Automated Markets and Trading Agents

Jeffrey K. MacKie-Mason   Michael P. Wellman

University of Michigan
Ann Arbor, MI 48109 USA
{ jmm, wellman }@umich.edu

April 8, 2005

Contents

1 Introduction **2**

2 Marketplace design framework **3**
2.1 Marketplace systems 4
2.2 Formal model 5
2.3 Possibilities and impossibilities 8

3 Automating market mechanisms **9**
3.1 Connecting: Discovery services 9
3.2 Dealing: Negotiation mechanisms 11
3.2.1 Smart markets for domain-specific applications 11
3.2.2 Combinatorial markets 17
3.3 Exchanging: Transaction services 22

4 Automating market participants **22**
4.1 Program trading 23
4.2 Market interfaces 24
4.3 Agent strategies 25
4.3.1 Continuous double auction strategies 25
4.3.2 Simultaneous ascending auction strategies 27
4.4 Case study: Trading agent competition 29
4.4.1 TAC travel-shopping rules 29
4.4.2 TAC experience 30

5 A computational reasoning methodology for analyzing mechanisms and strategies **32**
5.1 Generate candidate strategies 33
5.2 Estimate the "empirical game" 34

5.3 Solve the empirical game 35
5.4 Analyze the results 36
5.5 Discussion 37

1 Introduction

Many digitally mediated activities present participants with complex strategic decisions, involving significant interaction with other agents. The strategic dimension of electronic commerce, for instance, is obvious, not just for negotiation and trading, but also for ancillary commerce operations such as matchmaking, resource finding, advertising, recommendation, contracting, and executing transactions. All of these are increasingly subject to automation as part of online marketplaces [Wellman, 2004]. Other digital realms, not necessarily viewed as commerce per se, nevertheless involve pivotal strategic relationships. Examples include peer-to-peer resource sharing [Golle et al., 2001; Cox and Noble, 2003], formation of coalitions, teams, or affinity groups [Brooks and Durfee, 2003; Sandholm and Lesser, 1997; Tambe, 1997], scientific sharing of large-scale instrumentation and other infrastructure [Finholt and Olson, 1997; Finholt, 2003], and coordination of activity within organizations [Malone, 1987; Pynadath and Tambe, 2002].

There are a variety of possible benefits from automating markets. One is cost saving from automating some functions of existing non-computational markets. For example, search automation reduces the cost of finding goods and potential trading partners. Micropayment systems offer the hope—not yet fully realized—of enabling large volumes of remote, low-value transactions by reducing the execution overhead. Another benefit is the ability to extend markets in time and geographic scope by conducting them over networks. For example, eBay's main innovation is not in the form of its markets, but in its ability to make markets that bridge time and space.

The greatest disruptive potential may lie in the opportunity to deploy market mechanisms that are simply infeasible to operate without computer automation. Creating previously missing markets enables gains from trade, and the creation of new products and services, with first-order effects on social welfare. Such mechanisms were dubbed "smart markets", apparently by Vernon Smith. For example, a multi-airport landing slot allocation policy might require the solution of a constrained integer program as a function of bid messages from participating agents. Such policies are well beyond the capabilities of non-automated market mediators; in some applications they take CPU days to solve even with current hardware. The emergence of cheap, high-speed computation created excitement among market designers, because without automated computation many interesting allocation mechanisms were infeasible for problems with real-world scale.

We study issues in the design of automated markets with software agents: how to automate effectively various components of market transactions? We emphasize design issues impinging on strategy, and strategic behavior particular to the market setting. Our chapter complements Marks [2005], who focuses on the use of agent-based computational techniques as a *tool* for use in (not necessarily computational) market design. Thus, Marks emphasizes positive analysis: how we can use agent-based models to evaluate performance of various market designs. We adopt this perspective briefly in Section 5, where we describe a computational game methodology for analyzing agent strategies and computational market designs.

Given our focus on design, much of the contribution of our chapter to agent-based computational economics (ACE) is to the development of infrastructure. For ACE modelers to study the implications of various market designs and agent strategies they need to be able to implement computational representations that are correct, interesting, and tractable.

To assist in these endeavors, we first present, in Section 2, a conceptual market design framework. After graphically characterizing the design space for marketplace systems, we present a brief specification of a formal model that encompasses many of the interesting problems for market and agent design. The model provides a structured framework for organizing the literature review in the rest of the chapter.

