1. Introduction

What do economics and computation have to do with each other, and what can be gained through thinking about economics and computation at the same time?

In this chapter, we introduce the main theme of the book, namely the interplay between economic and computational considerations.

Economics can be described as the study of decision making by multiple actors, each with individual preferences, capabilities, and information, and motivated to act in regard to these preferences. Understanding what outcomes can be achieved in systems with multiple actors (e.g., what resource allocations, production decisions, actions) under incentive constraints is fundamental to economics.

Computer science can be described as the study of the representation and processing of informa- tion for the purpose of specific calculation tasks. Understanding what types of computation can be carried out efficiently, under time, resource and communication constraints is fundamental to computer science. Taken together, by studying topics at the intersection of economics and computer science ("EconCS" in short) we have in mind,

the analysis and design of systems whose successful performance depends on achieving good incentive properties and good computational properties.

We emphasize the need for combining economic and computational thinking for the purpose of design. For an economist, design is not just about aligning incentives and recognizing that participants may be individually motivated actors, but about recognizing the need for the computational problems associated with a design to be tractable. For a computer scientist, design is not just finding efficient ways to generate outputs from inputs, but about the effect that design decisions have on the behavior of participants in a system, this behavior affecting the inputs.

In the next sections, we consider three concrete examples where economic and computational thinking are important. These lead into a broad discussion of various touch-points at the intersection between economics and computer science.

1.1. The Braess Paradox

We start by thinking about a problem that involves flow on a network, and comparing the way the network is used when individuals make decisions in their own best interest vs the way the network would be used if the flow could be coordinated in a way that maximizes the interest of society.

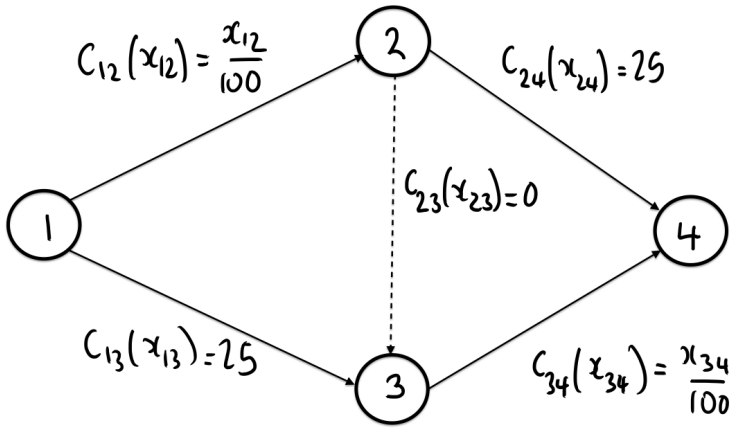


Figure 1.1.: A network in which 2000 people need to travel on this network from location 1 to location 4. Each directed edge has an associated cost that determines the time to traverse the edge as a function of the flow on the edge. Edge 2-3 models a super-highway.

Example 1.1. *Consider Figure 1.1. This could model for example a road network connecting four cities. Suppose that* 2000 *people need to move from location* 1 *to* 4*. Each edge in the network is directed, and is associated with a cost function that determines the time it takes to cross the edge. The time it takes to get from location* 1 *to* 2 *depends on the total flow* x_{12} *on that edge, in particular, the cost is* $c_{12}(x_{12}) = x_{12}/100$*. The edge from location* 2 *to* 4 *has cost* $c_{24}(x_{24}) = 25$*. On the second path, we have cost* $c_{13}(x_{13}) = 25$ *and cost* $c_{34}(x_{34}) = x_{34}/100$*. Optionally, there is a super-highway between locations* 2 *and* 3*, with cost* $c_{23}(x_{23}) = 0$*.*

The first question we ask is: what is the flow of traffic that is optimal for society, and what will be the travel time for each individual in this flow? We assume that the social objective is to minimize the maximum delay. This is only one possible goal; another goal would be to minimize the total sum of travel times. By symmetry of costs, the optimal flow without edge 2-3 is balanced with 1000 people taking path 1-2-4 and 1000 people taking on path 1-3-4. This results in a total travel time for each person of 35 minutes. At least one person would have a higher delay in any other flow.

It is interesting to compare this with what we would expect to happen if each person makes an individual choice, looking to minimize her own travel time given the choices of others. In fact, individual choices would lead to the same flow. Consider the choice of someone using path 1-2-4 when 1000 use 1-2-4 and 1000 use 1-3-4. If she was to change her mind and use path 1-3-4, then her travel time would increase to $25 + 1001/100 = 35.01$ because there would be 1001 people on this path. We say that this flow is an *equilibrium flow* on the network.

