lemma 6.4.

Lemma 6.36. *For* $X_1 = 0$*,* X_i *for* $i \geq 1$ *an indicator variable for a independent biased coin flipping to heads with probability* $p < 1/2$ *(tails otherwise), and sum* $S_i = \sum_{j \leq i} X_i$*,*

$$\mathbf{Pr}[\forall i,\ S_i \leq (i - S_i)] = 1 - \left(\tfrac{p}{1-p}\right)^2.$$

Proof. See Exercise 6.6. □

Theorem 6.37. *For any downward-closed environment and any valuation profile, the biased sampling profit extraction auction with* $p \approx .21$ *is dominant strategy incentive compatible and its expected revenue under a random permutation is a 18.2-approximation to the envy-free benchmark.*

Proof. We define the event $\mathcal{B}$ that $\mathbf{v}_M \geq \mathbf{v}_S$ and the event $\mathcal{C}$ that the highest-valued agent (a.k.a., agent 1) is in the market. Lemma 6.36 implies that $\mathbf{E}[\mathcal{B} \mid \mathcal{C}] = 1 - p^2/(1-p)^2$. Of course, $\mathbf{Pr}[\mathcal{C}] = 1 - p$.

The expected revenue of the biased sampling profit extraction mechanism is, by the definition of conditional expectation,

$$\begin{aligned}\mathbf{E}[\mathrm{EFO}(\mathbf{v}_S) \mid \mathcal{C} \wedge \mathcal{B}]\,\mathbf{Pr}[\mathcal{C} \wedge \mathcal{B}] &= \mathbf{E}[\mathrm{EFO}(\mathbf{v}_S) \mid \mathcal{C}]\,\mathbf{Pr}[\mathcal{C}] \\ &\quad - \mathbf{E}[\mathrm{EFO}(\mathbf{v}_S) \mid \mathcal{C} \wedge \neg\mathcal{B}]\,\mathbf{Pr}[\mathcal{C} \wedge \neg\mathcal{B}]\,.\end{aligned}$$

We now bound the terms on the right hand side in terms of $\mathrm{EFO}(\mathbf{v}_{-1})$, the envy-free optimal revenue on the valuation profile without the highest-valued agent. For the first term,

$$\begin{aligned}\mathbf{E}[\mathrm{EFO}(\mathbf{v}_S) \mid \mathcal{C}]\,\mathbf{Pr}[\mathcal{C}] &\geq p\,\mathrm{EFO}(\mathbf{v}_{-1})\mathbf{Pr}[\mathcal{C}] \\ &= p(1-p)\,\mathrm{EFO}(\mathbf{v}_{-1}).\end{aligned}$$

To see the inequality: Event $\mathcal{C}$ means that agent 1 is in M, the remaining valuation profile is $\mathbf{v}_{-1}$. Envy-free revenue is super-additive so the expectation of the envy-free optimal revenue is super-linear. For the second term,

$$\begin{aligned}\mathbf{E}[\mathrm{EFO}(\mathbf{v}_S) \mid \mathcal{C} \wedge \neg\mathcal{B}]\,\mathbf{Pr}[\mathcal{C} \wedge \neg\mathcal{B}] &\leq \mathbf{E}[\mathrm{EFO}(\mathbf{v}_{-1})]\,\mathbf{Pr}[\neg\mathcal{B} \mid \mathcal{C}]\,\mathbf{Pr}[\mathcal{C}] \\ &= \tfrac{p^2}{1-p}\mathbf{E}[\mathrm{EFO}(\mathbf{v}_{-1})]\,.\end{aligned}$$

The above inequality follows from the coarse upper bound that $\mathrm{EFO}(\mathbf{v}_S) \leq \mathrm{EFO}(\mathbf{v}_{-1})$ under event $\mathcal{C}$. Combining the bounds above we get:

$$\mathbf{E}[\mathrm{EFO}(\mathbf{v}_S) \mid \mathcal{C} \wedge \mathcal{B}]\,\mathbf{Pr}[\mathcal{C} \wedge \mathcal{B}] \geq \left(p(1-p) - \tfrac{p^2}{1-p}\right)\mathrm{EFO}(\mathbf{v}_{-1}).$$

Optimizing for p and using the inequality that $\mathrm{EFO}(\mathbf{v}_{-1}) \geq \mathrm{EFO}^{(2)}(\mathbf{v})/2$ (Exercise 6.7) we get the desired bound in the theorem. □

Exercises

6.1 Consider the following single-agent prior-free pricing game. There is a value $v \in [1, h]$. If you offer a price $p \leq v$ you get p otherwise you get zero.

(a) Design a randomized pricing strategy to minimize the ratio of the value to the revenue.

(b) Prove that your randomized pricing strategy is optimal. Hint: Use the lower-bounding technique for digital-goods auctions from class.

(c) Discuss the connection between your above results and the claim from class that it is impossible for a digital-goods auction to approximate the envy-free benchmark $\mathrm{EFO}(\mathbf{v}) = \max_i iv_{(i)}$.

6.2 Consider the design of prior-free incentive-compatible mechanisms with revenue that approximates the (optimal) social-surplus benchmark, i.e., $\mathrm{OPT}(\mathbf{v})$, when all values are known to be in a bounded interval $[1, h]$. For downward-closed environments, give a $\Theta(\log h)$-approximation mechanism.

6.3 Consider a generalization of the mechanism composition from the construction of the multi-unit variant of a digital good auction, i.e., where the $k+1$st-price auction and the given digital good auction are composed (Mechanism 6.3). Two dominant strategy incentive compatible mechanisms A and B can be composed as follows: Simulate mechanism A; run mechanism B on the winners of mechanism A; and charge the winners of B the maximum of their critical values for A and B.

(a) Show that the composite mechanism is dominant strategy incentive compatible if the set of winners of A is non-manipulable in the following sense. There are no two values for an agent i such that the sets of winners in A are distinct but contain i.

(b) Show that the set of winners in the surplus maximization mechanism in matroid environments is non-manipulable.

6.4 Prove the envy-free variant of Theorem 6.33, i.e., that there exists a valuation profile and a position environment for which the anonymous reserve benchmark is a $\Omega(\log n / \log \log n)$-approximation to the envy-free benchmark.

6.5 Show that for any downward-closed environment and valuation profiles $\mathbf{v}$ and $\mathbf{v}'$, the downward-closed profit extractor for $\mathbf{v}'$ is dominant strategy incentive compatible and if $\mathbf{v} \geq \mathbf{v}'$ then its expected revenue under random permutation is at least the envy-free optimal revenue for $\mathbf{v}'$. I.e., prove Theorem 6.35.

6.6 Prove Lemma 6.36: For $X_1 = 0$, X_i for $i \geq 1$ an indicator variable for a independent biased coin flipping to heads with probability $p < 1/2$ (tails otherwise), and sum $S_i = \sum_{j \leq i} X_i$,

$$\mathbf{Pr}[\forall i,\ S_i \leq (i - S_i)] = 1 - \left(\tfrac{p}{1-p}\right)^2.$$

6.7 Given a valuation profile $\mathbf{v}$ in sorted order, i.e., $v_1 \geq v_2 \geq \cdots \geq v_n$, and any (single-dimensional) downward-closed permutation environment, show that the envy-free optimal revenue for $\mathbf{v}^{(2)} = (v_2, v_2, \ldots, v_n)$ and $\mathbf{v}_{-1} = (v_2, v_3, \ldots, v_n, 0)$ are within a factor of two of each other.

Chapter Notes

The prior-free auctions for digital good environments were first studied by Goldberg et al. (2001) where the deterministic impossibility theorem and the random sampling optimal price auction were given. The proof that the random sampling auction is a prior-free 15-approximation is from Feige et al. (2005); the bound was improved to 4.68 by Alaei et al. (2009). Profit extraction and the random sampling profit extraction mechanism were given by Fiat et al. (2002). The extension of this auction to three partitions can be found in Hartline and McGrew (2005). The downward-closed profit extractor is from Ha and Hartline (2011).

The lower-bound on the approximation factor of prior-free auctions for digital goods of 2.42 was given by Goldberg et al. (2004). It is conjectured that this lower bound is tight for general n-agent environment; however, optimal prior-free auctions have not been identified for $n \geq 4$. The second-price auction is optimal for $n = 2$ and its approximation ratio is $\beta = 2$. The optimal three-agent auction can be found in Hartline and McGrew (2005), its approximation ratio is $\beta = 13/6 \approx 2.17$.

This chapter omitted a very useful technique for designing prior-free mechanisms using a "consensus mechanism" on statistically robust characteristics of the input. In this vein the consensus estimates profit extraction mechanism from Goldberg and Hartline (2003) obtains a 3.39-approximation for digital goods. This approach is also central in obtaining a tractable asymmetric deterministic auction that gives a good approximation (Aggarwal et al., 2005). Ha and Hartline (2011) extend the consensus approach to downward-closed permutation environments.

This chapter omitted asymptotic analysis of the random sampling auction which is given Balcan et al. (2008). This analysis allows agents to be distinguished by publicly observable attributes and agents with distinct attributes may receive distinct prices.

The formal prior-free design and analysis framework for digital good auctions was given by Fiat et al. (2002). This framework was refined for general symmetric auction problems and grounded in the theory of Bayesian optimal auctions by Hartline and Roughgarden (2008). The connection between prior-free mechanism design and envy-freedom was given by Hartline and Yan (2011).

Analysis of the random sampling auction for limited supply (i.e., k-unit auctions) was given by Devanur and Hartline (2009). This result implies prior-free approximation results for matroid permutation and position environments. This result is enabled by the equivalence between position auctions and convex combinations of k-unit auctions (for each k) that is described by Dughmi et al. (2009) and an equivalence between matroid permutation and position environments by Hartline and Yan (2011). Generalizations that give prior-free auctions for downward-closed permutation environments are given by Hartline and Yan (2011).

Chapter 7

Multi-dimensional Approximation

Throughout the majority of this text we have assumed that the agents' private preferences are given by a single value for receiving an abstract service, i.e., that agents' types are single dimensional. We now turn to multi-dimensional environments where the agents' preferences are given by a multi-dimensional type. E.g., a home buyer may have distinct values for different houses on the market; an Internet user may have distinct values for various qualities of service; an advertiser on an Internet search engine may value traffic for search phrase "mortgage" higher than that for "loan", etc.

One of the most important example environments for multi-dimensional mechanism design is that of *combinatorial auctions*. In combinatorial auctions each agent has a valuation function that is defined across all bundles. I.e., if agent i receives bundle $S \subset \{1, ..., m\}$ then she has value $v_i(S)$. A combinatorial auction assigns to agent i bundle S_i and payment p_i. For such an outcome, agent i's utility is given by $v_i(S_i) - p_i$, i.e., it is *quasi-linear*.

For the objective of social surplus, the single-dimensional-agent surplus maximization mechanism (Mechanism 3.1) generalizes and is optimal. In this generalization, agents report their multi-dimensional preferences, the mechanism chooses the outcome that maximizes social surplus for the reported preferences, and it charges each agent the externality imposed on the remaining agents. The proof of the following theorem follows in a similar fashion to that of Theorem 3.7 and Corollary 3.8.

Theorem 7.1. *For agents with (generally multi-dimensional) quasi-linear preferences, the surplus maximization mechanism is dominant strategy incentive compatible and maximizes the social surplus.*

Even though the surplus maximization mechanism is optimal (for social surplus), it is sometimes infeasible to run. For instance, in many environments posted-pricing mechanisms are used in place of auction-like mechanisms. We will show that posted-pricing mechanisms can approximate the optimal social surplus in some relevant environments, though not for general combinatorial auctions.

For the objective of profit, there are no general descriptions of optimal mechanisms for environments where agents have multi-dimensional preferences. Essentially, mechanisms for multi-dimensional environments are complex and optimizing over them does not yield concise or intuitive descriptions, nor does it yield practical mechanisms. In this section we will explore approximation for the objective of profit maximization. In particular, we will show that both surplus maximization with reserve prices and posted-pricing mechanisms can approximate the optimal mechanism. Furthermore, the prices in these mechanisms that perform well can be easily calculated and interpreted.

We will use as a running example in this chapter the environment of *matching markets*. In a matching market there are n agents and m items (e.g., houses). Each agent i has a value v_{ij} for house j. The agents are unit-demand, i.e., each wants at most one house, and the houses are unit-supply, i.e., each can be sold to at most one agents. Agent values are drawn independently at random, e.g., with $v_{ij} \sim F_{ij}$.

7.1 Item Pricing

We start with the special case of the matching markets where there is only one agent, i.e., $n = 1$. In this environment an important optimization problem is to identify revenue-optimal pricings. I.e., a pricing $\mathbf{p} = (p_1, \ldots, p_m)$ such that when the agent buys the item that generates the highest positive utility, i.e., the j that maximizes $v_j - p_j$, the revenue of the seller is maximized.

Unfortunately, there is no concise economic understanding of optimal pricings and their revenue. Therefore, in pursuit of goal approximately optimal pricings, the first hurdle is in finding concise understanding of an upper bound on the revenue of an optimal pricing. Then, if a pricing approximates this upper bound, it also approximates the optimal pricing.

The main idea in obtaining an upper bound is from the thought experiment where we imagine that instead of one agent with unit-demand preferences over the m items that we have m (single-dimensional) agents who each want their specific item, but with the constraint that at most one can be served. In this latter environment the optimal selling mechanism would be the optimal single-item auction derived in Chapters 3. Notice that while, in the pricing problem, the seller can only post a price on each item, in the auction problem, competition between agents can drive the price up. Therefore, intuition suggests that the revenue in the (single-dimensional) auction environment may be an upper bound on the revenue in the (multi-dimensional) pricing environment. This is indeed the case.

Theorem 7.2. *For any product distribution $\mathbf{F} = F_1 \times \cdots \times F_m$, the expected revenue of the optimal single-agent, m-item pricing when the agent's values for the items are drawn from $\mathbf{F}$ is at most that of the optimal single-item, m-agent auction when the agents' values for the item are drawn from $\mathbf{F}$.*

Proof. Any item pricing $\mathbf{p}$ can be converted into a single-item auction $\mathcal{A}_{\mathbf{p}}$ such that the expected revenue from the item pricing is at most that of the auction. For convenience define $v_0 = p_0 = 0$. The auction $\mathcal{A}_{\mathbf{p}}$ assigns the item to the agent j that maximizes $v_j - p_j$. For any fixed values of the other agents, $\mathbf{v}_{-j}$, this allocation rule is monotone in agent j's value and therefore ex post incentive compatible. It is also deterministic, so by Corollary 2.18

there is a critical value τ_j for agent j which is the infimum of values for which the agent wins the auction; the agent pays exactly this critical value on winning. Of course $\tau_j \geq p_j$.

Now notice that the allocation rule of the auction $\mathcal{A}_{\mathbf{p}}$ is identical to the allocation rule of the pricing $\mathbf{p}$. For the pricing the agent chooses the item that maximizes $v_j - p_j$; for the auction the winner is selected to maximize $v_j - p_j$. Furthermore, the revenue for the pricing is exactly the p_j that corresponds to this j whereas in the auction it is τ_j which, as discussed, is at least p_j. Therefore, the auction $\mathcal{A}_{\mathbf{p}}$ obtains at least revenue of the pricing $\mathbf{p}$.

Therefore, the optimal auction obtains at least the revenue of the optimal pricing. □

With the upper bound from optimal single-item auctions in hand, our goal of approximating the optimal pricing can be refined to approximating this optimal single-item auction revenue. In fact, the desired approximation result is an immediate consequence of Theorem 4.10 for single-item auctions, i.e., that for any $\mathbf{F}$ a sequential posted pricing with constant ironed virtual prices is a 2-approximation to the optimal single-item auction revenue. Of course, the revenue of our single-agent, m-item environment is no worst than that of a single-item, m-agent sequential posted pricing (because the sequential posted pricing revenue is, by definition, from the worst possible ordering of the agents).

Corollary 7.3. *For any independent, unit-demand, single-agent environment, a pricing with uniform ironed virtual prices is a 2-approximation to the optimal pricing revenue.*

For single-agent environments item pricings are equivalent to deterministic mechanisms. This equivalence follows from a multi-dimensional variant of Corollary 2.18 which is generally known as the *taxation principle*. Therefore, an approximation to the optimal pricing revenue is equivalently an approximation of the optimal deterministic mechanism. (We defer discussion of approximation of randomized mechanisms to Section 7.3.)