Section 3 covers the largest body of material. We discuss design issues and implementation research for mechanisms that provide the three different types of market transaction services we identified in our conceptual framework: discovery, negotiation (what is usually, narrowly, called "the market"), and execution. We devote disproportionate attention to negotiation or deal-making market mechanisms, reflecting the relative attention economists in general give to each of the three stages.

In Section 4 we focus on the other major area for design: trading agents, who interact through the market mechanisms discussed in Section 3. We consider both theoretical and practical problems of designing strategies needed to make economically-intelligent trading agents. We present a case study based on a several-year history of trading agent competitions that have attracted substantial attention.

We close the chapter by presenting an emerging computational agent-based methodology for empirical game-theoretic analysis. This method has been developed to address a fundamental problem in the design of both trading agent strategies and the market mechanisms through which they interact: optimal strategies for complex (realistic) markets are analytically intractable. We consider empirical game-theoretic analysis a promising approach for systematic investigation of agent strategies, and then for the evaluation of market mechanism performance when agents follow successful strategies. These agent-based methods offer one way to close the loop between the over-simplified theoretical models of agents and market, and the practical problems that designers must solve to implement realistic markets.

2 Marketplace design framework

Markets allocate resources through a series of *transactions*, each an exchange of goods and services expressed in terms of an underlying monetary system. We find it useful to organize the life cycle of a transaction into three stages, representing the fundamental steps that parties must go through in order to conduct trade.

1. **Connecting**: the search for and discovery of an opportunity to engage in a market interaction.

2. **Dealing**: the negotiation of terms.

3. **Exchanging**: the execution of the terms of an agreed transaction.

These steps are illustrated in Figure 1. Of course, the boundaries between steps are not sharp, and these activities may be repeated, partially concluded, retracted, or interleaved along the way to a complete commercial transaction. Nevertheless, keeping in mind the three steps is useful as a way to categorize particular resource allocation services, which tend to focus on one or the other.

<Figure 1 about here>

Rarely are all of these tasks automated. Only some agents may be automated, and even then perhaps only partially. Some of the market functions (say, finding connections, or negotiating deals) may be automated, but not others. Therefore, it is not very useful to discuss automation of an entire system as a single problem. In this chapter we consider the components separately, reflecting the complexity of the problem and the division of labor in the research literature.

2.1 Marketplace systems

To organize our discussion, we present and discuss a schematic representation of the overall design problem.[1] In Figure 2 we embed a marketplace system in an environment of social institutions (e.g., language, laws, etc.). The marketplace system itself consists of agents and the market mechanism through which they interact. The market mechanism can be roughly subdivided into structures, practices, and rules for the tasks of connecting, dealing, and exchanging. We now offer more precise definitions of the central concepts, and provide a formal framework within which we analyze them.

<Figure 2 about here>

Marketplace system The *agents* who participate in the resource allocation problem, together with the *market mechanisms* through which they interact.

Mechanism The rules, practices and social structures of a social choice process, specifying (1) permissible actions (often limited to messages, expressible as a communication protocol) and (2) outcomes as a function of agent actions. A mechanism is *mediated* if there is some entity, distinct from the participants, that manages the communication and implements the mechanism rules.

Market mechanism A mechanism where the possible ultimate outcomes comprise market-based exchange transactions.

Agent An autonomous decision-making locus in a system of multiple decision-making entities.[2] An agent has "type" attributes such as preferences, beliefs, intentions, and capabilities. Type information is generally considered private, not inherently

[1] The descriptive terms we use do not have standard definitions, so we need to establish our own for these purposes. For example, some use "market" to refer to what we call a marketplace system. But others use "market" just for the practices and structures for making deals, excluding the participating agents and the other activities (such as connecting) from the term.

[2] For purposes of framing the general design problem, we apply the term *agent* generically to humans, computational processes, or organizations as long as they exhibit agent characteristics.

accessible to others. For purposes of analysis, we may attribute to agents particular decision-making rules, or more generally, assume that they conform to some *decision rule*, specifying a form of consistency between the agent's behavior, beliefs and preferences. Such attributions may appeal to classical notions of rationality, as well as alternative bounded or otherwise nonstandard coherence criteria.

Our characterization of a marketplace system indicates that there is not just one design problem, but several. The first is design of the market mechanism, for use by human (undesigned), or computational (designed) agents. The market mechanism may be decomposed into several design subproblems: for example, mechanisms typically are designed separately for the connection, deal and exchange phases of a transaction. The second top-level problem is design of agents to interact (perhaps with human assistance) with existing market mechanisms, or with new mechanisms designed by others. In some situations one might be in a position to design an entire marketplace system, though we consider this unusual.