Now suppose that the super-highway is added, connecting locations 2 and 3 with zero cost. We see something surprising, which is that the system-optimal flow is unchanged while the equilibrium flow becomes worse. For there to be a flow x' with delay less than 35, we must

have $x'_{13} < 1000$ (otherwise, anyone using path 1-3-4 would incur cost at least 35.) Then, $x'_{12} > 1000$, and because the delay on path 1-2-4 would be more than 35, all this traffic must take path 1-2-3-4. Now there are two possibilities. First, $x'_{12} = 2000$ and everyone takes path 1-2-3-4, but then the delay would be 40. Alternatively, $x'_{13} > 0$ and the delay on path 1-3-4 would be 45 (because everyone is using edge 3-4 in the situation we are considering.) We conclude that there is no flow with delay less than 35, and so the same flow with 1000 people on path 1-2-4 and 1000 on path 1-3-4 remains optimal.

But what about the effect of the super-highway on the equilibrium flow? In fact, the system optimal flow is no longer an equilibrium. No one would ever choose edge 1-3 because this takes 25 minutes, and even if all 2000 people take the path from 1-2-3, this only takes 20 minutes. The same reasoning holds for edge 2-4, where path 2-3-4 is always better for an individual. We conclude that each person would choose path 1-2-3-4. This choice is a *dominant strategy*, being the right decision whatever the actions of others, and all traffic using 1-2-3-4 is the new equilibrium.

By adding an edge with zero delay, the equilibrium travel time has become worse for everyone, leading to a paradox! This is the *Braess Paradox*, named after the mathematician Dietrich Braess. This unfortunate situation makes it clear that incentives matter when designing systems that will be used by individuals with self-interest. The rational self-interested behavior of individuals can lead to outcomes that are not societally beneficial. In this case, there is a kind of *tragedy of the commons*, where individuals use too much of a shared resource (the commons, in this case the super-highway), and the end result is bad for everyone.

1.2. Using an Auction for Sharing Computational Resources

The Braess paradox suggests the importance of considering incentive constraints when designing systems to be used by multiple participants who may have misaligned interests. Such systems are becoming ubiquitous in the digital economy. Consider for example e-commerce platforms (eBay, Amazon), online advertising markets (Google, Yahoo, Microsoft), social and micro-blogging networks (Facebook, Twitter), and Web 2.0 platforms (Wikipedia, Yelp, Tripadvisor, Digg).

The design of computational systems with multiple, self-interested participants was already a challenge before the advent of the internet. Early computers were very costly, shared amongst many users, and in great demand. In the 1960s, researchers at Harvard University used an auction system to determine who would gain access to the PDP-1, the world's first interactive, commercial computer.

Example 1.2. *In the auction for access to Harvard's PDP-1, users could mark bids in integer amounts of currency for a block of time on a long roll of transparent paper, starting and ending on quarter hours, and with different colors to indicate the bid amount. The paper was moved up each day so that the next two weeks were always displayed. A user could out bid a current bid (including one of their own bids) by bidding for a block of time at a higher hourly rate, by writing a new bid above the current bid on the schedule.*

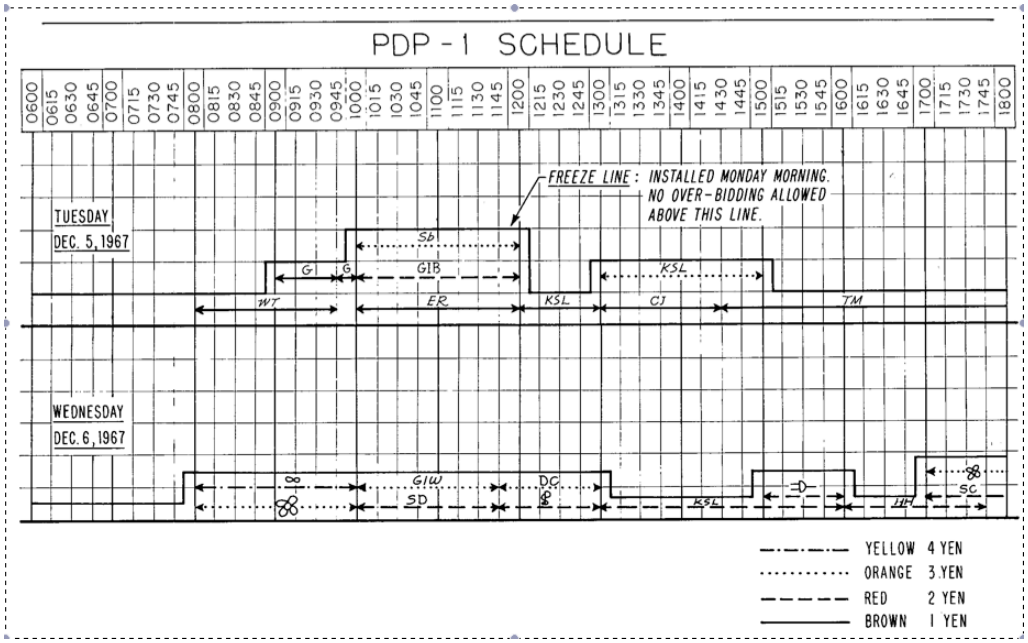


Figure 1.2.: Bidding in an auction for access to Harvard's PDP-1 computer (Sutherland 1968).