7.2 Reduction: Unit-demand to Single-dimensional Preferences

It should be noted that the construction in the preceding section can be viewed as a reduction from multi-dimensional unit-demand preferences to single-dimensional preferences. We can conclude that from the perspective of approximation, the multi-dimensional unit-demand preferences are similar enough to single-dimensional preferences that a good approach to unit-demand environments is to simulate the outcome of the corresponding single-dimensional environment. We now make that connection and the reduction precise. (Crucial to this connection is the independence of the agents' values.)

Formally, consider the following *general unit-demand environment*. There are n agents and m services each agent i has value v_{ij} for service j. An outcome is an assignment of agents to services (perhaps with some agents left unassigned). We will denote this assignment by the indicator $\mathbf{x}$ with $x_{ij} = 1$ if i receives service j and 0 otherwise. There is an arbitrary feasibility constraint over such assignments which we denote, as before, with a cost function $c(\cdot)$ which is zero or infinity for feasibility problems. We assume, without loss of generality, the implicit feasibility constraint that each agent can only receive one service, i.e., $\mathbf{x}$ such that $x_{ij} = x_{i'j} = 1$ for $i \neq i'$ have $c(\mathbf{x}) = \infty$.

A unit-demand environment is thus specified by the distribution $\mathbf{F}$ indexed by agent-service pairs and the cost function $c(\cdot)$ over outcomes $\mathbf{x}$, also indexed by agent-item pairs. In all of the results described herein, the agents will be independently distributed; in most of the results the items will also be independently distributed.

7.2.1 Single-dimensional Analogy

As in the pricing environment we can define the single-dimensional analog to any general unit-demand environment. In this analog, each unit-demand agent is replaced with a single-dimensional representative for eached desired service. Notice that in the single-dimensional analog the implicit feasibility constraint that a unit-demand agent can receive at most one service is translated to the constraint that only one of its representatives can be served at once.

Definition 7.4. *The* representative environment *for the n agent, m service unit-demand environment given by $\mathbf{F}$ and $c(\cdot)$ is the single-dimensional environment given by $\mathbf{F}$ and $c(\cdot)$ with nm single-dimensional agents indexed by coordinates ij.*

7.2.2 Upper bound

The restriction that only one representative of each unit-demand agent can be served at once induces competition between representatives. Intuitively this competition should result in an increased revenue in the optimal mechanism for the representative environment over the original unit-demand environment. Were this the whole story, the optimal revenue in the representative environment would be an upper bound on the optimal revenue in the original environment. In fact it is almost the whole story. The optimal mechanism for the representative environment (which is deterministic) is an upper bound on the optimal deterministic mechanism for the original (unit-demand) environment.

Detailed discussion of randomized mechanisms for multi-dimensional environments are deferred to Section 7.3, where we will see that, while a randomized mechanism for the unit-demand environment can obtain more revenue than the optimal mechanism for the representative environment, it is only by a constant factor more, e.g., a factor of two for single-agent environments. Therefore, a constant times the revenue of the optimal mechanism for the representative environment is an upper bound on the optimal (randomized) unit-demand mechanism. Such a bound is sufficient for obtaining constant approximations via the reduction described here.

Theorem 7.5. *For any independent, unit-demand environment, the optimal deterministic mechanism's revenue is at most that of the optimal mechanism for the single-dimensional representative environment.*

Proof. The proof of this theorem is similar to that of Theorem 7.2. See Exercise 7.2. □

	Item 1	Item 2
Agent 1	$v_{11} = 4$	$v_{12} = 5$
Agent 2	$v_{21} = 10$	$v_{22} = 8$

(a) agent values

	Item 1	Item 2
Agent 1	$p_{11} = \mathbf{2}$	$p_{12} = 4$
Agent 2	$(p_{21} = 5)$	$p_{22} = \mathbf{4}$

(b) agent specific item prices

Figure 7.1: The tables above depict agent values and posted prices in a two-agent two-item matching environment. When agent 1 arrives before agent 2, then agent 1 buys 1, agent 2 buys 2, and the revenue is 6 (purchace prices depicted in boldface). If the agents arrive in the opposite order a higher revenue is obtained.

7.2.3 Reduction

The goal of this section it to reduce the problem of designing a mechanism that approximates the optimal unit-demand mechanism to a single-dimensional-agent approximation problem. Following the techniques developed in Chapter 4 it may then be possible to instantiate the reduction by solving the single-dimensional-agent approximation problem.

For the unit-demand single-agent item-pricing example of Section 7.1, Corollary 7.3, which states that item pricing can approximate the Bayesian optimal auction in the single-agent unit-demand environment, follows from Theorem 4.10, which states that sequential posted pricings, i.e., where the agents arrive in any order, approximate the optimal multi-agent single-item auction. To see why this is, compare the tie-breaking rules in these two environments. In the unit-demand pricing problem the item is allocated that maximizes $v_j - p_j$. In the sequential posted pricing problem ties are broken in worst-case order, i.e., to maximize $-p_i$. Clearly, the expected revenue from multi-dimensional pricing is no worse than that of the single-dimensional pricing.

Extend the definition of sequential posted pricings to unit-demand environments with multiple agents (i.e., to generalize item prices). A sequential posted pricing is given by prices $\mathbf{p}$ with p_{ij} the price offered to agent i for service j. After the valuations are realized, the agents arrive in sequence and take their utility maximizing service that is still feasible, given the actions of preceding agents in the sequence. The revenue of such a process clearly depends on the sequence and we pessimisticly assume the worst-case. See Figure 7.1 for an example.

Definition 7.6. *A* sequential posted pricing *is an pricing of services (specialized) for each agent with the semantics that agents arrive in any order and take their favorite service that remains feasible. The revenue of such a pricing is given by the worst-case ordering.*

Consider the sequential posted pricing problem in both the original unit-demand environment and the representative single-dimensional environment. Suppose you had the choice of being the seller in one of these two environments, given the same distribution and costs, which environment would you choose? I.e., which environment gives a higher expected revenue? Whereas when considering auction problems, you would prefer the representative environment because of the increased competition, for sequential posted pricings there is no benefit from competition. In fact, the seller in the representative environment is at a disadvantage because the agents are in a worst case order and there are more possible orderings of the agents in the nm-agent representative environment than the n-agent original environment.

Theorem 7.7. *The expected revenue of a sequential posted pricing for unit-demand environments is at least the expected revenue of the same pricing in the representative single-dimensional environment.*

Proof. Compare sequential posted pricings for unit-demand environments (i.e., with n unit-demand agents) with sequential posted pricings for their representative environments (i.e., with nm single-dimensional agents). The difference between these two environments with respect to sequential posted pricings is that in the representative environment the nm agents can arrive in any order whereas in the original environment the an agent arrives and considers the prices on services ordered by utility. Thus, the set of orders in which the nm prices are considered in the representative environment contains the set of orders in the original environment. For worst-case sequences, then, the representative environment is worse. □

Combining this lower bound with the upper bound from Theorem 7.5 we have our reduction: approximation of the optimal mechanism by multi-dimensional sequential posted pricing reduces to that of single-dimensional sequential posted pricing.

Corollary 7.8. *If a sequential posted pricing is approximately optimal in the representative (single-dimensional) environment it is approximately optimal in the original (unit-demand) environment.*

7.2.4 Instantiation

It remains to instantiate the reduction from sequential posted pricing approximation in unit-demand environments to single-dimensional environments. I.e., we need to show that there are good sequential posted pricing mechanisms for single-dimensional environments. Here we will give such an instantiation for independent, regular, matching markets, i.e., where the services are items, and each item has only one unit of supply.

The representative environment for matching markets is one where there are nm agents and agent ij with value $v_{ij} \sim F_{ij}$ desires item j. For any original agent i and all j at most one representative ij can win. For any item j and all i at most one representative ij can win. The virtual surplus maximization mechanism, denoted VSM, is optimal for this single-dimensional environment.

Let q_{ij}^{VSM} be the probability that VSM serves representative ij. Let $p_{ij}^{\text{VSM}} = F_{ij}^{-1}(1-q_{ij}^{\text{VSM}})$ be the corresponding price at which, if posted to representative ij, would be accepted with probability q_{ij}^{VSM}. Now consider the pricing $p_{ij} = F_{ij}^{-1}(1 - q_{ij})$ for $q_{ij} = q_{ij}^{\text{VSM}}/2$. These probabilities and prices can be calculated, for instance, by simulating the optimal mechanism.

Definition 7.9. *For representative matching market environments, the* simulation prices, $\mathbf{p}$*, satisfy* $p_{ij} = F_{ij}^{-1}(1 - \frac{1}{2}\mathbf{Pr}[\textit{the optimal mechanism serves } ij])$ *for all* i *and* j*.*

We claim that sequential posted pricing with the simulation prices give an 8-approximation to the optimal mechanism's revenue. The theorem is proven in two steps, the first gives an upper bound on the revenue of the optimal mechanism in terms of the above prices and probability, the second gives a lower bound on the sequential pricing revenue in terms of the same. As will be evident from the proof, this bound is not tight; improving the bound is left for Exercise 7.3.

Theorem 7.10. *For regular distributions in the representative matching market environment, the sequential posted pricing with the simulation prices* $\mathbf{p}$ *is an 8-approximation to the revenue of the optimal mechanism.*

Lemma 7.11. *For regular distributions in the representative matching market environment, the expected revenue of the optimal mechanism,* VSM*, is at most* $\sum_{ij} p_{ij}^{\text{VSM}} q_{ij}^{\text{VSM}}$*.*

Proof. The proof of this lemma follows from a standard approach. Consider an "unconstrained" mechanism that allocates to each representative ij with probability at most q_{ij}^{VSM} but is not constrained by the original feasibility constraints, i.e., that only one representative ij of each agent i is served and that each item j is only allocated to at most one representative ij. In such an unconstrained environment the representatives do not interact at all. Furthermore, by regularity and the fact that the original p_{ij}^{VSM} are at least the monopoly price, the optimal unconstrained mechanism simply posts price p_{ij}^{VSM} to each representative ij. Its expected revenue is $\sum_{ij} p_{ij}^{\text{VSM}} q_{ij}^{\text{VSM}}$. Finally, VSM, the optimal mechanism for the constrained environment, is a valid solution to the unconstrained environment, therefore the optimal unconstrained mechanism revenue gives an upper bound on its revenue. □

Lemma 7.12. *For regular distributions in the representative matching market environment, the expected revenue from the sequential posted pricing of the simulation prices is at least* $\frac{1}{8} \sum_{ij} p_{ij}^{\text{VSM}} q_{ij}^{\text{VSM}}$*.*

Proof. If the sequential posted pricing is able to make an offer to agent ij then its expected revenue is $q_{ij} p_{ij} \geq q_{ij}^{\text{VSM}} p_{ij}^{\text{VSM}}/2$. This inequality follows because the $q_{ij} = q_{ij}^{\text{VSM}}/2$ and $p_{ij} \geq p_{ij}^{\text{VSM}}$ (since prices only increase with a lower selling probability). We now show that the probability that the sequential posted pricing is able to make the offer to representative ij is at least 1/4. As a consequence the expected revenue from representative ij is $q_{ij}^{\text{VSM}} p_{ij}^{\text{VSM}}/8$; and summing over all representatives ij gives the lemma.

To show that the probability that it is feasible to offer service to representative ij is at least 1/4, consider the worst-case ordering for this probability, i.e., the ordering where representative ij is last. Representative ij can be served if for all $j' \neq j$ representatives $j'i$ are not served, and for all $i' \neq i$ representatives $i'j$ are not served. The first event certainly happens if $v_{i'j} < p_{i'j}$ for all $i' \neq i$ and the second if $v_{ij'} < p_{ij'}$ for all $j' \neq j$. We now show that each of these events happens with probability at least 1/2; since the events are independent the probability that both occur is at least 1/4.

Consider the event that all $v_{i'j} < p_{i'j}$ for all $i' \neq i$ (the probability of the other event can be analized with the same approach). With respect to this event the possibility that $v_{i'j} \geq p_{i'j}$ is a bad event that happens with probability $q_{i'j}$. The probability of any of these bad events occuring can be bounded using the union bound by $\sum_{i'} q_{i'j}$. Of course, $\sum_{i'} q_{i'j}^{\text{VSM}} \leq 1$ since the optimal mechanism allocates to one of these $i'j$ representatives with probability at most one (by the feasibility constraint) so $\sum_{i'} q_{i'j} \leq 1/2$. Therefore, probability that none of the bad events happen is at least 1/2. □

This instantiation of the reduction above covers matching markets with regular distributions. Similar instantiations can be applied to generalizations that include irregular distributions and environments with feasibility constraints induced by matroids. Sequential posted pricings do not give good approximations in general downward-closed environments.

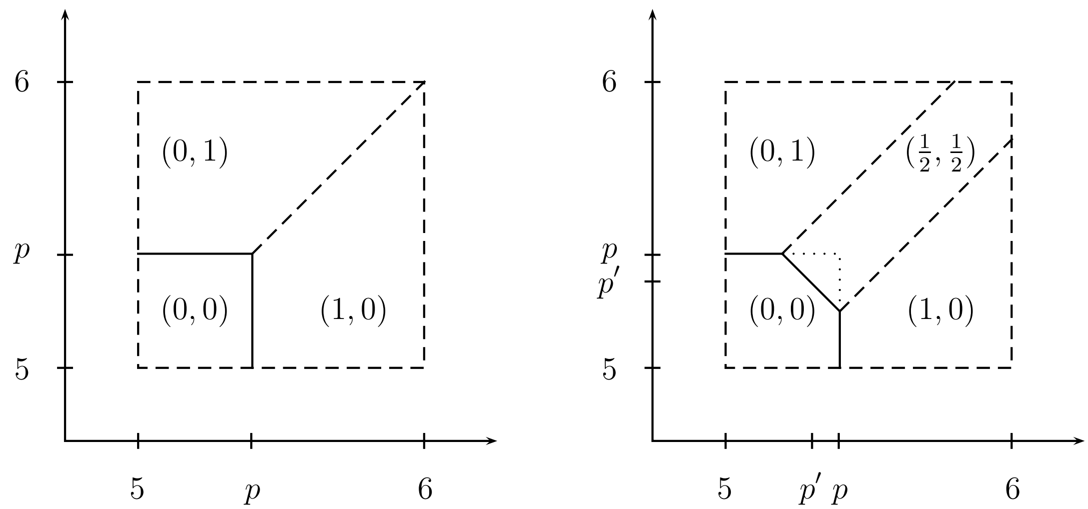
7.3 Lottery Pricing and Randomized Mechanisms

Thus far in this chapter we have showed that there are pricing mechanisms that approximate the optimal deterministic mechanism in multi-dimensional unit-demand environments. These results are a little unsatisfying because we would really like a mechanism that approximates the optimal, potentially randomized, mechanism. Even in the simple single-agent environments described previously in this chapter, the optimal mechanism may not be a deterministic pricing of items. Instead, it might price randomized outcomes, a.k.a., lotteries.

This distinction raises a sharp contrast with (Bayesian) single-dimensional environments where there is always an optimal mechanism that is deterministic. For instance, with a lexicographical tie-breaking rule, the ironed virtual surplus maximization mechanism has a deterministic allocation rule.

Consider the single-agent unit-demand problem of designing a mechanism to maximize the revenue of the seller. Deterministic mechanisms are equivalent to item pricings whereas randomized mechanisms are equivalent to *lottery pricings*. A lottery is a probability distribution over outcomes. For instance, for the $m = 2$ item case, a lottery could assign either item 1 or item 2 with probability 1/2 each. Lotteries do not have to be uniform, i.e., they can be biased in favor of some items, and they do not have to be complete, i.e., there may be some probability of assigning no item. A lottery pricing is then a set of lotteries and prices for each. For such a lottery pricing, the agent then chooses the lottery and price that give her highest utility for her given valuations for the items.

The following example shows that lottery pricings can give higher revenue than item pricings. There are two items (and one agent). The agent's value for each item is distributed independently and uniformly from the interval $[5, 6]$. The optimal item pricing for this environment to set a uniform price of 5.097 for each item. I.e., the agent is offered the option to buy item 1 at price 5.097 or to buy item 2 at price 5.097. The agent then buys the item that she values most as long as her value for that item is at most 5.097. Such an allocation rule is depicted in Figure 7.2(a) with $p = 5.097$. Now consider adding the additional option of buying at price 5.057 a lottery that realizes to item 1 or item 2 each



(a) Item pricing.

(b) Lottery pricing.