In this chapter, we focus on market mechanism design in Section 3, devoting most of our attention to the deal negotiation task. We also provide a brief discussion of automating the connecting and exchanging functions. We address agent design in Section 4.

2.2 Formal model

We present a formal model of a marketplace system to focus attention on the important design issues that we review the rest of the chapter. The formal representation is not essential to understand the rest of the chapter, but it provides a concise organization of the main themes. This representation also implicitly suggests (we do not provide extensive references) the links between the topics we cover and the more theoretical literature on mechanism design that we do not review in detail.

Marketplaces can be designed for transactions in goods, services, tasks, plans or other resources and activities. For simplicity, we will refer to these state variables as *goods*, and represent them as a vector of quantities that takes values from a domain X. Given N total agents, an *allocation* is an assignment of a matrix $\vec{x} \in X^N$ to agents $i \in \{1, \ldots, N\}$, with the individual allocation vector to each agent denoted by $\vec{x}_i \in X$.

A market mechanism specifies (1) the goods it recognizes, and (2) rules for determining allocation outcomes. There are typically two types of rules: those specifying a set of permissible actions (strategies), $s_i \in S_i$ for each agent i, and procedures for choosing an allocation based on the observable actions. We denote a mechanism by $\gamma = (s_1, ..., s_N, g(\cdot))$, where g maps the set of actions into allocations, $g : S_1 \times \cdots \times S_N \rightarrow X^N$. We denote the allocation this mechanism makes to a specific agent by $g_i(\vec{s}) = \vec{x}_i$. An example of the rules governing allowable strategies is the set of bidding rules in an ascending auction (for example, bids must be over single lots, and must exceed the previous bid by at least some specified increment). An example of an allocation rule is the English auction rule for unitary objects (the high bidder wins the object, and pays her announced bid).

When a market mechanism is designed, the designer presumably wishes to fulfill some objective subject to various constraints. Let $\vec{\theta}$ be the information state across all agents, defined on the Cartesian product of the individual information spaces Θ_i. Typically the objective can be expressed as a function (often called a *social choice function*) that maps from the information state of the agents to a preferred allocation, as $f : \Theta_1 \times \cdots \times \Theta_N \rightarrow X^N$. However, since elements of the $\vec{\theta_i}$ are private to the agents, and thus not directly accessible to the designer, the ability to achieve this objective depends on the extent to which agents choose to reveal this information, and the cost to the mediator of inducing revelation. Further, the space of possible mechanisms may be constrained by additional social restrictions such as "no external subsidies", or "maintain horizontal equity", which taken together restrict the set of allocations to some permissible space, $f(\vec{\theta}) \in F$.

Agents are distinguished by their possession of *private information* and *autonomy*. We let θ_i denote agent i's private information, which can be taken to include all of the agent's relevant knowledge of or beliefs about states of the world. This information is private in the sense that other agents j do not generally have access to all of the information in the set θ_i. An agent is autonomous if it exhibits *preferences* over allocations, and chooses actions according to some *decision rule*. Assume that i's preferences can be represented by a real-valued function $u_i(\vec{x}_i, \theta_i)$, called agent i's *utility* for a given allocation $\vec{x}_i$ given its private information, which has the property that $u_i(\vec{x}'_i, \theta_i) > u_i(\vec{x}_i, \theta_i)$ exactly when i prefers allocation $\vec{x}'_i$ to $\vec{x}_i$.

An autonomous agent maps its preferences into actions according to decisions that follow from its decision rule. For example, the canonical decision rule assumed by economists for non-strategic settings is that an agent will select feasible actions that maximize its (expected) utility. In strategic settings, one commonly assumed decision rule is that an agent will choose a dominant strategy if one exists. This decision rule is not complete, because it does not specify what to do when no dominant strategy exists. There are many more complete decision rules commonly studied in the literature. For example, every finite game of (incomplete) information has a set of agent strategies that form a mixed perfect (Bayesian) Nash equilibrium [Fudenberg and Tirole, 1991].[3] A corresponding decision rule is that an agent will play a strategy from the Bayes-Nash set. Unfortunately, this decision rule offers a choice for every (finite) problem, but may still be incomplete because there may be multiple equilibria, and then it is necessary to specify *which* Bayes-Nash strategy to play. A particular computational implementation of the agent's decision rule may or may not involve explicit optimization of u_i, or explicit models of beliefs and preferences [Russell and Norvig, 2003].