See Figure 1.2. For example, the bid by KSL on Tuesday at 3 *Yen (the name given to the virtual currency) for* [13:00,15:00] *is at rate* 3/2 *per hour. This outbids the bid by CJ at* 1 *Yen for* [13:00,14:30] *(rate* 1/1.5*) and the first* 30 *mins of the bid by TM for* [14:30,18:00] *at* 1 *Yen (rate* 1/3.5*). The effect is that the bid of CJ is no longer active and the bid of TM is retained at* 1 *Yen for* [15:00,18:00] *(rate* 1/3*).*

Different users were allocated different amounts of currency, this budget allowing for a simple priority scheme. A user's outstanding bids for future time could not exceed a user's allocated budget. Bids could be canceled and the currency re-used elsewhere, but a user could not lower a bid, or bid again at some later instance on a block of time where a bid had been canceled. Once an allocation was made to a user and payments collected, the currency was returned to the user and could be used again.

Other rules were designed to preclude fragmentation of time on the shared machine, so that users received large enough contiguous blocks of time. For example:

(i) outbidding part of an existing bid could only be done at the beginning or end of a block of time; e.g., a bid of 2 Yen for [1:00,3:00] could be outbid with a bid of 3 Yen for [12:30,1:30], but not with a bid of 3 Yen for [1:30,2:30],

(ii) outbidding part of an existing bid that was holding an hour or more could only be done while leaving the existing bid with at least an hour on the machine; e.g., a bid of 3 Yen for [12:30,2:00] but not 3 Yen for [12:30,2:30] would be allowed given an existing bid of 2 Yen for [1:00,3:00].

In summarizing the useful properties of the PDP-1 auction, Ivan Sutherland, the designer of the system, remarked:

> *"We have found that under the auction system... our computer utilization is very high... The computer is never idle, as often happens under other allocation schemes, merely because everyone has used up his current monthly allotment of time. If the computer ever is idle, its price automatically becomes attractively low... Although users complain when their bids are preempted, they are generally glad to have a choice between short periods of expensive prime time and long periods of time at less desirable periods..."*

Preemption refers to the possibility that one bid can be outbid by another, leaving some part of the original bid standing. These remarks make the case for the use of economic approaches for resource allocation within multi-party computational systems. The auction allowed users of the PDP-1 to express their intensity of preference for access at different times of the day. Anyone could use the machine when it was under-utilized. At other times, those with the highest value could gain access. The effect was to promote *allocative efficiency*, which insists that resources are allocated to maximize total value. Strategic behavior was observed by users of the PDP-1 auction, as illustrated in the next example.

Example 1.3. *The rules of the PDP-1 auction interpreted multiple bids from the same user on consecutive blocks of time as a single bid for the purposes of bidding rules (i) and (ii). This rule was intended to allow a user to express a preference for an early portion of time through a bid* 3 *Yen for* [1:00,1:30] *and* 1 *Yen for* [1:30,3:00]*. But the rule also enabled new kinds of strategic behavior. Suppose that a user wanted to obtain time block* [9:00,10:00] *for at most* 2 *Yen. Rather than bid* 2 *Yen for* [9:00,10:00]*, she could place of a composite bid of* 1 *Yen for* [9:00,9:15] *and* 1 *Yen for* [9:15,10:00]*. The straightforward bid could be outbid by a bid of* 3 *Yen for* [9:00,10:00]*. The composite bid cannot be outbid in this way, since this would leave only 15 minutes (the bid* 1 *Yen for* [9:15,10:00] *would remain), and violate rule (ii). Instead, another user would need to bid* 2 *Yen for* [9:15,10:00] *and* 2 *Yen for* [9:15,10:00]*, for a total of* 4 *Yen. Thus, users could gain by breaking up his bid into multiple consecutive bids. Indeed, User G is adopting this strategy in Figure 1.2.*

There are many reasons for designers to prefer systems in which users can't benefit from strategic behavior. First, this makes the system easier to use, removing the need for users to speculate about the behavior of others. Second, there are fairness considerations if some but not all users are aware of the possibility of strategic behavior. Third, strategic behavior may lead to inefficient resource allocation where resources are not allocated to those with the highest value. Fourth, the need for one user to modify his input in response to changes in the inputs of others can place overhead on the system, with inputs continually changing as users seek competitive advantage. This was a problem in early designs of ad auctions for internet advertising, which we discuss in the next section.