Figure 7.2: Depicted are the allocation regions for item pricing (p, p) and lottery pricing $\{((0, 1), p), ((1, 0), p), (\frac{1}{2}, \frac{1}{2}), p')\}$. The pricing and lotteries divide the valuation space into regions based on the preferred outcome of the agent. The diagonal line that gives the lower left boundary of the region where the lottery is preferred is the solution to the equation $v_1 + v_2 = p'$.

with probability 1/2. Now if the agent is nearly indifferent between the two items then she will buy the lottery and pay the lower price. Without the lottery option if the agent had average value bigger than 5.057 but no individual value over 5.097, the agent would buy nothing. Therefore, by adding this lottery option revenue is lost for some valuations of the agent and gained for others. One can calculate these losses and gains to conclude that the lottery pricing increases the expected revenue. Figure 7.2(b) with $p' = 5.057$ depicts the allocation rule that additionally offers the lottery option.

We would like to have a theory for approximating the optimal (possibly randomized) mechanism in multi-dimensional environments. Again, a crucial step in this endeavor is in identifying an analytically tractable upper bound on the optimal mechanism. Recall for the single-agent environment that the single-dimensional representative environment gave such an upper bound on optimal deterministic mechanisms (Theorem 7.2). The intuition for this bound was that the increased competition over the representative environment allowed the optimal mechanism for it to obtain more revenue than that of the original unit-demand environment. This intuition turns out to not be entirely correct when randomized mechanisms are allowed. In particular, there are examples where the optimal lottery pricing obtains more revenue than the optimal single-item auction for the representative environment.

To get some intuition for the failure of single-item auctions to provide an upper bound for lottery pricings consider the $m = 2$ item environment and the fair lottery which assigns item 1 or 2 each with probability 1/2. What is the value that the agent has for this lottery? It is the average of the agent's values. Averages of independent random variables concentrate around their expectations, therefore, the agent's value for this lottery has less randomness (in particular, a lower standard deviation) than her value for either of the individual items. A lottery pricing mechanism can take advantage of this sort of concentration. (In fact, later in this chapter we will apply this same intuition to environments with additive valuations, i.e., the value for a bundle of items is the sum of independently distributed values for each item in the bundle.)

We now show that the advantage that a lottery pricing has over a single-item auction in the representative environment is at most a factor of two.

Theorem 7.13. *For any product distribution $\mathbf{F} = F_1 \times \cdots \times F_m$, the expected revenue of the optimal single-agent, lottery pricing when the agent's values for the items are drawn from $\mathbf{F}$ is at most twice that of the optimal single-item, m-agent auction with the agents' values for the item drawn from $\mathbf{F}$.*

Proof. Our goal is to take a lottery pricing $\mathcal{L}$ and construct an auction $\mathcal{M}_{\mathcal{L}}$ for the representative environment such that the sum of its revenue with the revenue of the second-price auction is at least the revenue of the original lottery pricing. Both the constructed auction $\mathcal{M}_{\mathcal{L}}$ and the second-price auction revenues can be bounded from above by the optimal auction revenue; therefore, twice the optimal auction revenue is at least the revenue of the lottery pricing.

We construct $\mathcal{M}_{\mathcal{L}}$ as follows. Consider a valuation profile $\mathbf{v}$ for which the multi-dimensional agent selects lottery $l = ((q_1, \ldots, q_m), p)$. This agent receives utility $\sum_j v_j q_j - p$. As usual we denote by (j) the index of the jth highest value, i.e., $v_{(1)} \geq \ldots \geq v_{(m)}$. For such a valuation profile $\mathcal{M}_{\mathcal{L}}$ serves representative (1) with probability $q_{(1)}$ and charges this representative $p - \sum_{j \neq (1)} q_j v_j$ (always). This representative's utility is therefore perfectly aligned with our original multi-dimensional agent. Since the original agent preferred this lottery over all others, so does the representative; i.e., $\mathcal{M}_{\mathcal{L}}$ is incentive compatible.

By definition revenue of $\mathcal{M}_{\mathcal{L}}$ in the above environment is $p - \sum_{j \neq (1)} q_j v_j$. Consider the second part of this formula. This is the rebate we need to give representative (1) in order to incentivize the representative to prefer this lottery over all others. By definition the total probability to which the original agent is served by this lottery is at most one. Therefore, $\sum_{j \neq (1)} q_j v_j \leq v_{(2)}$, the second highest value. The revenue of $\mathcal{M}_{\mathcal{L}}$ is at least $p - v_{(2)}$. Recall that the revenue of the second-price auction is exactly $v_{(2)}$. Taking expectations over all valuation profiles, the expected revenue of $\mathcal{M}_{\mathcal{L}}$ is at least that of the original lottery pricing less that of the second price auction. Rearranging gives the desired inequality. □

A similar theorem can be proven in environments with multiple agents such as that of matching markets. The proof uses matroid properties and the basic intuition from the single-agent case, above. We omit the full proof from this text.

Theorem 7.14. *For independent, matching market environments, the optimal (randomized) mechanism's revenue is at most five times that of the optimal mechanism for the single-dimensional representative environment.*

7.4 Beyond Independent Unit-demand Environments

Up to this point the chapter has focused on independent unit-demand environments, i.e., ones where there is some set of services available, each agent desires at most one service, and the agent's value for each service are independent random variables. Unfortunately, not much is known about general distributions of preferences. In particular, environments where an agent desires more than one service or environments where an agent's value for distinct services are correlated. There are two notable exceptions, one from each of these classes.

The first exception is for "common base value" distributions, i.e., ones where the agents value for a service j is $v_0 + v_j$ and for $0 \leq j \leq m$, v_j are independent. The value v_0 is referred to as the base value because it offsets the values for each service. To this environment most of the preceding theorems can be extended, albeit with worse approximation factors. Unfortunately, the proofs of these extensions are brute-force and do not yield much additional understanding of the structure of good mechanisms in the common base value model.

The second exception is for additive preferences, i.e., where the agent's value for a bundle of services is the sum of the agent's value for each service. Again the agent's value for each individual service is independently distributed. In this environment, again for simple reasons, the optimal mechanism can be approximated. Sums of independent random variables tend to concentrate around their expectation. Therefore, it is possible to offer an agent a posted price for the grand bundle of items that is close to but below the this expectation and nearly the full surplus can be extracted.

Beyond these two cases, not much is known about approximately optimal mechanisms for general preferences. A major challenge in this research area is in identifying reasonable, analytically tractable upper bounds on the optimal multi-dimensional mechanism.

7.5 Optimal Lottery-pricing via Linear Programming

While there is little economic understanding of optimal mechanisms when agents' preferences are multi-dimensional that does not necessarily mean that the optimization problem is intractable. For the distributions on preference discussed heretofore, e.g., when the value that an agent has for various items is distributed independently, then an exponentially large type space can be described succinctly. In particular, each single-dimensional distribution need only be described. For such a distribution, a mechanism that was brute-force, i.e., its calculation explicitly considers every type in the type space, would be intractable.

On the other hand, if the type space is small enough that the distribution can be given explicitly, i.e., each type is given with its associated probability, then mechanisms that are brute-force in the type space may be reasonable. For the m-item, single-agent environment, for instance, the optimal lottery pricing in such a situation can be easily calculated.

Suppose the type space is given explicitly as follows. The agent has type $t \in \{1, \ldots, N\}$; denote by $\mathbf{v}$ the $N \times m$ matrix of values; let v_{tj} be the agent's value for item j when her type is t; and let π_t be the probability her type is t. We can write the optimization problem now as a linear program. The linear program will associate with each type t a lottery given by a price p_t and the probabilities for receiving each of the m items $(x_{t1}, \ldots, x_{tm})$. Notice that the agent's utility with type t for the lottery predesignated for type t' is $\sum_t v_{tj} x_{t'j} - p_{t'}$. The linear program will maximize expected payments (weighted by the distribution) subject to incentive constraints, individual rationality constraints, and probabilities summing to at most one (feasibility).

Maximize:

$$\sum\nolimits_t \pi_t p_t \qquad \text{(expected revenue)}$$

Subject to:

$$\begin{aligned}
&\forall t, t' && \sum\nolimits_t v_{tj} x_{tj} - p_t \geq \sum\nolimits_t v_{tj} x_{t'j} - p_{t'} && \text{(incentive compatibility)} \\
&\forall t && \sum\nolimits_t v_{tj} x_{tj} - p_t \geq 0 && \text{(individual rationality)} \\
&\forall t && \sum\nolimits_j x_{tj} \leq 1 && \text{(feasibility)}
\end{aligned}$$

It is easy to see that when the type space and distribution are given explicitly that this program can be easily solved for the optimal set of lotteries to offer.

Lottery pricings correspond to fractional solutions of the linear program above; when the variables x_{tj} are integer these are simply (determinstic) item pricings. Calculating optimal item pricings in correlated environments, i.e., solving this mixed-integer program where x_{tj}s are constrained to be integral, is extremely challenging. For this problem, obtaining any approximation factor that is asymptotically better than linear in the number of items is computationally intractable under reasonable assumptions. Of course, a linear factor approximation is trivial. Formally:

Theorem 7.15. *Under complexity-theoretic assumptions, the problem of computing prices that $o(m)$-approximate the revenue of the optimal item pricing is computationally intractable.*

The computational intractability of a problem, and this perspective is discussed more in Chapter 8, suggests that there is inherent inability to make important structural observations.

Exercises

7.1 Consider the design of prior-free incentive-compatible mechanisms with revenue that approximates the (optimal) social-surplus benchmark, i.e., $\text{OPT}(\mathbf{v})$, when all values are known to be in a bounded interval $[1, h]$. For general (multi-dimensional) combinatorial auctions, i.e., there are m items and each agent i has a value $v_i(S') \in [1, h]$ for each subset $S' \subseteq S = \{1, \ldots, m\}$ of the m items, give a prior-free $\Theta(\log h)$-approximation mechanism.

7.2 Prove Theorem 7.5: For any independent, unit-demand environment, the optimal deterministic mechanism's revenue is at most that of the optimal mechanism for the single-dimensional representative environment.

7.3 Recall that Theorem 7.10 shows that for the representative matching market environment, a sequential posted pricing gives an 8-approximation to the optimal (single-dimensional) mechanism. This bound can be improved.

(a) Give an improved bound.

(b) Assume that the agents are identically distributed (but not necessarily the items) and give an improved bound.

(c) Assume that both the agents and the items are identically distributed and given an improved bound.

7.4 Consider the design of prior-independent mechanisms for (multi-dimensional) unit-demand agents. Suppose there are n agents and $m = n$ houses and agent i's value for house j is drawn independently from a regular distribution F_j. (I.e., the agents are i.i.d., but the houses are distinct.) Give a prior-independent mechanism that approximates the Bayesian optimal mechanism. What is your mechanism's approximation factor?

Chapter Notes

There is a long history of study of multi-dimensional pricing and mechanism design in economics. Wilson's text *Nonlinear Pricing* is a good reference for this area (Wilson, 1997).

Algorithmic questions related to item-pricing for unit-demand agents were initiated by Aggarwal et al. (2004) and Guruswami et al. (2005) in an environment where the agent's values are correlated. The hardness of $o(m)$-approximation for such an m-item environment, i.e., Theorem 7.15, is due to Briest (2008). On the other hand Briest et al. (2010) show that optimal lottery pricings can be calculated via a linear program that is polynomially big in the support of the (correlated) distribution of the agent's valuations.

Approximation for item-pricings when the agent's values are independent were first studied by Chawla et al. (2007) where a 3-approximation was given. The 2-approximation via prophet inequalities that is presented in this chapter is due to Chawla et al. (2010a). Cai and Daskalakis (2011) show that it is computationally tractable to construct a pricing that approximates the revenue of the optimal pricing to within any multiplicative factor. The example presented herein that shows that a lottery pricing can give more revenue than the optimal item pricing was given by Thanassoulis (2004). Lottery pricings and the theorem that shows that the optimal lottery pricing is at most a factor of two more than the optimal mechanism's revenue in the single-dimensional representative environment is due to Chawla et al. (2010b).

The study of sequential posted pricing mechanisms in multi-dimensional environments that is discussed in this chapter is given by Chawla et al. (2010a); these sequential posted pricings are constant approximations to the optimal deterministic mechanisms. Alaei (2011) gives a refined analysis and approach. Extensions of these results to bound the revenue of the sequential posted pricing in terms of the optimal (randomized) mechanism's revenue are from Chawla et al. (2010b). Neither the bound of two (for single-agent lottery pricing) or five (for matching markets) is known to be tight.

Extensions from product distributions to the common base value environment are given in Chawla et al. (2010b). Briest et al. (2010) study general environments with correlated values and show that when more than $m = 4$ services are available then the ratio between the optimal lottery pricing (i.e., randomized mechanism) and the optimal item pricing (i.e., deterministic mechanism) is unbounded. This contrasts starkly with environment with independent values where Theorem 7.13 shows that the ratio is at most two. Finally, the independent additive values case, where pricing the grand bundle gives an asymptotically optimal revenue, was studied by Armstrong (1996).

Chapter 8

Computational Tractability

In many relevant environments optimal mechanisms are computationally intractable. A mechanism that is computationally intractable, i.e., where no computer could calculate the outcome of the mechanism in a reasonable amount of time, seems to violates even the loosest interpretation of our general desideratum of simplicity.

We will try to address this computational intractability by considering approximation. In particular we will look for an approximation mechanism, one that is guaranteed to achieve a performance that is close to the optimal, intractable, mechanism's performance. A first issue that arises in this approach is that approximation algorithms are not generally compatible with mechanism design. The one approach we have discussed thus far, following from generalization of the second-price auction, fails to generically convert approximation algorithms into dominant-strategy incentive-compatible approximation mechanisms.

Dominant-strategy incentive-compatible and prior-free mechanisms may be too demanding for this setting. In fact, without making an assumptions on the environment the approximation factor of worst case algorithms can be provably too far from optimal to be practically relevant. We therefore turn to Bayesian mechanism design and approximation. Here we give a reduction from BIC mechanism design to algorithm design. This reduction is a simple procedure that converts any algorithm into a Bayesian incentive-compatible mechanism without compromising its expected social surplus. If the algorithm is is tractable then the resulting mechanism is too. We conclude that under the BIC implementation concept incentive constraints and tractability constraints can be completely disentangled and any good algorithm or heuristic can be converted into a mechanism.

8.1 Tractability

Our philosophy for mechanism design is that a mechanism that is not computationally tractable is not a valid solution to a mechanism design problem. To make this criterion formal we review the most fundamental concepts from computational complexity. Readers are encouraged to explore these topics in greater detail outside this text.

Definition 8.1. *$\mathcal{P}$ is the class of problems that can be* solved *in polynomial time (in the* *"size of the input").*

Recall, from Chapter 4 the greedy algorithm for optimizing independent sets in a matroid. This algorithm sorts the agents by value, and then greedily, in this order, tries to add the agents to an independent set. Sorting takes $O(n \log n)$ and checking independence is usually fairly easy. Therefore, this algorithm runs in polynomial time.

Definition 8.2. *$\mathcal{NP}$ is the class of problem that can be* verified *in polynomial time.*

$\mathcal{NP}$ stands for *non-deterministic polynomial time* in that problems in this class can be solved by "guessing" the solution and then verifying that indeed the solution is correct. Of course, non-deterministic computers that can guess the correct solution do not exist. A real computer could of course simulate this process by iterating over all possible solutions and checking to see if the solution is valid. Unfortunately, nobody whether there is an algorithm that improves significantly on exhaustive search. Exhaustive search, for most problems, requires exponential runtime and is therefore considered intractable.

Consider the problem of verifying whether a given solution to the single-minded combinatorial auction problem has surplus at least V. If we were given an outcome $\mathbf{x}$ that for which $\text{Surplus}(\mathbf{v}, \mathbf{x}) \geq V$ it would be quite simple to verify. First, we would verify whether it is feasible by checking all i and i' with $x_i = x_{i'} = 1$ (i.e., all pairs of served agents) that $S_i \cap S_{i'} = \emptyset$ (i.e., their bundles do not overlap). Second, we would calculate the total welfare $\sum_i v_i x_i$ to ensure that it is at least V. The total runtime of such a verification procedure is $O(n^2)$.

While the field of computer science has failed to determine whether or not $\mathcal{NP}$ problems can be solved in polynomial time or not, it has managed to come to terms with this failure. The following approach allows one to leverage this collective failure to argue that a given problem X is unlikely to be polynomial-time solvable by a computer.