This formal description of a marketplace system highlights many of the problems that must be addressed by the designer of an automated system. One crucial thing to remember when reading the literature, and when engaging in computational marketplace design, is that some features will be explicitly designed while others will be unspecified (and thus taken as found in the environment in which the system is applied). The performance of the system is likely to depend at least as crucially on the features that

[3] A Nash equilibrium is one in which each agent is playing a strategy that is a best response to the strategies of the others: that is, all strategies are mutual best responses. Bayes-Nash equilibrium means that when information is incomplete, players update their beliefs as new information arrives in the game according to Bayes' Rule, and then play Nash strategies with respect to their expectations.

are not designed as those that are.

We defined a role for goods, mechanisms and agents. We now use the model to illustrate with just a few examples the design issues for these features in turn.

Design and goods. One issue for the designer is which goods the system will recognize, and in particular whether the domain of the goods a market mechanism allocates corresponds to the domain of the goods over which agents have preferences. For example, externality problems (such as pollution) have long been characterized as problems of "missing markets" for certain goods. Some computational markets are designed specifically to enable allocations of goods that matter to agents but which are not generally traded in spontaneous (undesigned) markets (see, e.g., Ledyard and Szakaly [1994]). The issue of which goods to transact is central to the interest in combinatorial mechanisms, which we discuss below in Section 3.2.2.

Designing market mechanisms. An important issue for automated market designers is to decide which of the several market mechanism features to design, and which to leave unspecified. Recall from Figure 2 that market transactions require mechanisms for mediating the functions of connecting, dealing and exchanging. Much of the market mechanism literature focuses on the dealing function: the determination of terms of trade and the assignment of allocations. Even when there is attention to mechanisms for the other functions, the most common approach is to define for each function as a separate entity; thus in Section 3 we discuss separately design for each. However, ignoring interactions between the mechanisms may result in inefficiencies and failures. We expect that as automated market design matures we will see increasing attention to mechanisms that integrate more of these several necessary functions.

Recall also (see Figure 2) that a marketplace system operates in the context of a problem environment, consisting of technological and institutional constraints. Other institutions are features of the environment that restrict the set of possible mechanism designs. In this chapter we treat other institutions and technology as given, and immutable by the designer; e.g., laws, common languages, government structures, CPU capabilities. The institutions restrict feasible mechanisms to some space, Γ. The design problem is to configure a feasible mechanism $\gamma \in \Gamma$—that is, to define a set of goods over which agents can deal, rules specifying permissible actions, and rules mapping actions to allocations—that implements the constrained social choice function $f(\vec{\theta}) \in F$. Designers of computational markets thus need to either implicitly or explicitly make assumptions about laws, languages and other social institutions necessary to support transactions.

Designing agents. We defined agents by their information, preferences and decision rules. Each raises important design considerations. For example, to predict the performance of a particular mechanism the designer must make assumptions about the decision rule the agents will follow when interacting with the mechanism, which in turn depends on assumptions about agent information and preferences. When building automated agents themselves, the designer must deal with information acquisition, storage and processing problems (for example, to compute Bayesian updates or other predictions of relevant events). Designers must endow agents with feasible algorithms to implement their decision rules, which is no small matter in some settings. For example, in many market mechanisms—such as the ubiquitous multiple simultaneous ascending auctions—it is generally computationally infeasible to determine Bayes-Nash strategies. We discuss this problem and a method for analyzing agent strategies for intractable mechanisms in Section 5.

2.3 Possibilities and impossibilities

It is often difficult in complex problem environments to find a mechanism that implements the desired objective while satisfying even a few, seemingly reasonable constraints on the social choice functions that it implements. Indeed, in very general classes of problems, no such mechanisms of any sort exist. A considerable body of design theory characterized the space of social choice functions that are possible; this literature provides crucial guidance for the design of computational markets.

For example, suppose the marketplace designer wants a system that will satisfy the following rather weak requirements (we use $f_i(\vec{\theta})$ to denote the subset of allocation $f(\vec{\theta})$ received by agent i):

1. Ex-post efficiency (Pareto optimality): For no profile $\vec{\theta} = \{\theta_1, ..., \theta_N\} \in \Theta_1 \times \cdots \times \Theta_N$ is there an $\vec{x} \in X$ such that $u_i(x_i, \theta_i) \geq u_i(f_i(\vec{\theta}), \theta_i)$ for all i, and $u_i(x_i, \theta_i) > u_i(f_i(\vec{\theta}), \theta_i)$ for some i. That is, there is no alternative outcome in which at least one agent would be better off, and no agent would be worse off.