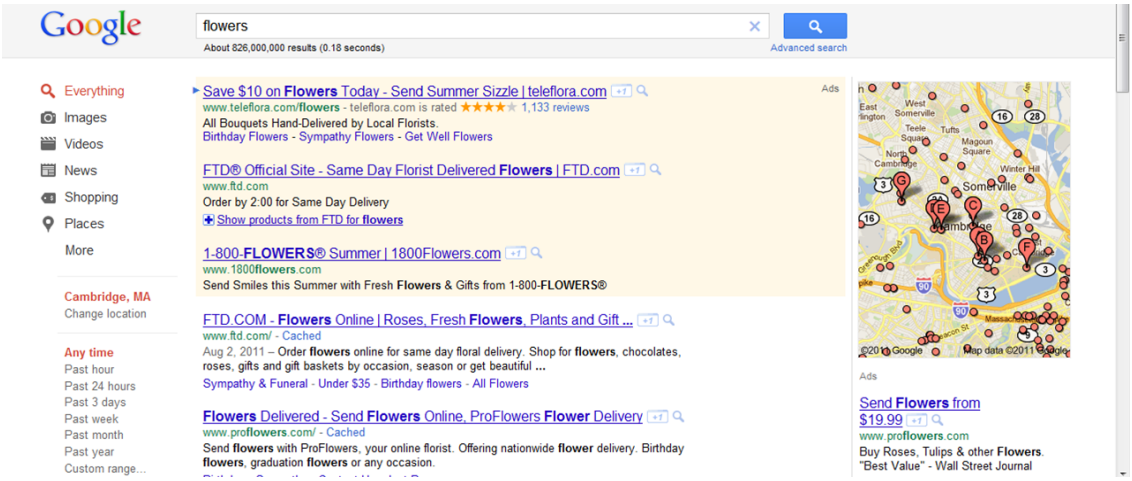


Figure 1.3.: Sponsored search listings, with ads shown above and to the right of the search results.

1.3. Using an Auction to Allocate Advertisements

Alongside the organic search results, which are links to content that an internet search engine considers to be most relevant for a user query, are sponsored search results. For example, Figure 1.3 shows three ads above and one ad to the right in response to a query for "flowers." Which ads are displayed to a user is determined via an auction.

Auctions are suitable for selling items without a standardized value. This is the case when selling ad space to users of a search engine. Not only is there an essentially unlimited number of possible search queries, but the value of advertisers can depend on context such as time of day, location, user demographic, and so forth. Ad auctions are the main driver of revenue for many large Internet firms.

Multiple ad positions are offered for sale in sponsored search, both above and to the right of organic results. Higher positions tend to be more desirable. In standard designs, bids are placed per-click, with payment only collected from an advertiser in the event of a click on an ad that is shown. Bids are ranked according to the expected value of the bid. The search engine estimates the clickthrough rate (CTR) for an ad, which is the probability the ad will receive a click, and considers CTR $\times$ bid amount.

Early auction designs used *first price* or *pay-your-bid* payment rules. In the event of being allocated a position and receiving a click from a user, an advertiser would pay the amount of her bid. The effect of this rule was that strategic behavior was beneficial. For example, an advertiser might try to maintain a particular position while bidding as little as possible.

Early auctions didn't estimate the quality of ads, and simply ranked bids by bid amount. These auctions also reported all the current bids on a keyword associated with a search query. Based on this, it was a simple matter to automatically monitor an auction and update bids to

maximize profit. This led to bidding wars, as illustrated in the following example:

Example 1.4. *Consider an auction with two positions, and where the clickthrough rate is* $CTR_1 = 0.1$ *on position* 1 *and* $CTR_2 = 0.02$ *on position* 2*, assumed to be the same for all ads. Suppose that there are three advertisers, with values* \$15*,* \$11 *and* \$5.99 *per-click, for advertisers* 1*,* 2 *and* 3 *respectively. If advertiser* i *with per-click value* v_i *wins position* j *with* CTR_j *and pays bid amount* b_i *per click, her expected profit is* $CTR_j(v_i - b_i)$*, i.e., the difference between value and bid, multiplied by the probability of receiving a click.*

We assume that all bids must be placed in cents, and that advertiser 3 *always bids* \$5.99*. A best-response for an advertiser is place a bid that maximizes her expected profit, assuming the other bids remain the same. Starting with bids* \$6.01*,* \$6.00 *from advertisers* 1 *and* 2 *respectively, a sequence of best-response moves leads to the following dynamic:*