Definition 8.3. *A problem Y* reduces *(in polynomial time) to a problem X if we can solve any instance y of Y with a polynomial number (in the size of y) of basic computational steps and queries to a blackbox that solves instances of X.*

Definition 8.4. *A problem X is $\mathcal{NP}$-*hard *if all problems $Y \in \mathcal{NP}$ reduce to it.*

Definition 8.5. *A problem X is $\mathcal{NP}$-*complete *if $X \in \mathcal{NP}$ and X is $\mathcal{NP}$-hard.*

The point of these definitions is this. Many computer scientists have spent many years trying to solve $\mathcal{NP}$-complete problems and failed. When one shows a new problem X is $\mathcal{NP}$-hard, one is showing that if this problem can be solved, then so can all $\mathcal{NP}$ problems, even the infamously difficult ones. While showing that a problem cannot be solved in polynomial time is quite difficult, showing that it is $\mathcal{NP}$-hard is usually quite easy (if it is true). Therefore it is quite possible to show, for some new problem X, that under the assumption that $\mathcal{NP}$-hard problems cannot be solved in polynomial time (i.e., $\mathcal{NP} \neq \mathcal{P}$), that X cannot be solved in polynomial time.

We will make the standard assumption that $\mathcal{NP} \neq \mathcal{P}$ which implies that $\mathcal{NP}$-hard problems are computationally intractable.

8.2 Single-minded Combinatorial Auctions

Consider the example environment of single-minded combinatorial auctions. This environment is important as it is a special case of more general auction settings such as the FCC spectrum auctions (for selling radio-frequency broadcast rights to to cellular phone companies) and sponsored search auctions (for selling advertisements to be show along side on search-results page of Internet search engines). In single-minded combinatorial auctions each agent i has a value v_i for receiving a bundle S_i of m distinct items. Of course each item can be allocated to at most one agent so the intersection of the desired bundles of all pairs of served agents must not intersect.

The optimization problem of single-minded combinatorial auctions, also known as *weighted set packing*, is intractable. We state but do not prove this result here.

Theorem 8.6. *The single-minded combinatorial auction problem is $\mathcal{NP}$-complete.*

8.2.1 Approximation Algorithms

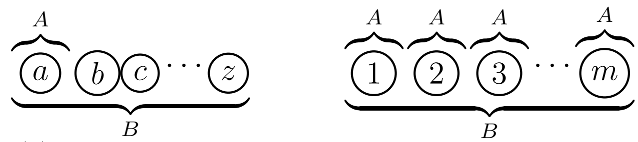
When optimally solving a problem is $\mathcal{NP}$-hard the standard approach from the field of algorithms is to obtain a polynomial time approximation algorithm, i.e., an algorithm that guarantees in worst-case to output a solution that is within a prespecified factor of the optimal solution.

As a first step at finding an approximation algorithm it is often helpful to look at simple-minded approaches that fail to give good approximations. The simplest algorithmic design paradigm is that of *static greedy algorithms*. Static greedy algorithms for general feasibility settings follow the following basic framework.

Algorithm 8.1. *A* static greedy algorithm *is*

1. *Sort the agents by some prespecified criterion.*
2. $\mathbf{x} \leftarrow \mathbf{0}$ *(the null assignment).*
3. *For each agent i (in this sorted order),*
   if $(\mathbf{x}_{-i}, 1)$ is feasible, $x_i \leftarrow 1$.
   (I.e., serve i if i can be served along side previously served agents.)
4. *Output* $\mathbf{x}$.

The first failed approach to consider is *greedy by value*, i.e., the prespecified sorting criterion in the static greedy template above is by agent values v_i. This algorithm is bad because it is an $\Omega(m)$-approximation on the following $n = m+1$ agent input. Agents i, for $0 \leq i \leq m$, have $S_i = \{i\}$ and $v_i = 1$; agent $m+1$ has $v_{m+1} = 1 + \epsilon$ and demands the grand bundle $S_{m+1} = \{1, \ldots, m\}$ (for some small $\epsilon > 0$). See Figure 8.1(a) with $A = 1$ and $B = 1 + \epsilon$. Greedy-by-value orders agent $m+1$ first, this agent is feasible and therefore served. All remaining agents are infeasible after agent $m+1$ is served. Therefore, the



(a) m small agents. (b) 1 small agent.

Figure 8.1: Challenge cases for greedy orderings as a function of value and bundle size.

algorithm serves only this one agent and has surplus $1 + \epsilon$. Of course OPT serves the m small agents for a total surplus of m. The approximation factor of greedy-by-value is the ratio of these two performances, i.e., $\Omega(m)$.

Obviously what went wrong in greedy-by-value is that we gave preference to an agent with large demand who then blocked a large number of mutually-compatible small-demand agents. We can compensate for this by instead sorting by value-per-item, i.e., $v_i/|S_i|$. *Greedy by value-per-item* also fails on the following $n = 2$ agent input. Agents 1 has $S_1 = \{1\}$ and $v_1 = 1 + \epsilon$ and agent 2 has $v_2 = m$ demands the grand bundle $S_2 = \{1, \ldots, m\}$. See Figure 8.1(b) with $A = 1+\epsilon$ and $B = m$. Greedy-by-value-per-item orders agent 1 first, this agent is feasible and therefore served. Agent 2 is infeasible after agent 1 is served. Therefore, the algorithm serves only agent 1 and has surplus $1 + \epsilon$. Of course OPT serves agent 2 and has surplus of m. The approximation factor of greedy-by-value-per-item is the ratio of these two performances, i.e., $\Omega(m)$.

The flaw with this second algorithm is that it makes the opposite mistake of the first algorithm; it undervalues large-demand agents. While we correctly realized that we need to trade off value for size, we have only considered extremal examples of this trade-off. To get a better idea for this trade-off, consider the cases of a single large-demand agent and either m small-demand agents or 1 small-demand agent. We will leave the values of the two kinds of agents as variables A for the small-demand agent(s) and B for the large-demand agent. Assume, as in our previous examples, that $mA > B > A$. These settings are depicted in Figure 8.1.

Notice that any greedy algorithm that orders by some function of value and size will either prefer A-valued or B-valued agents in both cases. The A-preferred algorithm has surplus Am in the m-small-agent case and surplus A in the 1-small-agent case. The B-preferred algorithm has surplus B in both cases. OPT, on the other hand, has surplus mA in the m-small-agent case and surplus B in the 1-small-agent case. Therefore, the worst-case ratio for A-preferred is B/A (achieved in the 1-small-agent case), and the worst-case ratio for B-preferred is mA/B (achieved in the m-small-agent case). These performances and worst-case ratios are summarized in Table 8.1.

If we are to use the static greedy algorithm design paradigm we need to minimize the worst-case ratio. The approach suggested by the analysis of the above cases would be trade off A versus B to equalize the worst-case ratios, i.e., when $B/A = mA/B$. Here m was a stand-in for the size of the large-demand agent. Therefore, the greedy algorithm that this suggests is to order the agents by $v_i/\sqrt{|S_i|}$. This algorithm was first proposed for use in

	m small agents	1 small agent	worst-case ratio
OPT	mA	B	(n.a.)
A-preferred	mA	A	B/A
B-preferred	B	B	mA/B

Table 8.1: Performances of A- and B-preferred greedy algorithms and their ratios to OPT.

single-minded combinatorial auctions by Daniel Lehmann, Liadan O'Callaghan, and Yoav Shoham and is often referred to as the LOS algorithm.

Algorithm 8.2. *Sort the gents by value-per-square-root-size, i.e.,* $v_i/\sqrt{|S_i|}$*, and serve them greedily while supplies last.*

Theorem 8.7. *The greedy by value-per-square-root-size algorithm is a* $\sqrt{m}$*-approximation algorithm (where* m *is the number of items).*

Proof. Let APX represent the greedy by value-per-square-root-size algorithm and its surplus; let REF represent the optimal algorithm and its surplus. Let I be the set selected by APX and I^* be the set selected by REF. We will proceed with a *charging argument* to show that if $i \in I$ blocks some set of agents $F_i \subset I^*$ then the total value of the blocked agents is not too large relative to the value of agent i.

Consider the agents sorted (as in APX) by $v_i/\sqrt{|S_i|}$. For an agent $i^* \in I^*$ not to be served by APX, it must be that at the time it is considered by LOS, another agent i has already been selected that *blocks* i^*, i.e., the demand sets S_i and S_{i^*} have non-empty intersection. Intuitively we will charge i with the loss from not accepting this i^*. We define F_i as the set of all $i^* \in I^*$ that are charged to i as described above. Of special note, if $i^* \in I$, i.e., it was not yet blocked when considered by APX, we charge it to itself, i.e., $F_{i^*} = \{i^*\}$. Notice that the sets F_i partition the agents I^* of REF.

The theorem follows from the inequalities below. Explanations of each non-trivial step are given afterwards.

$$\mathrm{REF} = \sum_{i^*} v_{i^*} = \sum_{i \in I} \sum_{i^* \in F_i} v_{i^*} \tag{8.1}$$

$$\leq \sum_{i \in I} \tfrac{v_i}{\sqrt{|S_i|}} \sum_{i^* \in F_i} \sqrt{|S_{i^*}|} \tag{8.2}$$

$$\leq \sum_{i \in I} \tfrac{v_i}{\sqrt{|S_i|}} \sum_{i^* \in F_i} \sqrt{m/|F_i|} \tag{8.3}$$

$$= \sum_{i \in I} \tfrac{v_i}{\sqrt{|S_i|}} \sqrt{m|F_i|} \tag{8.4}$$

$$\leq \sum_{i \in I} v_i \sqrt{m} = \sqrt{m} \cdot \mathrm{APX}\,. \tag{8.5}$$

Line (8.1) follows because F_i partition I^*. Line (8.2) follows because $i^* \in F_i$ implies that i precedes i^* in the greedy ordering and therefore $v_i^* \leq v_i\sqrt{|S_{i^*}|}/\sqrt{|S_i|}$. Line (8.3) follows because the demand sets S_{i^*} of $i^* \in F_i$ are disjoint (because they are a subset of I^* which is feasible and therefore disjoint). Thus we can bound $\sum_{i^* \in F_i} |S_{i^*}| \leq m$. The square-root function is concave and the sum of a concave function is maximized when each term is equal, i.e., when $S_{i^*} = m/|F_i|$. Therefore, $\sum_{i^* \in F_i} \sqrt{|S_{i^*}|} \leq \sum_{i^* \in F_i} \sqrt{m/|F_i|} = \sqrt{m|F_i|}$. This last equality gives line (8.4). Finally, line (8.5) follows because $|F_i| \leq |S_i|$ which holds because each $i^* \in F_i$ is disjoint but blocked by i because each contains some demanded item in S_i. Thus, S_i contains at least $|F_i|$ distinct items. □

The as witnessed by the theorem above, the greedy by value-per-square-root-size algorithm gives a non-trivial approximation factor. A $\sqrt{m}$-approximation, though, hardly seems appealing. Unfortunately, it is unlikely that there is a polynomial time algorithm with better worst-case approximation factor, but we do not provide proof in this text.

Theorem 8.8. *Under standard complexity-theoretic assumptions,*[1] *no polynomial time algorithm gives an $o(\sqrt{m})$-approximation to weighted set packing.*

8.2.2 Approximation Mechanisms

Now that we have approximated the single-minded combinatorial auction problem without incentive constraints, we need add these constraints back in and see whether we can derive a $\sqrt{m}$-approximation mechanism.

We first note that we cannot simply use the formula for externalities from the surplus maximization mechanism for payments when we replace the optimal algorithm OPT with some approximation algorithm $\mathcal{A}$. I.e., $\mathbf{x} = \mathcal{A}(\mathbf{v})$ and $p_i = \mathcal{A}(\mathbf{v}_{-i}) - \mathcal{A}_{-i}(\mathbf{v})$ is not incentive compatible. An example demonstrating this with the greedy algorithm can be seen by m agents each i demanding the singleton bundle $S_i = \{i\}$ with value $v_i = 1$ and a final agent $m+1$ demanding the grand bundle $S_{m+1} = \{1, \ldots, m\}$ with value $\sqrt{m}+\epsilon$ (See Figure 8.1(a) with $A = 1$ and $B = \sqrt{m} + \epsilon$). On such an input the greedy algorithm APX accepts only agent $m + 1$. However, when computing the payment with the externality formula $p_{m+1} = \text{APX}(\mathbf{v}_{-(m+1)}) - \text{APX}_{-(m+1)}(\mathbf{v})$ we get $p_{m+1} = m$. This payment is higher than agent $m + 1$'s value and the resulting mechanism is clearly not incentive compatible.

Mirroring our derivation of the monotonicity of the surplus maximization mechanism in Chapter 3 Section 3.2, the BNE characterization requires each agent's allocation rule be monotone, therefore any incentive compatible mechanism must be monotone. Even though, in our derivation of the greedy algorithm no attempt was made to obtain monotonicity, it is satisfied anyway.

Lemma 8.9. *For each agent i and all values of other agents $\mathbf{v}_{-i}$, the i's allocation rule in the greedy by value-per-square-root-size algorithm is monotone in i's value v_i.*

[1] I.e., assuming that $\mathcal{NP}$-complete problems cannot be solved in polynomial time by a randomized algorithm.

Proof. It suffices to show that if i with value v_i is served by the algorithm on $\mathbf{v}$ and i increases her bid to $b_i > v_i$ then they will continue to be served. Notice that the set of available items is non-increasing as each agent is considered. If i increases her bid she will be considered only earlier in the greedy order. Since items S_i were available when i is originally considered, they will certainly be available if i is considered earlier. Therefore, i will still be served with a higher bid. □

The above proof shows that there is a critical value τ_i for i and if $v_i > \tau_i$ then i is served. It is easy to identify this critical value by simulating the algorithm on $\mathbf{v}_{-i}$. Let i' be the earliest agent in the simulation to demand and receive an item from S_i. Notice that if i comes after i' then i will not be served because S_i will no longer be completely available. However, if i comes before i' then i can and will be served by the algorithm. Therefore i's critical value is the τ_i for which $v_i = \tau_i$ would tie agent i' in the ordering. I.e., $\tau_i = v_{i'}\sqrt{|S_i|}/\sqrt{|S_{i'}|}$.

We conclude by formally giving the approximation mechanism induced by the greedy by value-per-square-root-size algorithm and the theorem and corollary that describe its incentives and performance.

Mechanism 8.1. *The greedy by value-per-square-root-size mechanism is:*

1. *Solicit and accept sealed bids* $\mathbf{b}$.
2. $\mathbf{x} =$ *GVPSS*($\mathbf{b}$), *and*
3. $\mathbf{p} =$ *critical values for GVPSS on* $\mathbf{b}$,

where GVPSS denotes the greedy by value-per-square-root-size algorithm.

Theorem 8.10. *The greedy by value-per-square-root-size mechanism is dominant strategy incentive compatible.*

Corollary 8.11. *The greedy by value-per-square-root-size mechanism gives a $\sqrt{m}$-approximation to the optimal social surplus in dominant strategy equilibrium.*

At this point it is important to note that again we have gotten lucky in that we attempted to approximate our social surplus objective without incentive constraints and the approximation algorithm we derived just happened to be monotone, which is all that is necessary for the mechanism with that allocation and the appropriate payments to be incentive compatible. If we had been less fortunate and our approximation algorithm not been monotone, it could not have been turned into a mechanism so simply. We conclude this section with three important questions.

Question 8.1. *When are approximation algorithms monotone?*

Question 8.2. *When an approximation algorithm is not monotone, is it possible to derive from it a monotone algorithm that does not sacrifice any of the original algorithm's performance?*

Question 8.3. *For real practical mechanism design where there are no good approximation algorithms, what can we do?*

To get a hint at the answer to the first of these questions, we note that the important property of the LOS algorithm that implied its monotonicity was the greedy ordering. We conclude that any static greedy algorithm that orders agents as a monotone function of their value is monotone. The proof of this theorem is identical to that for LOS (Lemma 8.9).

Theorem 8.12. *For any set of n monotone non-decreasing functions $f_1(\cdot), \ldots, f_n(\cdot)$ the static greedy algorithm that sorts the agents in a non-increasing order of $f_i(v_i)$ is monotone.*

8.3 Bayesian Algorithm and Mechanism Design

In the preceding section we saw that worst case approximation factors for tractable algorithms can may be so large that they do not distinguish between good algorithms and bad ones. We also noted that mechanisms must satisfy an additional requirement beyond just having good performance; the allocation rule must also be monotone. For the example of combinatorial auctions we were lucky and our approximation algorithm was monotone. Beyond greedy algorithms, such luck is the exception rather than the rule. Indeed, the entanglement of the monotonicity constraint with the original approximate optimization problem that the designer faces suggests that approximation mechanism design, from a computational point of view, could be more difficult than approximation algorithm design.