2. Ex-interim participatory efficiency (individual rationality): Suppose agent i begins with an endowment of goods, $\omega_i \in \Omega_i$, which it keeps if it refrains from participating in the mechanism. Then $u_i(f_i(\vec{\theta}), \theta_i) \geq u_i(\omega_i, \theta_i)$ for every i. That is, agents must be willing to voluntarily participate given the rules of the mechanism and their private knowledge about their own situation (before the final allocation is revealed).

3. No subsidies: The mechanism does not require any external injection of resources (e.g., payments to agents) that are not obtainable through the allocation of endowments $\Omega_1 \times \cdots \times \Omega_N \to X$.

Myerson and Satterthwaite [1981] showed that in general for a bilateral exchange problem there is no mechanism that satisfies (1)–(3) (if agents are assumed to use a Bayes-Nash strategy as their decision rule).[4]

Given this strong impossibility result, designers must choose ways in which to relax the design requirements. Three typical approaches are to (1) assume (or impose if under designer control) agent preferences that are more tightly restricted in the space of rational preferences; (2) assume or impose that agent decision rules are restricted more narrowly; or (3) relax some of the social choice constraints on an acceptable mechanism.

[4] A bit more precisely, the result holds for at least bilateral trade between agents each of whom is autonomous and self-interested, has private information about its own value, satisfies Bayes-Nash rationality, and for whom the support for those valuations overlap.

Two of the more important constructive results are the Vickrey-Clarke-Groves (VCG) family of social efficiency maximizing mechanisms and the Maskin-Riley revenue maximizing mechanism. In VCG (discussed in more detail in Section 3.2.2, below), agent preferences are restricted to those that can be expressed as quasilinear utility functions, and the "no subsidies" constraint is abandoned. The Maskin and Riley [1989] mechanism limits the space of goods, and replaces ex-post efficiency with revenue maximization (that is, the social choice function depends on the preferences of only one agent, the seller).

3 Automating market mechanisms

We organize our review on the design of computational market mechanisms to follow the three stages in our diagram of the canonical transaction problem: discovery, negotiation and execution (see Figure 1).

We focus primarily on computational mechanisms for *negotiation*, or making the deal, which is the second of the three steps. This focus largely reflects the bias in the economics field, which is most relevant for the audience of this book. However, we think it is important to recognize the plethora of ancillary services that must also be provided to support trading. Each is potentially subject to automation as well. As agent-based computational systems mature, we hope to see increasing attention to the design of mechanisms for connecting and exchanging. These are relatively open-ended problems, with services often provided by third parties outside the scope of a particular marketplace, as well as within the marketplace itself.

In the first subsection we provide a brief overview of some discovery facilities to illustrate some of the opportunities provided by the online medium, as well as requirements for operating a successful marketplace. In the second and longest subsection we discuss in some detail research on the design of computational mechanisms for deal negotiation (the "market" to many, though we use the term more expansively to describe all three functions). In the third subsection we survey briefly a few systems to facilitate transaction execution. The need for additional attention to discovery and execution as problems of market design should become evident.

3.1 Connecting: Discovery services

At a bare minimum, marketplaces must support discovery to the extent of enabling users to navigate the opportunities available at a site. More powerful discovery services might include electronic catalogs, keyword-based or hierarchical search facilities, and the like. The world-wide web precipitated a resurgence in the application of information retrieval techniques [Belew, 2000], especially those based on keyword queries over large textual corpora.

Going beyond generic search, industry groups proposed a variety of standards for describing and accessing goods and services across organizations. Examples include languages extending XML with commerce-specific constructs [Hofreiter et al., 2002]), and protocols and registration infrastructure supporting web services [Curbera et al., 2002]. Some recent proposals suggested using *semantic web* [Berners-Lee et al., 2001] techniques to provide matchmaking services based on inference over richer representations of goods and services offered and demanded [Di Noia et al., 2004; Li and Horrocks, 2004].