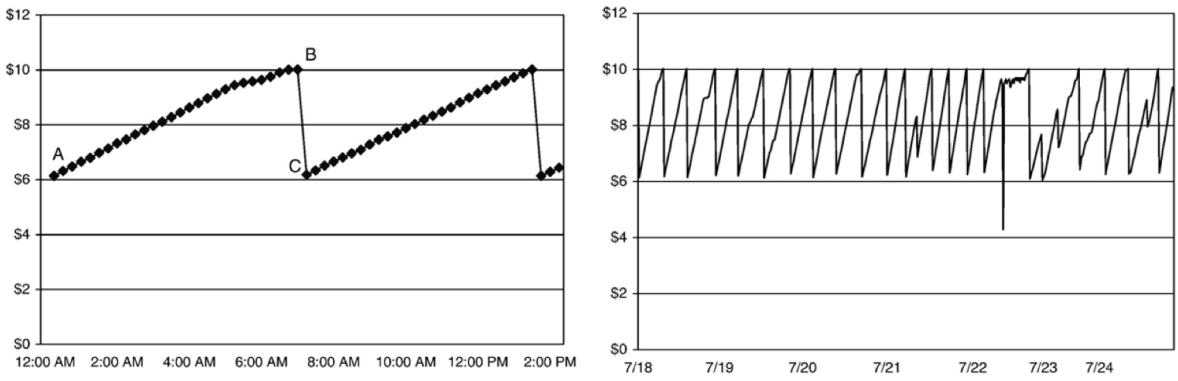
period	1	2	3	4	5				$*$			
b_1:	6.01	6.01	**6.03**	6.03	**6.05**	...	**9.98**	9.98	**10.00**	10.00	**6.01**	...
b_2:	6.00	**6.02**	6.02	**6.04**	6.04	...	9.97	**9.99**	9.99	**6.00**	6.00	...
b_3:	5.99	5.99	5.99	5.99	5.99	...	5.99	5.99	5.99	5.99	5.99	...

The modified bid in each period is indicated in bold. In period 1*, advertiser* 1 *wins position* 1 *for \$6.01 and advertiser* 2 *wins position* 2 *for* \$6.00*. Assuming that advertisers* 1 *and* 3 *leave their bids unchanged, then advertiser* 2 *can now do the following analysis:*

- *Bid* \$6.00*: win position* 2*, for expected profit of* $0.02(\$11.00 - \$6.00) = \$0.1$*.*
- *Bid* \$6.02*: win position* 1*, for an expected profit of* $0.1(\$11.00 - \$6.02) = \$0.498$*.*

The best response is to bid \$6.02 *and win position* 1*. In the next period, advertiser* 1 *faces a similar decision (comparing a bid of* \$6.00 *and a bid of* \$6.03*), and* \$6.03 *is the best response. This leads to a cyclic bidding pattern, which continues until the bids reach* \$10.00*. At that point (indicated* $*$*), the situation changes for advertiser* 2*. Bidding* \$10.01 *to win position* 1 *would lead to an expected profit of* $0.1(\$11.00 - \$10.01) = \$0.099$*, while bidding* \$6.00 *to win position* 2 *would lead to an expected profit of* \$0.1*. The best-response is to bid* \$6.00*, and the cycle repeats.*

Figure 1.4 (a) shows the highest bid submitted for a particular keyword in a deployed ad auction system during a day in July 2002. Two bidders are engaged in a bidding war for the top position. Between periods A and B the bidders continually out bid each other, increasing the highest bid in the auction. At period B one of the two bidders stops competing, and instead bids just enough to take second position in the auction, and win the second-highest position. At period C the other bidder lowers his bid to be just above the new, lower bid his competitor, and retain the top position at a minimal bid amount. The cycle then begins again. Figure 1.4 (b) shows that this sawtooth dynamic continued quite robustly throughout the week of July 18, 2002.



a: 14 hours

b: 1 week

Figure 1.4.: A sawtooth bidding pattern in first-price sponsored search auctions (Edelman and Ostrovsky 2007).

This bid dynamic reduces allocative efficiency, with the top position allocated to the advertiser with the second highest value half of the time. Although the effect on revenue could in theory be positive or negative, in practice it was estimated to be at least 7% lower than the revenue that could have been achieved with a more stable design. In addition, these bid dynamics placed a heavy load on the auction servers, with advertisers continuously sending messages to the auction servers to change their bids.

Many ad auction systems in use today address this design flaw by using a *generalized second price* (GSP) payment rule. The payment in the event of a click is not the bid price but a so-called second price, namely the smallest bid amount at which the advertiser would have retained the same position. In effect, the auction and not the advertiser determines the smallest successful bid that an advertiser could have placed. This new payment rule was first adopted by Google in 2002, and adopted shortly thereafter by Overture (acquired by Yahoo), and is used today by major search engines.