Imagine a realistic setting where a designer wishes to design a mechanism for some environment where worst-case approximation guarantees do not provide practical guidance in selecting among algorithms and mechanisms. Without some foreknowledge of the environment, improving beyond the guarantees of worst-case approximation algorithms is impossible. Therefore, let us assume that our designer has access to a representative data set. The designer might then attempt to design a good algorithm for this data set. Such an algorithm would have good performance on average over the data set. In fact, in most applied areas of computer science this methodological paradigm for algorithm design is prevalent.

Algorithm design in such a statistical setting is a bit of an art; however, the topic of this text is mechanism design not algorithm design. So let us assume that this algorithmic design problem is solved. Our mechanism design challenge is then to reduce the mechanism design problem to this algorithm design problem, i.e., to show that any algorithm, with access to the true private values of the agents, can be turned into a mechanisms, where the agents may strategize, and in equilibrium the outcome of the mechanism is as good as that of the algorithm. Such a result would completely disentangle the incentive constraints from the algorithmic constraints. Notice that the approach of the surplus maximization mechanism (Chapter 3) approach solves this mechanism design problem for an optimal algorithm designer; here we solve it for an *ad hoc* algorithm designer.

The statistical environment discussed above is well suited to the Bayesian mechanism design approach that has underlied most of the discussion in this text. The main result of this section is the polynomial-time constructive proof of the following theorem.

Theorem 8.13. *For any single-dimensional agent environment, any product distribution* $\mathbf{F}$*, and any algorithm* $\mathcal{A}$*, there is a BIC mechanism* $\bar{\mathcal{A}}$ *satisfying* $\mathbf{E}_{\mathbf{v}\sim\mathbf{F}}\big[\bar{\mathcal{A}}(\mathbf{v})\big] \geq \mathbf{E}_{\mathbf{v}\sim\mathbf{F}}[\mathcal{A}(\mathbf{v})]$.

8.3.1 Monotonization

Let $\mathbf{x}(\mathbf{v})$ denote the allocation produced by the algorithm on input $\mathbf{v}$. For agent i the algorithms interim allocation rule is $x_i(v_i) = \mathbf{E}_{\mathbf{v}\sim\mathbf{F}}[x_i(\mathbf{v}) \mid v_i]$. Recall, this is the probability that we allocate to i when i has value v_i and the other agents' values are drawn from the distribution. If $x_i(\cdot)$ is monotone non-decreasing then there exist a payment rule, via the payment identity in the BIC characterization (Corollary 2.16), such that truthtelling is a best response for i (assuming others also truthtell). If $x_i(\cdot)$ is non-monotone then there is no such payment rule. Therefore the challenge before us is the potential non-monotonicity of $x_i(\cdot)$. Our goal will be to construct an $\bar{x}_i(\cdot)$ from $x_i(\cdot)$ with the following properties.

1. (monotonicity) $\bar{x}_i(\cdot)$ is monotone non-decreasing.
2. (surplus preservation) $\mathbf{E}_{v_i\sim F_i}[v_i\bar{x}_i(v_i)] \geq \mathbf{E}_{v_i\sim F_i}[v_i x_i(v_i)]$.
3. (locality) No other agent j can tell whether we run $x_i(v_i)$ or $\bar{x}_i(v_i)$.

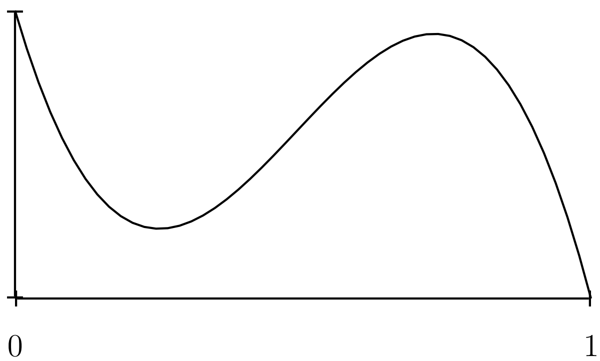
That the first two conditions are needed is intuitively clear. Monotonicity is required for Bayesian incentive compatibility. Surplus preservation is required if our construction is to not harm our objective. The requirement of locality is more subtle; however, notice that if no other agent can tell whether we are running $x_i(\cdot)$ or $\bar{x}_i(\cdot)$ then we can independently apply this construction to each agent.

We will assume that the distribution F_i is continuous on its support and we will consider the allocation rule in quantile-space (cf. Chapter 3 Section 3.3.1). The transformation from value space to quantile space, recall, is given by $q_i = 1 - F_i(v_i)$. We denote the value associated with a given quantile as $v_i(q_i) = F_i^{-1}(1-q_i)$ and we express the allocation rule in quantile space as $x_i(q_i) = x_i(v_i(q_i))$. Notice that the quantile of agent i is drawn uniformly from $[0, 1]$.

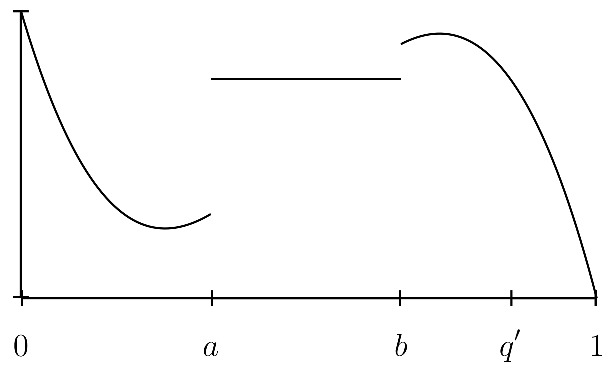
We will focus for the sake of exposition on a two agent case and name the agents Alice and Bob. Our goal will be to monotonize Alice's allocation rule without affecting Bob's allocation rule. Our discussion will focus on Alice and for notational cleanliness we will drop subscripts. Alice's quantile is $q \sim U[0, 1]$, her value is $v(q)$, her allocation rule (in quantile space) is $x(q)$. We assume that $x(\cdot)$ is not monotone non-increasing and focus on monotonizing it.

Resampling and Locality

Notice first, if the allocation rule for Alice is non-monotone over some interval $[a, b]$ then one way to make it monotone in this interval is to treat her the exact same way regardless of where her value lies within this interval. This would result in a constant allocation in the interval and a constant allocation is non-decreasing, as desired. There are many ways to do this, for instance, if $q \in [a, b]$ we can run $x(q')$ instead of $x(q)$. (Back in valuation space, this can be implemented by ignoring Alice's value v and inputing $v' = v(q')$ into the algorithm instead.) Unfortunately, if we did this, Bob would notice. The distribution of Alice's input would no longer be F, for instance it would have no mass on interval (a, b) and a point mass on q'. See Figure 8.2(b).



(a) allocation rule $x(q)$.



(b) allocation rule $x'(q)$.

Figure 8.2: The allocation rule $x(q)$ and $x'(q)$ constructed by running $x(q')$ when $q \in [a, b]$.

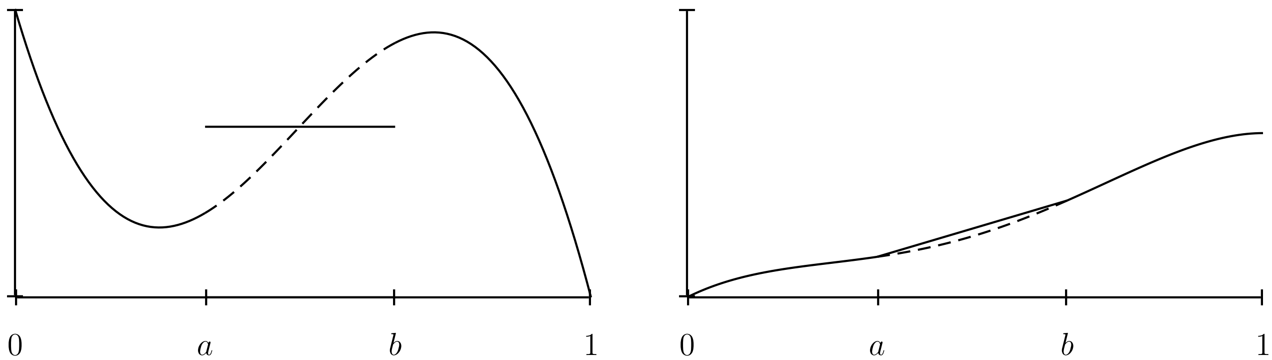
Notice second, there is a very natural way to fix the above construction to leave the distribution of Alice's input to the algorithm unchanged. Instead of the arbitrary choice of inputing q' into the algorithm we can resample the distribution F on interval $[a, b]$. In quantile space this is corresponds precisely with uniformly picking q' from $[a, b]$. Formally, the proposed transformation is the following. If $q \in [a, b]$ then resample $q' \sim U[a, b]$. If $q \notin [a, b]$ then simply set $q' = q$. Now run $x(q')$ instead of $x(q)$, i.e., simulate the algorithm with input $v' = v(q')$ in place of Alice's original value. Let $x'(q)$ denote the allocation rule of the simulation as a function of Alice's original value probability. Notice that for $q \notin [a, b]$ we have $x'(q) = x(q)$. For $q \in [a, b]$, Alice receives the average allocation probability for the interval $[a, b]$, i.e, $\frac{1}{b-a} \int_a^b x(r)\, dr$. See Figure 8.3(a).

Notice third, this approach is likely to improve Alice's expected surplus. Suppose $x(q)$ is increasing on $[a, b]$, meaning higher values are less likely to be allocated than low values, then this monotonization approach is just shifting allocation probability mass from low values to higher values.

Interval Selection and Monotonicity

The allocation rule $x'(\cdot)$ constructed in this fashion, while monotone over $[a, b]$, may still fail to be monotone. Though intuitively it should be clear that the resampling process can replace a non-monotone interval of the allocation rule with a constant one, we still need to ensure that the final allocation rule is monotone. Of special note is the potential discontinuity of the allocation rule at the end-points of the interval.

Notice first, that $x'(q)$ is monotone if and only if its integral, $X'(q) = \int_0^q x'(r)dr$, is convex. We will refer to $X'(q)$ and $X(q)$ (defined identically for $x(q)$) as *cumulative allocation rules.*



(a) allocation rules $x(q)$ and $x'(q)$. (b) cumulative allocation rules $X(q)$ and $X'(q)$.

Figure 8.3: The allocation rule $x(q)$ (dashed) and $x'(q)$ (solid) constructed by drawing $q' \sim U[a, b]$ and running $x(q')$ when $q \in [a, b]$. Corresponding cumulative allocation rules $X(q)$ (dashed) and $X'(q)$ (solid).

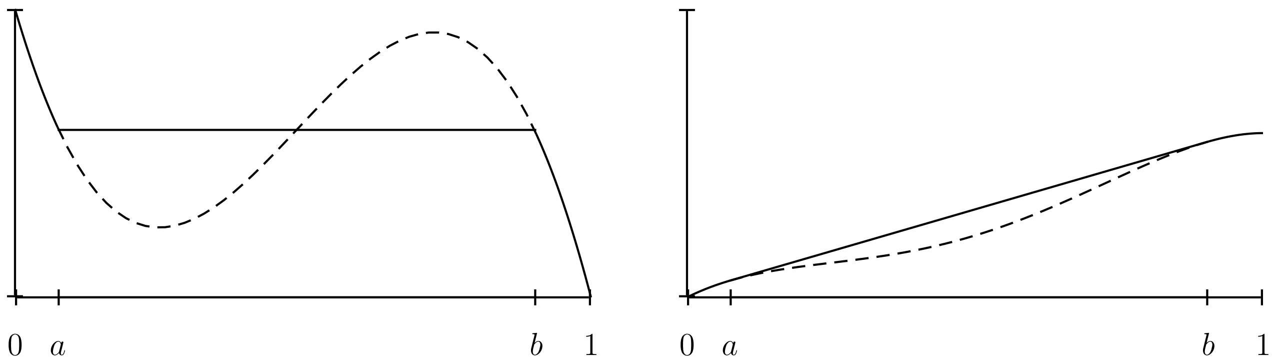
Notice second, the implication for $X'(q)$ of the resampling procedure on $X(q)$. Consider some $q \leq a$ as $x(q) = x'(q)$, clearly $X(q) = X'(q)$. In particular $X(a) = X'(a)$. Now calculate $X(b)$ and $X'(b)$. These are equal to $X(a)$ plus the integral of the respective allocation rules on $[a, b]$. Of course $x'(q)$ is constant on $[a, b]$ and equal, by definition, to $\frac{1}{b-a}\int_a^b x(r)dr$. The integral of a constant function is simply the value of the function times the length of the interval. Therefore $\int_a^b x'(r)dr = \int_a^b x(r)dr$. We conclude that $X(b) = X(b')$. Therefore, for all $q \geq b$, $X(q) = X'(q)$ as $x(q) = x'(q)$ for all such q. Thus, $X(q)$ and $X'(q)$ are identical on $[0, a]$ and $[b, 1]$. Of course $x'(q)$ is a constant function on $[a, b]$ so therefore its integral is a linear function; therefore, it must be the linear function that connects $(a, X(a))$ to $(b, X(b))$ with a straight line. See Figure 8.3(b).

Notice third, our choice of interval can be arbitrary and will always simply replace an interval of $X(q)$ with a straight line. Let $\bar{X}(\cdot)$ be the smallest concave function that upper bounds $X(\cdot)$. Let k be the number of contiguous subintervals of $[0, 1]$ for which $\bar{X}(q) \neq X(q)$ and let I_j be the jth interval. Let $\mathcal{I} = (I_1, \ldots, I_k)$. The following resampling procedure implements cumulative allocation rule $\bar{X}(\cdot)$. If $q \in I_j \in \mathcal{I}$ then resample $\bar{q} \sim U[I_j]$ (the uniform distribution on I_j), otherwise set $\bar{q} = q$. This is implemented by running the algorithm on the value corresponding to $\bar{q}$, i.e, $\bar{v} = v(\bar{q})$. The resulting allocation rule $\bar{x}(q)$ is $\frac{d\bar{X}(q)}{dq}$ which, by the convexity of $\bar{X}(\cdot)$, is monotone. See Figure 8.4.

Surplus Preservation

Notice that $\bar{X}(\cdot)$, as the smallest concave function that upper bounds $X(\cdot)$, satisfies $\bar{X}(q) \geq X(q)$. This dominance has the following interpretation: higher values have receive higher service probability. We formalize this argument in the following lemma.

Lemma 8.14. *For $x(\cdot)$ and $\bar{x}(\cdot)$ (as defined in the construction above),* $\mathbf{E}[v(q)\bar{x}(q)] \geq \mathbf{E}[v(q)x(q)]$.



(a) allocation rule $\bar{x}(q)$.   (b) cumulative allocation rule $\bar{X}(q)$.

Figure 8.4: The allocation rule $\bar{x}(q)$ (solid) and cumulative allocation rule $\bar{X}(q)$ (solid) constructed by taking the convex hull of $X(q)$ (dashed). Interval $I = [a, b]$ is defined as $\{q : \bar{X}(q) \neq X(q)\}$.

Proof. We show that $\mathbf{E}[v(q)(\bar{x}(q) - x(q))]$ is non-negative.

$$\begin{aligned}\mathbf{E}[v(q)(\bar{x}(q) - x(q))] &= \int_0^q v(q)(\bar{x}(q) - x(q))\,dq \\ &= \Big[v(q)(\bar{X}(q) - X(q))\Big]_0^1 - \int_0^1 v'(q)(\bar{X}(q) - X(q))\,dq.\end{aligned}$$

The second line follows from integrating by parts. Of course: $v(\cdot)$ is decreasing so $v'(\cdot)$ is non-positive, $\bar{X}(q) = X(q)$ for $q \in \{0, 1\}$, and $\bar{X}(q) - X(q)$ is non-negative; therefore, in the second line above the first term is zero and the second term is non-negative. ☐

Reduction

The general reduction from BIC mechanism design to algorithm design is the following.

Mechanism 8.2. *Construct the BIC mechanism $\bar{\mathcal{A}}$ from $\mathcal{A}$ as follows.*

1. *For each agent i, identify intervals of non-monotonicity $\mathcal{I}_i$ by taking the convex hull of the cumulative allocation rule (in quantile space).*
2. *For each agent i, if $v_i \in I \in \mathcal{I}_i$ resample $\bar{v}_i \sim F_i[I]$ otherwise set $\bar{v}_i \leftarrow v_i$. (Here $F_i[I]$ denotes the conditional distribution of $v_i \in I$ for F_i.)*
3. $\bar{\mathbf{x}} \leftarrow \mathcal{A}(\bar{\mathbf{v}})$.
4. $\bar{\mathbf{p}} \leftarrow$ *payments from payment identity for* $\bar{\mathbf{x}}(\cdot)$.

This mechanism satisfies our requirements of monotonicity, surplus preservation, and locality. These lemmas follow directly from the construction and we will not provide further proof. Theorem 8.13 directly follows.

Lemma 8.15. *The construction of $\bar{\mathcal{A}}$ from $\mathcal{A}$ is monotone.*

Lemma 8.16. *The construction of $\bar{\mathcal{A}}$ from $\mathcal{A}$ is local.*

Lemma 8.17. *The construction of $\bar{\mathcal{A}}$ from $\mathcal{A}$ is surplus preserving.*

8.3.2 Blackbox Computation

It should be immediately clear that the reduction given in the preceding section relies on incredibly strong assumptions about our ability to obtain closed form expressions for the allocation rule of the algorithm and perform calculus on these expressions. In fact theoretical analysis of the statistical properties of algorithms on random instances is exceptionally difficult and this is one of the reasons that theoretical analysis of algorithms is almost entirely done in the worst case. Therefore, it is unlikely these assumptions hold in practice.

Suppose instead we cannot do such a theoretical analysis but we can make blackbox queries to the algorithm and we can sample from the distribution. While we omit all the details from this text, it is possible get accurate enough estimates of the allocation rule that the aforementioned resampling procedure can be approximated arbitrarily precisely. Because this approach is statistical, it may fail to result in absolute monotonicity. However, we can take the convex combination of the resulting allocation rule with a blatantly monotone one to fix these potentially small non-monotonicities.

Naturally, such an approach may lose surplus over the original algorithm because if the small errors it makes. Nonetheless, we can make this loss arbitrarily small. For convenience in expressing the theorem the valuation distribution is normalized over $[0, h]$. The following theorem results.

Theorem 8.18. *For any n-agent single-dimensional agent environment, any product distribution $\mathbf{F}$ over $[0, h]^n$, any algorithm $\mathcal{A}$, and any ϵ, there is a BIC mechanism $\bar{\mathcal{A}}_\epsilon$ satisfying $\mathbf{E}_{\mathbf{v}\sim\mathbf{F}}\big[\bar{\mathcal{A}}_\epsilon(\mathbf{v})\big] \geq \mathbf{E}_{\mathbf{v}\sim\mathbf{F}}[\mathcal{A}(\mathbf{v})] - \epsilon$. Furthermore if $\mathcal{A}$ is polynomial time in n, then $\bar{\mathcal{A}}_\epsilon$ polynomial time in n, $1/\epsilon$, and $\log h$.*

While this construction seems to be a great success, it is important to note where it fails. Bayesian incentive compatibility is a weaker notion of incentive compatibility than dominant strategy incentive compatibility. Notably, the construction only gives a monotone allocation rule in expectation when other agents' values are drawn from the distribution. It is not, therefore, a dominant strategy for the agents to bid truthfully. So while we have verified that BIC mechanism design is computationally equivalent to Bayesian algorithm design. Are these both computationally equivalent to DSIC mechanism design? This question has been partially answered in the negative; the conclusion being that there is loss in restricting attention to dominant strategy mechanisms in computationally bounded environments.

8.3.3 Payment Computation

Recall that the payment identity requires that a monotone allocation rule $x_i(v_i)$ be accompanied by a payment rule $p_i(v_i) = v_i x_i(v_i) - \int_0^{v_i} x_i(z)dz$. At first glance, this appears to require having access to the functional form of the allocation rule. Again, such a requirement is unlikely to be satisfied. This problem, however, is much easier than the monotonization discussed in the previous section because the payment rule only must satisfy the payment identity in expectation. We show how to do this with only two blackbox calls to the algorithm.

We compute a random variable P_i with expectation $\mathbf{E}[P_i] = p_i(v_i)$. We will do this for the two parts of the payment identity separately.

Algorithm 8.3. *The* blackbox payment algorithm *for $\mathcal{A}$ computes payment P_i for agent i with value v_i as follows:*

1. $X_i \leftarrow \begin{cases} 1 & \text{if } i \text{ wins in } x_i(v_i) \\ 0 & \text{otherwise.} \end{cases}$

   I.e., X_i is an indicator variable *for whether i wins or not when running $\mathcal{A}(\mathbf{v})$.*

2. $Z_i \sim U[0, v_i]$ *(drawn at random).*

3. $Y_i \leftarrow \begin{cases} 1 & \text{if } i \text{ wins in } x_i(Z_i) \\ 0 & \text{otherwise.} \end{cases}$

   I.e., Y_i is an indicator variable for whether i would win or not when simulating the algorithm on $\mathcal{A}(\mathbf{v}_{-i}, Z_i)$.

4. $P_i \leftarrow v_i(X_i - Y_i)$.

We first note that this payment rule is *individually rational* in the following strong sense. For any realization $\mathbf{v}$ of agent values and any randomization in the algorithm and payment computation, the utilities of all agents are non-negative. This is clear because the utility of an agent is her value minus her payment, i.e., $v_iX_i - P_i = v_iY_i$. Since $Y_i \in \{0, 1\}$, this utility is always non-negative. Oddly, this payment rule may result in a losing agent being paid, i.e., there may be *positive transfers*. This is because the random variables X_i and Y_i are independent. We may instantiate $X_i = 0$ and $Y_i = 1$. Agent i then loses and has a payment of $-v_i$, i.e., i is paid v_i.

Lemma 8.19. *The blackbox payment computation algorithm satisfies $\mathbf{E}[P_i] = p_i(v_i)$.*

Proof. This follows from linearity of expectation and the definition of expectation in the following routine calculation. First calculate $\mathbf{E}[Y_i]$ notating the probability density function for Z_i is $f_{Z_i}(z) = 1/v_i$ for $Z_i \sim U[0, v_i]$.

$$\begin{aligned}
\mathbf{E}[Y_i] &= \int_0^{v_i} x_i(z) f_{Z_i}(z) dz \\
&= \tfrac{1}{v_i} \int_0^{v_i} x_i(z) dz.
\end{aligned}$$

Now we calculate $\mathbf{E}[P_i]$ as,

$$\begin{aligned}\mathbf{E}[P_i] &= \mathbf{E}[v_i X_i] - \mathbf{E}[v_i Y_i] \\ &= v_i x_i(v_i) - v_i \mathbf{E}[Y_i] \\ &= v_i x_i(v_i) - \int_0^{v_i} x_i(z)dz. \end{aligned}$$ □

This construction was slightly odd because a loser may be paid a positive transfer. This can be partially addressed. There is a slightly more complicated construction wherein all losers have payments identically equal to zero, but winners may be paid to receive service. We leave the construction as an exercise.

8.4 Computational Overhead of Payments

The focus of this chapter has been on ascertaining the extent to which mechanism design is, computationally, more difficult than algorithm design. Positive results, such as our reduction from BIC mechanism design to algorithm design enable designers a generic approach for the computer implementation of a mechanism.

We conclude this chapter with a esoteric-seeming question that has important practical consequences. Notice that when turning a monotone algorithm, either by externality payments $p_i = \mathrm{OPT}(\mathbf{v}_{-i}) - \mathrm{OPT}_{-i}(\mathbf{v})$ or by our blackbox payment calculation of random variable P_i, the number of calls to the algorithm is $n + 1$. One call to compute the outcome and an additional call for each agent to compute that agent's payment. One question is then, from a computational point of view, whether n times more computation must be performed to run a mechanism than it takes to just compute its allocation.

A more practically relevant viewpoint on this question is whether repeatability of the algorithm is necessary. We know from our BIC characterization that any mechanism must be monotone. However, approaches described thus far for calculating payment have required that the algorithm be repeatable by simulation as well.

This repeatability requirement poses severe challenges in some practical contexts. Consider an online advertising problem where there is a set of advertisers (agents) who each have an ad to be shown on an Internet web page. An advertiser is "served" if her ad is clicked on. Each advertiser i has a private value v_i for each click she receives. Each ad i has a *click-through rate* c_i, i.e., a probability that the ad will be clicked if it is shown. If the mechanism designer knew these click-through rates in advance, the surplus maximizing rule would be to show the advertiser with the highest $v_i c_i$. Unfortunately, these click-through rates are often unknown to the mechanism designer and the advertisers. The mechanism can attempt to use any of a number of learning algorithm to learn advertiser click-through rates as it shows ads. Many of these learning algorithms are in fact monotone, meaning the higher i's value v_i the more clicks i will receive in expectation. Unfortunately it is difficult to turn these learning algorithms into incentive compatible mechanisms because there is no way to go back in time and see what would have happened if a different advertiser had been shown. What is needed here is a way to design a mechanism from a monotone algorithm with only a single call to the algorithm.

8.4.1 Communication Complexity Lower Bound

For single-dimensional agent problems with special structure (i.e., on the cost function of the designer, $c(\cdot)$) it is possible to design an algorithm that computes payments at the same time as it computes the allocation with no significant extra computational effort. For instance, for optimizations based on linear programming duality, a topic we will not cover here, the *dual variables* are often exactly the payments required for the surplus maximization mechanism.

It is likely that such a result does not hold generally. However, the conclusion of our discussion of computational tractability earlier in this chapter was that proving lower bounds on computational requirements is exceptionally difficult. We therefore analyze a related question, namely the *communication complexity* of an allocation rule versus its associated payment rules. To analyze communication complexity we imagine that each of our n agents has a private input and the agents collectively want to compute some desired function. Each agent may perform an unlimited amount of local computation and can broadcast any information simultaneously to all other agents. The challenge is to come up with a protocol that the agents can follow so that at the end of the protocol all agents know the value of the function and the total number of bits broadcast is minimized. In this exercise, agents are assumed to be non-strategic and will follow the chosen protocol precisely.

As an example consider a *public good* with cost C. In this setting we must either serve all agents or none of them. If we serve all of them we incur the cost of C. This is a single-dimensional agent environment with cost function given by

$$c(\mathbf{x}) = \begin{cases} C & \text{if } \sum_i x_i = n \\ 0 & \text{if } \sum_i x_i = 0 \\ \infty & \text{otherwise.} \end{cases}$$

Notice that it is infeasible to serve one agent and not another. This problem arises naturally. Assume the government is considering whether or not to build a bridge. It costs C to build the bridge. Naturally the government only wants to build the bridge if the total value of the bridge to the people exceeds the cost. Of course, if the bridge is built then everyone can use it. For the objective of social surplus, clearly we wish to serve the agents whenever $\sum_i v_i \geq C$.

Consider the special case of the above problem with two agents, each with an integral value ranging between 0 and C. Let $k = \log C$ be the number of bits necessary to represent each agent's value in binary. To compute the surplus maximizing outcome, i.e., whether to allocate to both agents or neither of them, the following protocol can be employed:

1. Agent 1 broadcasts her k bit value v_1.

2. Agent 2 determines whether $v_1 + v_2 \geq C$ and broadcasts 1 if it is and 0 otherwise.

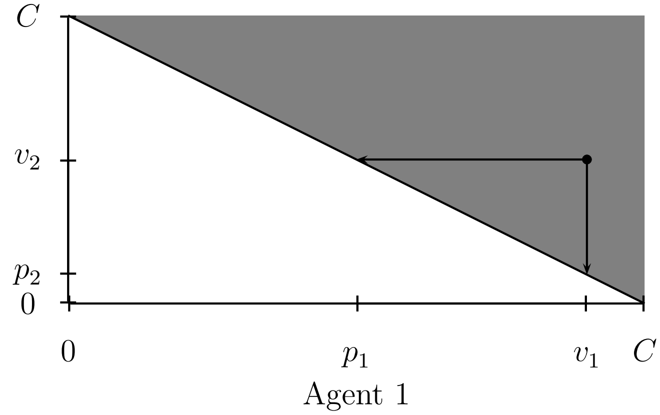


Figure 8.5: As a function of agent 1's value (x-axis) and agent 2's value (y-axis) and the the region of allocation (gray) of the surplus maximizing mechanism for the public project with cost C is depicted. Valuation profile $\mathbf{v} = (v_1, v_2)$ is depicted by a point and the payments $\mathbf{p} = (p_1, p_2)$ can be calculated by following the accompanying arrows.

Notice that the total number of bits broadcast by this protocol is $k + 1$ and both parties learned the desired outcome.

Now suppose we also wish the communication protocol to compute the incentive compatible payments for this monotone allocation rule wherein both agents must learn p_1 and p_2. How many bits of communication are required? Notice that in the case that we serve the agents, $p_1 = C - v_2$ and $p_2 = C - v_1$. Importantly, there is a unique $\mathbf{p}$ for each unique $\mathbf{v}$ that is served. There are $C^2/2$ such payment vectors in total. The broadcast bits must uniquely determine which of these $\mathbf{p}$ is the correct outcome. Given such a large number of payment vectors the most succinct representation would be to number them and write the index of the desired payment vector in binary. This takes a number of bits that is logarithmic in the number of payment vectors possible. In our case this is $\log(C^2/2)$. Of course, $C = 2^k$ so the number of bits is $2k-1$. Agent 1 has k bits, but the other $k-1$ bits should be communicated from Agent 2, and vice versa. Therefore a total of $2k - 2$ bits must be communicated for both agents to learn $\mathbf{p}$.

We conclude that in this two-agent environment about twice as many bits are required to compute payments than to compute the outcome. We summarize this in the following lemma.

Lemma 8.20. *There exists a two-agent single-dimensional agent environment where the communication complexity of computing payments is twice that of computing the allocation.*

The above two agent example can be generalized to an n-agent environment where the communication complexity of payments is n times more than that for computing the allocation. The single-dimensional agent problem that exhibits such a separation is contrived, i.e., there is no natural economic interpretation for it. The real challenge in this generalization is in determining an environment where the allocation can be computed with very low communication. We omit the proof and construction.

Theorem 8.21. *There exists an n-agent single-dimensional agent environment where the communication complexity of computing payments is n times that of computing the allocation.*

The conclusion of the above discussion on the communication complexity of computing payments is that there are environments where payments are a linear factor harder to compute. If we wish to take any algorithm and construct payments we can expect that the construction, in worst-case, will require a linear factor more work, for example, by invoking the algorithm $n + 1$ times as is done by the payment algorithms previously discussed. Notice, however, that the above lower bound leaves open the possibility that subtle changes to the allocation rule might permit payments to be calculated without such computational overhead.

8.4.2 Implicit Payments

The main result of this section is to describe a procedure for taking any monotone algorithm and altering it in a way that does not significantly decrease its surplus and enables the outcome and all payments to be computed implicitly from a single invocation of the algorithm.

The presentation in this text will focus on the Bayesian setting as the approach and result are the most relevant for application to any algorithm. (There is a similar approach, that we do not discuss, for achieving the same flavor of result without a prior distribution.) Therefore, assume as we did in preceding sections that agents' values are distributed according to a product distribution that is continuous on its support, i.e., each agent's density function is strictly positive.

Two main ideas underlie this approach. Recall that the blackbox payment computation drew $Z_i \sim U[0, v_i]$ and defined Y_i as an indicator for $x_i(Z_i)$. The expected value of this Y_i is then related to $\int_0^{v_i} x_i(z)dz$. The first idea is that there is nothing special about the uniform distribution; we can do the exact same estimation with any continuous distribution, e.g., with F_i. The second idea is that if with some small probability ϵ we redraw $V_i' \sim F_i$ and input that into the algorithm instead of v_i then this changes the allocation rule in an easy to describe way. It multiplies it by $(1 - \epsilon)$ and adds a constant $\mathbf{E}\big[x_i(V_i')\big]$. For the purpose of computing payments, which is a function of the integral of the allocation rule, adding a constant to it has no effect.

Notice in the definition of the implicit payment mechanism, below, that the algorithm $\mathcal{A}$ is only called once.

Mechanism 8.3. *For algorithm $\mathcal{A}$, distribution $\mathbf{F}$, and parameter $\epsilon > 0$, the* implicit payment mechanism *$\mathcal{A}'_\epsilon$ is*

1. *for all i:*
   (a) *Draw $Z \sim F_i$*
   (b) $V_i' \leftarrow \begin{cases} v_i & \textit{with probability } 1 - \epsilon \\ Z & \textit{otherwise.} \end{cases}$

2. $\mathbf{X}' \leftarrow \mathcal{A}(\mathbf{V}')$.