The idea of charging an amount equal to the next highest bid goes back to the birth of auction theory. Suppose we're selling the right to show an ad on a billboard. In a first price auction, the ad rights are sold to the highest bidder for the amount of his bid. In a second price auction, the ad rights are sold to the highest bidder for the amount of the second-highest bid. For example, if the bids are $1,700, $1,500 and $1,000, then the rights are sold to bidder 1 for $1,500. The optimal strategy for a bidder in a second-price auction is to bid her true value, whatever the other bids. For example, if advertiser 1's value is truly $1,700 then this is her optimal bid. Bidding less does not change the amount she pays when she wins, and could lead her to lose to another bidder. Bidding more could lead to winning for a price greater than her

value. This auction design is *truthful*, meaning that the best possible thing for a bidder to do is to report her true value whatever anyone else does.

1.4. A Rich Tapestry

Economics and computer science have found a rich interaction since the 1920s, when pioneers such as John von Neumann contributed basic theory on equilibrium concepts to economics and also the architectural basis for modern computers.

The essential idea of an equilibrium is that every participant is best-responding to the actions of everyone else. In the setting of the Braess paradox, an equilibrium is a flow of traffic such that no user has an incentive to change his route. But what about the computational difficulty in identifying an equilibrium? If it is too hard to identify an equilibrium, then we have reason to worry that equilibria cannot be descriptive of behavior. For this reason, it can be useful to bring computational thinking to bear on these fundamental concepts from economic theory. Indeed, some have observed 'if your laptop can't compute it then neither can the market,'' expressing the concern that economic theory should not be predicated on being able to solve intractable computational problems.

Looking to practice, there are many e-commerce and social-computing technologies for which economic considerations such as incentives are central to their design. These include systems of *peer production* such as Wikipedia and Quora, *crowdsourcing* platforms such as oDesk and Amazon Mechanical Turk, *social networks* such as Facebook and Google+, *micro-blogging* systems such as Twitter, and *reputation and recommendation systems* such as those used by eBay, Amazon and TripAdvisor. Computational finance is another area where computational and economic thinking comes together, with markets that were originally designed for human traders increasingly used by automated trading programs.

The following sections provide examples of computational systems with economic characteristics and economic systems with computational characteristics. Section 1.4.3 introduces some of the cross-cutting issues involved in the study of economics and computation, for example those related to models of user motivations, and to the broader considerations of privacy and security.

1.4.1. Computational Systems with Economic Characteristics

The internet itself. The internet is owned and operated by commercial entities, and thus exhibits the characteristics of both an economic and a computational system. Distributed algorithms that are used for determining the way traffic passes through the fiber-optic networks of the internet can be configured to prefer some routes over others according to business preferences.

In particular, the *autonomous systems* (ASs) that route traffic through the internet are business entities. Some ASs form peering agreements with each other, and do not charge each other for traffic flows that originate in one and flow through the other. In other cases, ASs reach agreements to make payments based on traffic flows (e.g., ''you pay me when I transit traffic on your behalf.'') Through the internet's *Border Gateway Protocol* (BGP),

ASs exchange information about the different routes that can be taken over the internet in order to reach a particular destination address. Rather than provide a specific algorithm for determining a path to share with others, the paths that are advertised in this way depend on how individual routers are configured. For example, one AS might configure its routers to prefer a longer (higher latency) path if this avoids making a payment. Another AS might configure its routers to selectively advertise some paths and not others. This raises questions in regard to the effect that incentives and self-interest have on the robustness of the basic infrastructure of the internet.

Internet search. The position of a website in the ranking of search results is extremely important in driving traffic to a site. For this reason, an entire *search engine optimization* industry is dedicated to influencing search results. This is paradigmatic of one of the new considerations that economic thinking brings to computer science. Traditionally, we think about the inputs to algorithms as coming from some known distribution, or being selected to provide hard instances. But, when algorithms are used in economic domains, the design of the algorithm affects the inputs! In the case of search, firms want to try to change links and other information on the web in order to improve their search rank.

Peer-to-peer systems. In a computational peer-to-peer (P2P) system, each computer both contributes and consumes resources. This can enable systems to reach massive scale without the deployment of very much centralized infrastructure. For example, Voice-over-IP platforms such as Skype employ P2P technology: each user's computer routes calls for other users. P2P systems are also used for file sharing. The *BitTorrent protocol* splits a large file into many smaller pieces, and allows machines to exchange pieces until the entire file can be put together. As in other settings, a fundamental problem in P2P systems is *free-riding*: users may prefer to consume resources without contributing resources. Whereas earlier file-sharing protocols failed to successfully align incentives, the BitTorrent protocol incorporates basic reciprocation. Each peer allocates upload bandwidth preferentially to peers from which it is able to download at a high rate. In this way, users are generally able to improve their own download speed by sharing some bandwidth for upload-- and the system provides better incentive alignment.

Social computing systems. Social computing systems and the social web, such as social news aggregators (e.g., Digg, Reddit), collaborative content systems (e.g., Wikipedia), online communities and question-and-answer systems (e.g., Flickr, Quora, Yahoo Q&A), and social networks (e.g., Facebook, Google+), rely both on user participation and well designed algorithms and systems for their success. For example, the social news aggregator Digg considers user votes along with the recency of an article in deciding how to rank articles. Yahoo Q&A provides users with points for answering questions, and requires users to spend points to post questions, with leader boards provided to encourage users to accumulate a large number of points. The contributions of users on Wikipedia can be acknowledged by other users through virtual awards such as barnstars. Social networks are designed to make it easy to share content and easy to tag photos with a person's name, providing a mechanism by which photos are shared with

friends.

1.4.2. Economic systems with Computational Characteristics

Ad Exchanges. Ad exchanges are used for selling display ads on web pages. Publishers such as news websites notify an exchange when a user loads a page. In realtime, the exchange notifies multiple *ad networks*, each representing multiple advertisers. This call provides information in regard to the web content and the user. Each network can submit a bid, and the highest bid wins and determines the ad shown to the user. All of this occurs in tens of a millisecond. All aspects of these markets are automated, including the processes for determining (i) whether a publisher should send an ad opportunity to an ad exchange, (ii) whether and how much to bid, and which ad to show, and (iii) the outcome of the auction.

Wireless Spectrum Auctions. National governments use auctions to allocate rights to use wireless spectrum and operate cellular networks. Increasingly, these rights are sold through *combinatorial auctions* (CAs). Multiple items are sold at the same time in a CA, each representing different geographic spectrum and different frequency band, and bidders can bid on packages of items. For example, a bidder might bid on a package of spectrum licenses in a contiguous region such as across the Eastern seaboard of the U.S. By allowing bids on packages, bidders avoid the risk of winning some but not all items that they want. An example of a computational challenge in the design of CAs is that of designing a *bidding language* that does not require bidders to place a bid on every possible package of items. This is important because the number of packages of items grows exponentially in the number of items. Another computational challenge is that the winner determination problem is hard to solve; this is the problem of finding a set of disjoint bids on packages that maximizes total bid value.

Prediction markets. Prediction markets aggregate the information of multiple parties in order to predict uncertain future events. For example, the price of an asset that pays \$1 if a category-5 hurricane will hit Florida in 2015 can provide an estimate of the probability of this event. If the price is \$0.3, it implies a consensus belief that a hurricane will hit with probability 0.3. The *Iowa Electronic Market* forecasts the outcome of political elections. Inkling provides hosted software to allow organizations to run prediction markets, for example, to predict whether a target date for a software release will be met. Figure 1.5 shows the price on the asset for Barack Obama being re-elected U.S. President in 2012 on the Intrade marketplace. There are many computational challenges in the design of prediction markets. These include the design of languages to allow traders to express complex information such as ''if event A does not occur, then I think event B will occur with probability 0.7,'' and in algorithms to compute the current price for these kinds of conditional outcomes.

Matching problems. Matching problems include those of matching students to schools, doctors to hospitals, students to dorm rooms, and patients to organ-transplant donors. For example, *deferred-acceptance* algorithms are used in a number of cities in the U.S. to match

Figure 1.5.: The price in a prediction market for a contract on whether Obama would be re-elected in the 2012 U.S. election

students to schools based on preference lists submitted by students and priorities provided by schools. Through the use of deferred-acceptance, which is also a tractable procedure, there are no opportunities for students to use strategic behavior and they can thus report their true rank list on schools.

Other matching problems are computationally intractable. In *kidney paired-donation*, the participants are patient-donor pairs, where the patient needs a kidney and the donor is a friend or relative willing to donate a kidney, but is incompatible with the patient. Finding a matching that maximizes the number of transplants is tractable when only considering swaps, but intractable when also allowing for 3-cycles. There are also incentive considerations, in encouraging hospital systems to share complete patient-donor lists with a centralized exchange rather than just difficult-to-match pairs.

Recommender and Reputation systems. Successful e-commerce models such as Amazon and Netflix use recommender systems to help consumers to find the products they want amongst a large number of offerings. One kind of recommender system uses *collaborative filtering*, where feedback collected from users is used to predict the items that other users might like. Intuitively, if in the past, User 1 liked items that User 2 liked, and disliked items that User 2 disliked, then User 2's ratings may be a good predictor for how much User 1 will like a new item. In addition to providing high-value recommendations to users, recommender systems should be robust to manipulation through such behaviors as submitting false feedback to unfairly boost or hurt a particular product.