3. for all i: $P'_i \leftarrow \begin{cases} v_i X'_i & \textit{if } V'_i = v_i \\ -\frac{1-\epsilon}{\epsilon} \cdot \frac{X'_i}{f_i(V'_i)} & \textit{if } V'_i < v_i \\ 0 & \textit{otherwise.} \end{cases}$

As you can see, the implicit payment mechanism sometimes calls $\mathcal{A}$ on v_i into and sometimes it draws a completely new value V'_i and inputs that to $\mathcal{A}$ instead. The actual payments are a bit strange. If it inputs i's original value and serves i then i is charged her value. If it inputs a value less than i's original value and serves i then i is given a very large rebate, as ϵ and $f(V'_i)$ should be considered very small. If we do not serve i then i pays nothing. Furthermore, if the implicit payment mechanism inputs a value greater than i's value and allocate to i then i also pays nothing.

Such a strange payment computation warrants a discussion as to the point of this exercise in the first place; it seems like such a crazy payment rule would completely impractical. Recall, though, that our search for a BIC mechanism was one of existence. We wanted to know if there existed a mechanism in our model (in this case, with a single call to the algorithm) with good BNE. We have verified that. In fact, BIC mechanisms are often not practical. The omitted step is in finding a practical mechanism with good BNE, and this means undoing the revelation principle to ask what other more natural mechanism might have the same BNE but also satisfy whatever additional practicality constraints we may require. This final step of mechanism design is often overlooked, and it is because designing non-BIC mechanisms with good properties is difficult.

With such practical considerations aside, we now prove that monotonicity of $\mathcal{A}$ (in a Bayesian sense) is enough to ensure that $\mathcal{A}'_\epsilon$ is BIC. Furthermore, we show that the expected surplus of $\mathcal{A}'_\epsilon$ is close to that of $\mathcal{A}$.

Lemma 8.22. *For* $\mathbf{v} \sim \mathbf{F}$*, if the allocation rule* $x_i(\cdot)$ *for* $\mathcal{A}$ *is monotone, then the allocation rule* $x'_i(\cdot)$ *for* $\mathcal{A}'_\epsilon$ *is monotone.*

Proof. From each agent i's perspective we have not changed the distribution of other agents' values. Furthermore, $\mathbf{x}'(\mathbf{v}) = (1-\epsilon)\cdot\mathbf{x}(\mathbf{v}) + \epsilon\cdot\mathbf{E}_{\mathbf{V}'}\big[\mathbf{x}(\mathbf{V}')\big]$, i.e., we have scaled the algorithms allocation rule by $(1-\epsilon)$ and added a constant. Therefore, relative to prior distribution F, if $\mathcal{A}$ was monotone before then $\mathcal{A}'_\epsilon$ is monotone now. (Furthermore, if $\mathcal{A}$ was monotone in an dominant strategy sense before then the implicit payment mechanism is monotone in an dominant strategy sense now.) ∎

Lemma 8.23. *For agent i with value* v_i*,* $\mathbf{E}\big[P'_i\big] = p'_i(v_i)$ *satisfying the payment identity.*

Proof. Since our allocation rule for agent i is $x'_i(v_i) = (1-\epsilon)\cdot x_i(v_i) + \epsilon \cdot \mathbf{E}\big[x_i(V'_i)\big]$ where the final term is a constant. We must show that $\mathbf{E}[P_i] = (1-\epsilon)p_i(v_i) = p'_i(v_i)$.

Define indicator variable A for the event that $V'_i = v_i$ and B for the event $V'_i < v_i$. With these definitions, $P_i = v_i X'_i A - \frac{1-\epsilon}{\epsilon} \cdot \frac{X'_i B}{f(V'_i)}$. The expectation of the first term is easy to analyze:

$$\begin{aligned}\mathbf{E}\big[v_i X_i' A\big] &= \mathbf{Pr}[A=1]\cdot \mathbf{E}\big[v_i X_i' \mid A=1\big]\\ &= (1-\epsilon) v_i x_i(v_i).\end{aligned}$$

This is exactly as desired. Now we turn to the second term. For convenience we ignore the constants until the end.

$$\begin{aligned}\mathbf{E}\big[X_i' B / f_i(V_i')\big] &= \mathbf{Pr}[B=1]\cdot \mathbf{E}\big[X_i'/f_i(V_i') \mid B=1\big]\\ &= \epsilon F_i(v_i)\cdot \mathbf{E}\big[X_i'/f_i(V_i') \mid B=1\big].\end{aligned}$$

Notice $B=1$ implies that we drawn $V_i' \sim F_i[0, v_i]$ which has density function $f_i(z)/F_i(v_i)$. Thus, we continue our calculation as,

$$\begin{aligned}&= \epsilon F_i(v_i)\cdot \int_0^{v_i} \frac{x_i(z)}{f_i(z)}\cdot\frac{f_i(z)}{F_i(v_i)} dz\\ &= \epsilon \int_0^{v_i} x_i(z) dz.\end{aligned}$$

Combining this with the constant multiplier $\frac{1-\epsilon}{\epsilon}$ and the calculation of the first term, we conclude that $\mathbf{E}[P_i] = (1-\epsilon)p_i(v_i) = p_i'(v_i)$ as desired. □

From the two lemmas above we conclude that $\mathcal{A}_\epsilon'$ is BIC. To discuss the performance of $\mathcal{A}_\epsilon'$ (i.e., surplus) we consider two of our general single-dimensional agent environments separately. We consider the general costs environment because it is the most general and the general feasibility environment because it is easier and thus permits a nicer performance bound.

Lemma 8.24. *For any distribution* $\mathbf{F}$ *and any general feasibility setting,* $\mathbf{E}\big[\mathcal{A}_\epsilon'(\mathbf{v})\big] \geq (1-\epsilon)\cdot\mathbf{E}[\mathcal{A}(\mathbf{v})]$.

Proof. This follows from considering each agent separately and using linearity of expectation. Our expected surplus from i is

$$\begin{aligned}\mathbf{E}_{v_i}\big[v_i x_i'(v_i)\big] &= (1-\epsilon)\cdot \mathbf{E}_{v_i}[v_i x_i(v_i)] + \epsilon\cdot\mathbf{E}_{v_i}[v_i]\cdot\mathbf{E}_{v_i}[x_i(v_i)]\\ &\geq (1-\epsilon)\cdot\mathbf{E}_{v_i}[v_i x_i(v_i)] \end{aligned} \tag{8.6}$$

The final step follows because the second term on the right-hand side is always non-negative and thus can be dropped. □

Lemma 8.25. *For any distribution* $\mathbf{F}$ *and any* n*-agent general costs setting,* $\mathbf{E}\big[\mathcal{A}_\epsilon'(\mathbf{v})\big] \geq \mathbf{E}[\mathcal{A}(\mathbf{v})] - \epsilon h n$ *where* h *is an upper bound on any agent's value.*

Proof. This proof follows by noting that the algorithm always runs on an input that is random from the given distribution $\mathbf{F}$. Therefore, the expected costs are the same, i.e., $\mathbf{E}_{\mathbf{v}}\big[c(\mathbf{x}'(\mathbf{v}))\big] = \mathbf{E}_{\mathbf{v}}[c(\mathbf{x}(\mathbf{v}))]$. However, the expected total value to the agents is decreased relative to the original algorithm. Our expected surplus from i is

$$\begin{aligned}\mathbf{E}_{v_i}\big[v_i x_i'(v_i)\big] &\geq (1-\epsilon)\cdot \mathbf{E}_{v_i}[v_i x_i(v_i)] \\ &\geq (1-\epsilon)\cdot \mathbf{E}_{v_i}[v_i x_i(v_i)] + \epsilon \cdot \mathbf{E}_{v_i}[v_i x_i(v_i)] - \epsilon h \\ &= \mathbf{E}_{v_i}[v_i x_i(v_i)] - \epsilon h\end{aligned}$$

The first step follows from equation (8.6), we can then add the two terms on the right-hand side because the negative one is higher magnitude than the positive one. The final result follows from summing over all agents and the expect costs. □

The following theorems follow directly from the lemmas above.

Theorem 8.26. *For any single-dimensional agent environment with general costs, any product distribution* $\mathbf{F}$ *over* $[0,h]^n$*, any Bayesian monotone algorithm* $\mathcal{A}$*, and any* $\epsilon > 0$*, the implicit payment mechanism* $\mathcal{A}'_\epsilon$ *is BIC and satisfies* $\mathbf{E}_{\mathbf{v}\sim\mathbf{F}}\big[\mathcal{A}'_\epsilon(\mathbf{v})\big] \geq \mathbf{E}_{\mathbf{v}\sim\mathbf{F}}[\mathcal{A}(\mathbf{v})] - \epsilon h n$.

Theorem 8.27. *For any single-dimensional agent environment with general feasibility constraints, any product distribution* $\mathbf{F}$*, any Bayesian monotone algorithm* $\mathcal{A}$*, and any* $\epsilon > 0$*, the implicit payment mechanism* $\mathcal{A}'_\epsilon$ *is BIC and satisfies* $\mathbf{E}_{\mathbf{v}\sim\mathbf{F}}\big[\mathcal{A}'_\epsilon(\mathbf{v})\big] \geq (1-\epsilon)\mathbf{E}_{\mathbf{v}\sim\mathbf{F}}[\mathcal{A}(\mathbf{v})]$.

We conclude this section by answering our opening question. If we are willing to sacrifice an arbitrarily small amount of the surplus, we do not require algorithms to be repeatable to turn them into mechanisms. Our only requirement is monotonicity.

Exercises

8.1 Section 8.3 we showed that we can turn a non-monotone algorithm for any single-dimensional agent environment into a BIC mechanism that has at least the same expected social surplus. Consider another important objective in computer systems: *makespan*. Suppose we have n machines and m jobs. Each machine i is a selfish agent with privately known slowness-parameter $|v_i|$ and each job j has an intrinsic length w_j. The time it takes i to perform job j is $|v_i w_j|$ and we view this as a cost to agent i.[2] The load on machine i for a set of jobs J is $|v_i| \sum_{j\in J} w_j$. A scheduling is a partitioning of jobs among the machines. Let J_i be the jobs assigned to machine i. The makespan of this scheduling is the maximum load of any machine, i.e., $\max_i |v_i| \sum_{j\in J_i} w_j$. The goal of a scheduling algorithm is to minimize the makespan. (Makespan is important outside of computer systems as "minimizing the maximum load" is related to fairness.)

[2] We are putting these quantities in absolute values because if the private value represents a cost, it is most consistent with the course notation to view v_i as negative.

This is a single-dimensional agent environment, though the outcome for each agent i is not a binary $x_i \in \{0, 1\}$. Instead if i is allocated jobs J_i then $x_i = \sum_{j \in J_i} w_j$. Of course $x_i(v_i)$ is, as usual, $\mathbf{E}_{\mathbf{v} \sim \mathbf{F}}[x_i(\mathbf{v}) \mid v_i]$. The agent's cost for such an outcome is $|v_i x_i(v_i)|$.

Show that the method of Mechanism 8.2 for monotonizing a non-monotone algorithm fails to preserve the expected makespan. (Hint: All you need to do is come up with an instance and a non-monotone algorithm for which expected makespan increases when we apply the transformation. Do not worry about the values being negative nor about the allocation being non-binary; these aspects of the problem do not drive the non-monotonicity result.)

8.2 The blackbox payment algorithm (Algorithm 8.3) sometimes makes positive transfers to agents. Give a different algorithm for calculating payments with the correct expectation where (a) winners never pay more than their value and (b) the payment to (and from) losers is always identically zero. The number of calls your payment algorithm makes to the allocation algorithm should be at most a constant (in expectation).

8.3 Dominant strategy incentive compatible mechanisms can be combined.

(a) Consider the following algorithm:

- Simulate greedy by value (i.e., sorting by v_i).
- Simulate greedy by value-per-item (i.e., sorting by $v_i/|S_i|$).
- Output whichever solution has higher surplus.

Prove that the resulting algorithm is monotone.

(b) We say a deterministic algorithm is *independent of irrelevant alternatives* when

$$\mathbf{x}(\mathbf{v}_{-i}, v_i) = \mathbf{x}(\mathbf{v}_{-i}, v_i') \qquad \text{iff} \qquad x_i(\mathbf{v}_{-i}, v_i) = x_i(\mathbf{v}_{-i}, v_i'),$$

i.e., the outcome is invariant on the value of a winner or loser. Prove that the algorithm $\mathcal{A}$ that runs k deterministic monotone independent-of-irrelevant-alternatives algorithms $\mathcal{A}_1 \ldots, \mathcal{A}_k$ and then outputs the solution of the one with the highest surplus is itself monotone.

8.4 Consider the following knapsack problem: each agent has a private value v_i for having an object with publicly known size w_i inserted into a knapsack. The knapsack has capacity C. Any set of agents can be served if all of their objects fit simultaneously in the knapsack. We denote an instance of this problem by the tuple $(\mathbf{v}, \mathbf{w}, C)$. Notice that this is a single-dimensional agent environment with cost function:

$$c(\mathbf{x}) = \begin{cases} 0 & \text{if } \sum_i x_i w_i \leq C \\ \infty & \text{otherwise.} \end{cases}$$

The knapsack problem is $\mathcal{NP}$-complete; however, very good approximation algorithms exist. In fact there is a *polynomial time approximation scheme* (PTAS). A PTAS is an family of algorithms parameterized by $\epsilon > 0$ where $\mathcal{A}_\epsilon$ is a $(1+\epsilon)$-approximation runs in polynomial time in n, the number of agents, and $1/\epsilon$.

PTASs are often constructed from pseudo-polynomial time algorithms. Let V be an upper bound on the value of any agent, i.e., $V \geq v_i$ for all i, and assume all agent values are integers. For the integer-valued knapsack problem there is an algorithm with runtime $O(n^2 V)$. (Constructing this algorithm is non-trivial. Try to do it on your own, or look it up in an algorithms text book.) Notice that this is not fully polynomial time as V is a number that is part of the input to the algorithm. The "size" of V is the amount of space it takes to write it down. We ordinarily write numbers on a computer in binary (or by hand, in decimal) which therefore has size $\log V$. An algorithm with runtime polynomial in V is exponential in $\log V$ and, therefore, not considered polynomial time. It is *pseudo-polynomial time*.

A pseudo-polynomial time algorithm, $\mathcal{A}$, for surplus maximization in the integer values setting can be turned into a PTAS $\mathcal{A}_\epsilon$ for the general values setting by rounding. The construction:

- $v'_i \leftarrow v_i$ rounded up to the nearest multiple of $V\epsilon/n$.
- $v''_i \leftarrow v'_i n/(V\epsilon)$, an integer between 0 and n/ϵ.
- Simulate $\mathcal{A}(\mathbf{v}'', \mathbf{w}, C)$, the integer-valued pseudo-polynomial time algorithm.
- Simulate the algorithm that allocates only to the highest valued agent.
- Output whichever solution has higher surplus.

Notice the following about this algorithm. First, its runtime is $O(n^3/\epsilon)$ when applied to the $O(n^2 V)$ pseudo-polynomial time algorithm discussed above. Second, it is a $(1+\epsilon)$-approximation if we set $V = \max_i v_i$. (This second observation involves a several line argument. Work it out yourselves or look it up in an algorithms text book.)

In this question we will investigate the incentives of this problem which arise because we do not know a good choice of V in advance.

(a) Suppose we are given some V. Prove that $\mathcal{A}_\epsilon$, for any ϵ, is monotone.

(b) Suppose we do not know V in advance. A logical choice would be to choose $V = \max_i v_i$ and then run $\mathcal{A}_\epsilon$. Prove that this combined algorithm is not monotone.

(c) For any given ϵ, derive an DSIC $(1+\epsilon)$-approximation mechanism from any integer-valued pseudo-polynomial time algorithm. Your algorithm should run in polynomial time in n and $1/\epsilon$. (Hint: The hard part is not knowing V.)

Chapter Notes

Richard Karp (1972) pioneered the use of $\mathcal{NP}$-completeness reductions to show that a number of relevant combinatorial optimization problems, including set packing, are $\mathcal{NP}$-complete. Lehmann et al. (2002) and Nisan and Ronen (2001) introduced the concept of computationally tractable approximation mechanisms. The single-minded combinatorial auction problem is an iconic single-dimensional agent mechanism design problem.

There are several reductions from approximation mechanism design to approximation algorithm design (in single-dimensional settings). Briest et al. (2005) show that any polynomial time approximation scheme (PTAS) can be converted into an (dominant strategy) incentive compatible mechanisms. The reduction from BIC mechanism design to Bayesian algorithm design that was described in this text was given by Hartline and Lucier (2010); Hartline et al. (2011) improve on the basic approach. For the special case of single-minded (and a generalization that is not strictly single-dimensional) combinatorial auctions Babaioff et al. (2009a) give a general reduction that obtains a $O(c \log h)$ approximation mechanism from any c-approximation algorithm where h is an upper-bound on the value of any agent.