Whereas recommender systems are about making predictions about the idiosyncratic preferences of different users, reputation systems are designed to elicit and aggregate facts such as

whether or not a seller shipped a good. In this way, reputation systems automate traditional word-of-mouth propagation. The design of a reputation system needs to consider the broader context, including the possibility for strategic behavior. For example, an early reputation system on eBay suffered from a problem where buyers were reluctant to give negative feedback because a seller could retaliate and leave negative feedback about the buyer. Another question about the design of reputation systems is in regard to personalized feedback, and questions of *transitive trust*: if A trusts B and B trusts C then should A trust C, and how much?

1.4.3. Cross-cutting Considerations

In this section, we draw together some cross-cutting issues about networks, user motivations and broader concerns such as those of privacy and security.

Networks. Networks such as the web, social networks, and trading networks often have an important role in mediating transactions and user behavior, and in controlling the flow of information. Some interesting EconCS considerations in regard to networks include:

(i) *Network formation*: How do networks grow, and what behavioral aspects of participants and motivations affect this process?

(ii) *Information cascades and influence*: How does information propagate over networks, and what causes a participant in a network to communicate with neighbors on the network? How can the influence of an individual on the spread of information be measured, and how can this influence be used for personal profit?

(iii) *Cooperative behavior*: How can cooperative behavior, for example reciprocation in solving problems, sharing information, or annotating content, be promoted in networks and what determines the behavior of individuals?

(iv) *Platform-mediated networks*: How can platforms mediate the formation and use of networks, by controlling how they grow, how benefits are distributed to new participants, and by promoting particular actions? For example, can a platform mobilize an initial user base and promote user actions that bring value to other users?

User Motivations. Traditional economic models generally assume the *selfish-rational actor model*, where participants only care about themselves and take actions to maximize their own happiness. This provides a precise and simple model, and is indeed relevant in e-commerce domains such as auction platforms. On the other hand, it does not seem to explain very well the contributions made by users to Wikipedia or Yahoo Q&A.

Alternative models of user motivations include *other-regarding preferences* such as *inequity aversion*, *reciprocity*, and *altruism*. Understanding these kinds of motivations leads to new design patterns, such as providing mechanisms for social recognition such as through the badges on Stackoverflow and Wikipedia's barnstars. Another consideration in the design of systems is that users may not act rationally but rather follow simple decision rules in deciding how to act, leading them to make bad tradeoffs between short-term and long-term rewards.

Privacy and Security The privacy of individuals is a significant concern about the advance of the digital economy is that of *privacy*. On the one hand, information about individuals allows for personalization of content and services and enables revenue-generating advertising. On the other hand, the continued aggregation of data about individuals leads to concerns in regard to privacy, and the possibility of unintended and unanticipated consequences. For this reason, there is great interest in developing principled methods to protect user privacy, including methods to aggregate or add noise to data in order to reduce the possibility that information about any one individual is revealed.

The *security* of systems that are used in support of the digital economy is another broad-ranging concern. For example, *phishing* (pretending to be a web site or an entity that you are not) and *spamming* (gaining user attention through sending unwanted communications) have a significant impact on the effectiveness of e-commerce platforms. Not only are there direct costs that arise, for example through fraud, but there are also indirect costs where it becomes difficult for example to communicate with customers by email.

Digital currencies such as *Bitcoin* are a place where privacy and security concerns come together in new ways. Payments can be made through pseudonyms (an identity owned by an individual but not publicly tied to an individual), so that it is not possible to track cash flows through the system. This protects user privacy, while also raising concerns about international flows of money and the use of digital currencies to support illegal activity. Moreover, the infrastructure supporting the use of digital currencies must be robust against attacks, for example to protect the servers used to store digital cash from being attacked, with the bits that corresponding to money stolen.

1.5. Chapter Notes

Easley and Kleinberg (2010) provides a broad treatment of similar topics, but with more emphasis given to social science and network science. Nisan *et al.* (2007) provides an advanced treatment of a number of specialized economics and computer science topics. Feigenbaum *et al.* (2009) provide a general discussion of computational challenges in e-commerce. The PDP-1 auction is discussed in Sutherland (1968). The sawtooth bidding pattern in Figure 1.4 is discussed in Edelman and Ostrovsky (2007), and was observed by these authors in the *Overture* auction system, later acquired by Yahoo. Al Roth has written about the economist as an engineer. The comment on ‘‘your laptop computing” is due to Kamal Jain.