Several of the above results have extensions to environments with multi-dimensional agent preferences. Specifically, Dughmi and Roughgarden (2010) show that in multi-dimensional additive-value packing environments any PTAS algorithm can be converted into a PTAS mechanism. The result is via a connection between smoothed analysis and PTASes: A PTAS can be viewed as optimally solving a perturbed problem instance and any optimal algorithm is incentive compatible via the standard approach. Hartline et al. (2011) and Bei and Huang (2011) generalized the construction of Hartline and Lucier (2010) to reduce BIC mechanism design to algorithm design in general multi-dimensional environments. Notably, the approach is brute-force in the type space of each agent.

The unbiased estimator payment computation was given by Archer et al. (2003). The communication complexity lower bound for computing payments is given by Babaioff et al. (2008). Babaioff et al. (2010) developed the approximation technique that permits payments to be computed with one call to the allocation rule (i.e., the algorithm). This result enables good mechanisms in environments where the algorithm cannot be repeated, for instance, in online auction settings such as the one independently considered by Babaioff et al. (2009b) and Devanur and Kakade (2009).

Appendix A

Mathematical Reference

Contained herein is reference to mathematical notations and conventions used throughout the text.

A.1 Big-oh Notation

We give asymptotic bounds using big-oh notation. Upper bounds are given with O, strict upper bounds are given with o, lower bounds are given with Ω, strict lower bounds are given with ω, and exact bounds are given with Θ. Formal definitions are given as follows:

Definition A.1. *Function $f(n)$ is $O(g(n))$ if there exists a $c > 0$ and $n_0 > 0$ such that*

$$\forall n > n_0,\ \ f(n) \leq c\, g(n).$$

Definition A.2. *Function $f(n)$ is $\Omega(g(n))$ if there exists a $c > 0$ and $n_0 > 0$ such that*

$$\forall n > n_0,\ \ f(n) \geq c\, g(n).$$

Definition A.3. *Function $f(n)$ is $\Theta(g(n))$ if it is $O(g(n))$ and $\Omega(g(n))$.*

Definition A.4. *Function $f(n)$ is $o(g(n))$ if it is $O(g(n))$ but not $\Theta(g(n))$.*

Definition A.5. *Function $f(n)$ is $\omega(g(n))$ if it is $\Omega(g(n))$ but not $\Theta(g(n))$.*

A.2 Common Probability Distributions

Common continuous probability distributions are *uniform* and *exponential*. Continuous distributions can be specified by their *cumulative distribution function*, denoted by F, or its derivative $f = F'$, the *probability density function*.

Definition A.6. *The* uniform distribution *on support $[a, b]$, denoted $U[a, b]$, is defined as having a constant density function $f(z) = 1/(b - a)$ over $[a, b]$.*

For example, the distribution $U[0, 1]$ has distribution $F(z) = z$ and density $f(z) = 1$. The expectation of the uniform distribution on $[a, b]$ is $\frac{a+b}{2}$. The monopoly price for the uniform distribution is $\max(b/2, a)$ (See Definition 4.1).

Definition A.7. *The* exponential distribution *with rate* λ *has distribution* $F(z) = 1 - e^{-\lambda z}$ *and density* $f(z) = \lambda e^{-\lambda z}$. *The support of the exponential distribution is* $[0, \infty)$.

The exponential distribution with rate λ has expectation $1/\lambda$ and monopoly price $1/\lambda$. The exponential distribution has constant *hazard rate* λ.

A.3 Expectation and Order Statistics

The *expectation* of a random variable $v \sim F$ is its "probability weighted average." For continuous random variables this expectation can be calculated as

$$\mathbf{E}[v] = \int_{-\infty}^{\infty} z\, f(z)\, dz. \tag{A.1}$$

For continuous, non-negative random variables this expectation can be reformulated as

$$\mathbf{E}[v] = \int_{0}^{\infty} (1 - F(z))\, dz \tag{A.2}$$

which follows from (A.1) and integration by parts.

An *order statistic* of a set of random variables is the value of the variable that is at a particular rank in the sorted order of the variables. For instance, when a valuation profile $\mathbf{v} = (v_1, \ldots, v_n)$ is drawn from a distribution then the ith largest value, which we have denoted $v_{(i)}$, is an order statistic. A fact that is useful for working out examples with the uniform distribution.

Fact A.8. *In expectation, i.i.d. random variables chosen uniformly from a given interval will evenly divide the interval.*

A.4 Integration by Parts

Integration by parts is the integration analog of the product rule for differentiation. We will denote the derivative of a function $\frac{d}{dz}g(z)$ by $g'(z)$. The product rule for differentiation is:

$$[g(z)\, h(z)]' = g'(z)\, h(z) + g(z)\, h'(z). \tag{A.3}$$

The formula for integration by parts can be derived by integrating both sides of the equation and rearranging.

$$\int g'(z)\, h(z)\, dz = g(z)\, h(z) - \int g(z)\, h'(z)\, dz. \tag{A.4}$$

As an example we will derive (A.2) from (A.1). Plug $g(z) = 1 - F(z)$ and $h(z) = z$ into equation A.4.

$$\begin{aligned}
\mathbf{E}[v] &= \int_0^\infty z\, f(z)\, dz \\
&= -\int_0^\infty h(z)\, g'(z)\, dz \\
&= -\Big[h(z)\, g(z)\Big]_0^\infty + \int_0^\infty h'(z)\, g(z)\, dz \\
&= -\Big[z\,(1 - F(z))\Big]_0^\infty + \int_0^\infty 1\,(1 - F(z))\, dz \\
&= \int_0^\infty (1 - F(z))\, dz.
\end{aligned}$$

The last equality follows because $z\,(1 - F(z))$ is zero at both zero and ∞.

A.5 Hazard Rates

The *hazard rate* of distribution F (with density f) is $h(z) = \frac{f(z)}{1-F(z)}$ (See Definition 4.16). The distribution has a *monotone hazard rate* (MHR) if $h(z)$ is monotone non-decreasing.

A distribution is completely specified by its hazard rate via the following formula.

$$F(z) = 1 - e^{-\int_{-\infty}^{z} h(z)\, dz}.$$

Bibliography

Aggarwal, G., Feder, T., Motwani, R., and Zhu, A. (2004). Algorithms for multi-product pricing. In *Proc. of ICALP*.

Aggarwal, G., Fiat, A., Goldberg, A., Hartline, J., Immorlica, N., and Sudan, M. (2005). Derandomization of Auctions. In *Proc. 37th ACM Symposium on the Theory of Computing*. ACM Press, New York.

Alaei, S. (2011). Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. In *Proc. 52th IEEE Symp. on Foundations of Computer Science*.

Alaei, S., Malekian, A., and Srinivasan, A. (2009). On random sampling auctions for digital goods. In *ACM Conference on Electronic Commerce*, pages 187–196.

Archer, A., Papadimitriou, C., Talwar, K., and Tardos, E. (2003). An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *Proc. 14th ACM Symp. on Discrete Algorithms*. ACM/SIAM.

Archer, A. and Tardos, E. (2001). Truthful mechanisms for one-parameter agents. In *Proc. 42nd IEEE Symp. on Foundations of Computer Science*.

Armstrong, M. (1996). Multiproduct Nonlinear Pricing. *Econometrica*, 64(1):51–75.

Ausubel, L. and Milgrom, P. (2006). The lovely but lonely vickrey auction. In Cramton, P., Shoham, Y., and Steinberg, R., editors, *Combinatorial Auctions*, chapter 1, pages 17–40. MIT Press.

Babaioff, M., Blumrosen, L., Naor, M., and Schapira, M. (2008). Informational overhead of incentive compatibility. In *Proc. 10th ACM Conf. on Electronic Commerce*, pages 88–97.

Babaioff, M., Kleinberg, R., and Slivkins, A. (2010). Truthful mechanisms with implicit payment computation. In *Proc. 12th ACM Conf. on Electronic Commerce*.

Babaioff, M., Lavi, R., and Pavlov, E. (2009a). Single-value combinatorial auctions and algorithmic implementation in undominated strategies. *Journal of the ACM*.

Babaioff, M., Sharma, Y., and Slivkins, A. (2009b). Characterizing truthful multi-armed bandit mechanisms: extended abstract. In *Proc. 11th ACM Conf. on Electronic Commerce*, pages 79–88.

Balcan, M.-F., Blum, A., Hartline, J., and Mansour, Y. (2008). Reducing mechanism design to algorithm design via machine learning. *J. Comput. Syst. Sci.*, 74(8):1245–1270.

Bei, X. and Huang, Z. (2011). Bayesian incentive compatibility via fractional assignments. In *Proc. 22nd ACM Symp. on Discrete Algorithms*.

Briest, P. (2008). Uniform budgets and the envy-free pricing problem. In *ICALP*, pages 808–819.

Briest, P., Chawla, S., Kleinberg, R., and Weinberg, S. (2010). Pricing randomized allocations. In *SODA*, pages 585–597.

Briest, P., Krysta, P., and Vöcking, B. (2005). Approximation techniques for utilitarian mechanism design. In *Proc. 36th ACM Symp. on Theory of Computing*, pages 39–48.

Bulow, J. and Klemperer, P. (1996). Auctions versus negotiations. *American Economic Review*, 86:180–194.

Bulow, J. and Roberts, J. (1989). The simple economics of optimal auctions. *The Journal of Political Economy*, 97:1060–90.

Cai, Y. and Daskalakis, C. (2011). Extreme-value theorems for optimal multidimensional pricing. In *Proc. 52th IEEE Symp. on Foundations of Computer Science*.

Chawla, S., Hartline, J., and Kleinberg, R. (2007). Algorithmic pricing via virtual valuations. In *Proc. 9th ACM Conf. on Electronic Commerce*.

Chawla, S., Hartline, J., Malec, D., and Sivan, B. (2010a). Sequential posted pricing and multi-parameter mechanism design. In *Proc. 41st ACM Symp. on Theory of Computing*.

Chawla, S., Malec, D., and Sivan, B. (2010b). The power of randomness in bayesian optimal mechanism design. In *ACM Conference on Electronic Commerce*, pages 149–158.

Clarke, E. H. (1971). Multipart pricing of public goods. *Public Choice*, 11:17–33.

Coase, R. (1972). Durability and monopoly. *Journal of Law and Economics*, 15:143–149.

Day, R. and Milgrom, P. (2007). Core selecting package auctions. *International Journal of Game Theory*, 36:393–407.

Demers, A., Keshav, S., and Shenker, S. (1989). Analysis and simulation of a fair queueing algorithm. In *Symposium proceedings on Communications architectures & protocols*, pages 1–12.

Devanur, N. and Hartline, J. (2009). Limited and online supply and the bayesian foundations of prior-free mechanism design. In *Proc. 11th ACM Conf. on Electronic Commerce*, pages 41–50.

Devanur, N. and Kakade, S. (2009). The price of truthfulness for pay-per-click auctions. In *Proc. 11th ACM Conf. on Electronic Commerce*, pages 99–106.

Dhangwatnotai, P., Roughgarden, T., and Yan, Q. (2010). Revenue maximization with a single sample. In *ECOM10*.

Dughmi, S. and Roughgarden, T. (2010). Black-box randomized reductions in algorithmic mechanism design. In *Proc. 51th IEEE Symp. on Foundations of Computer Science*, pages 775 –784.

Dughmi, S., Roughgarden, T., and Sundararajan, M. (2009). Revenue submodularity. In *Proc. 11th ACM Conf. on Electronic Commerce*.

Dwork, C. and Naor, M. (1992). Pricing via processing or combatting junk mail. In *12th Annual International Cryptology Conference*, pages 139–147.

Feige, U., Flaxman, A., Hartline, J., and Kleinberg, R. (2005). On the Competitive Ratio of the Random Sampling Auction. In *Proc. 1st Workshop on Internet and Network Economics*, pages 878–886.

Fiat, A., Goldberg, A., Hartline, J., and Karlin, A. (2002). Generalized competitive auctions. In *Proc. 34th ACM Symp. on Theory of Computing*, pages 72–81. ACM Press.

Goldberg, A., Hartline, J., Karlin, A., and Saks, M. (2004). A lower bound on the competitive ratio of truthful auctions. In *Proc. 21st Annual Symp. on Theoretical Aspects of Computer Science*. Springer.

Goldberg, A. V. and Hartline, J. D. (2003). Competitiveness via consensus. In *Proc. 14th ACM Symp. on Discrete Algorithms*. ACM/SIAM.

Goldberg, A. V., Hartline, J. D., and Wright, A. (2001). Competitive auctions and digital goods. In *Proc. 12th ACM Symp. on Discrete Algorithms*, pages 735–744. ACM/SIAM.

Griffin, T., Shepherd, B., and Wilfong, G. (2002). The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10:232–243.

Groves, T. (1973). Incentives in teams. *Econometrica*, 41:617–631.

Guruswami, V., Hartline, J., Karlin, A., Kempe, D., Kenyon, C., and McSherry, F. (2005). On profit-maximizing envy-free pricing. In *Proc. 16th ACM Symp. on Discrete Algorithms*.

Ha, B. and Hartline, J. (2011). Mechanism design via consensus estimates, cross checking, and profit extraction. *Transactions on Economics and Computation*, 1. to appear.

Hartline, J., Kleinberg, R., and Malekian, A. (2011). Bayesian incentive compatibility via matchings. In *Proc. 22nd ACM Symp. on Discrete Algorithms*.

Hartline, J. and Lucier, B. (2010). Bayesian algorithmic mechanism design. In *Proc. 41st ACM Symp. on Theory of Computing*.

Hartline, J. and McGrew, R. (2005). From optimal limited to unlimited supply auctions. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 175–182.

Hartline, J. and Roughgarden, T. (2008). Optimal mechanism design and money burning. In *Proc. 39th ACM Symp. on Theory of Computing*.

Hartline, J. and Roughgarden, T. (2009). Simple versus optimal mechanisms. In *Proc. 11th ACM Conf. on Electronic Commerce*.

Hartline, J. and Yan, Q. (2011). Envy, truth, and profit. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 243–252.

Karp, R. (1972). Reducibility among combinatorial problems. In Miller, R. and Thatcher, J., editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press.

Kirkegaard, R. (2006). A short proof of the bulow-klemperer auctions vs. negotiations result. *Economic Theory*, 28:449–452.

Kruskal, J. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proceedings of the American Mathematical Society*, volume 7, pages 48–50.

Lehmann, D., O'Callaghan, L., and Shoham, Y. (2002). Truth revelation in rapid, approximately efficient combinatorial auctions. *Journal of the ACM*, 49.

Myerson, R. (1981). Optimal auction design. *Mathematics of Operations Research*, 6:58–73.

Nisan, N. and Ronen, A. (2001). Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196.

Oxley, J. (2006). *Matroid Theory*. Oxford graduate texts in mathematics. Oxford University Press.

Reynolds, S. and Wooders, J. (2009). Auctions with a buy price. *Economic Theory*, 38:9–39.

Samuel-Cahn, E. (1984). Comparison of threshold stop rules and maximum for independent nonnegative random variables. *The Annals of Probability*, 12(4):1213–1216.

Taghi-Hajiaghayi, M., Kleinberg, R., and Sandholm, T. (2007). Automated online mechanism design and prophet inequalities. In *AAAI*, pages 58–65.

Talwar, K. (2003). The price of truth: Frugality in truthful mechanisms. In *Proc. 20th Annual Symp. on Theoretical Aspects of Computer Science*, pages 608–619.

Thanassoulis, J. (2004). Haggling over substitutes. *J. of Economic Theory*, 117:217–245.

Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. *J. of Finance*, 16:8–37.

Vincent, D. and Manelli, A. (2007). Multidimensional mechanism design: Revenue maximization and the multiple-good monopoly. *Journal of Economic Theory*, 137(1):153–185.

Wang, X., Montgomery, A., and Srinivasan, K. (2008). When auction meets fixed price: a theoretical and empirical examination of buy-it-now auctions. *Quantitative Marketing and Economics*, 6:339–370.

Welsh, D. (2010). *Matroid Theory*. Dover Books on Mathematics. Dover Publications.

Wilson, R. (1987). Game theoretic analysis of trading processes. *Advances in Economic Theory*.

Wilson, R. (1997). *Nonlinear Pricing*. Oxford University Press.

Index