The task of discovering commerce opportunities inspired several innovative approaches that go beyond matching of descriptions to gather and disseminate information relevant to comparing and evaluating commerce opportunities. Here we merely enumerate some of the important service categories:

Recommendation. [Resnick and Varian, 1997; Schafer et al., 2001]. Automatic recommender systems suggest commerce opportunities (typically products and services to consumers) based on prior user actions and a model of user preferences. Often this model is derived from cross-similarities among activity profiles across a collection of users, in which case it is termed *collaborative filtering* [Resnick et al., 1994; Hill et al., 1995; Riedl and Konstan, 2002]. A familiar example of collaborative filtering is Amazon.com's "customers who bought" feature.

Reputation. When unfamiliar parties consider a transaction with each other, third-party information bearing on their reliability can be instrumental in establishing sufficient trust to proceed. In particular, for person-to-person marketplaces, the majority of exchanges represent one-time interactions between a particular buyer and seller.

Reputation systems [Dellarocas, 2003; Resnick et al., 2002] fill this need by aggregating and disseminating subjective reports on transaction results across a trading community. One of the most prominent examples of a reputation system is eBay's "Feedback Forum" [Cohen, 2002; Resnick and Zeckhauser, 2002], which some credit significantly for eBay's ability to achieve a critical-mass network of traders.

Comparison shopping. The ability to obtain deal information from a particular marketplaces suggests an opportunity to collect and compare offerings across multiple marketplaces. The emergence on the web of *price comparison services* followed soon on the heels of the proliferation of searchable retail web sites. One early example was BargainFinder [Krulwich, 1996], which compared prices for music CDs available across nine retail web sites. The University of Washington ShopBot [Doorenbos et al., 1997] demonstrated the ability to automatically learn how to search various sites, exploiting known information about products and regularity of retail site organization. Subsequent research systems emphasized issues such as adaptivity to user preferences Menczer et al. [2002]. Today's shopping engines employ direct data feeds from product vendors, and provide standard interfaces with typically price-based product rankings.

Auction aggregation. The usefulness of comparison shopping for fixed-price offerings suggested that similar techniques might be applicable to auction sites. Such information services might be even more valuable in a dynamically priced setting, as there is typically greater inherent uncertainty about the prevailing terms. The problem is also more challenging, however, as auction listings are often idiosyncratic, thus making it difficult to recognize all correspondences. Nevertheless, several auction aggregation services (BidFind, AuctionRover, and others) launched in the late 1990s. Concentration in the online auction industry and resistance from auction sites has combined with the difficulty of delivering reliable information to limit the usefulness of such services, however, and relatively few are operating today.

3.2 Dealing: Negotiation mechanisms

Negotiations are the major component of many computational market institutions (see Figure 2), and probably the component that received the most research attention. We use the word "negotiation" to refer to any process through which potential traders come to agreement on the terms of a deal. This includes a range of practices, from two agents haggling over price in a bazaar to a standard retail transaction in which the selling agent posts a fixed price and the buying agent says either "yes" or "no".

Computational negotiation mechanisms often involve a mediator: an entity that collects offer messages from the potential traders, and facilitates the mapping of those messages into an outcome. Well-known non-computational examples include an auctioneer and a market maker on a stock exchange floor. Auction web sites such as eBay are the best known examples of mediation in computational markets. In general a mediator may have a stake in the outcome (e.g., as party to transactions, or through commissions), in which case it also plays the role of an agent. However, to sharpen the distinction we maintain a strict separation between the agent and mediator roles, modeling the latter as following a fixed policy determined by the mechanism designer. For example, an eBay auction is mediated by the process that receives and validates bids, following the specified eBay rules for showing the current high bid, and determining the final winner and price.

In this section we discuss research on the design of mediated computational negotiation mechanisms. We start with a review of designs (and some implementations) for a smorgåsbord of domain-specific applications, ranging from computer file systems to energy markets to belief aggregation. We describe the main goals, assumptions and some results, without attempting to be comprehensive or exhaustive. We selected applications areas because they are significant in the historical development of thought in this area, or because they received intensive research attention in recent years.

We then turn to the large body of recent work that focuses on mostly technical questions arising from the design of an important class of computational markets: combinatorial mechanisms. We give extra attention to this particular area of the market design theory literature because it emerged from important real market design applications (most notably public spectrum auction), it attracted the attention of many top researchers in both economics and computer science, and it represents an important area at the current leading edge of research. Further, many of the problems that arise in other settings are similar to those in combinatorial markets, so it is a good representative for other bodies of literature we have insufficient space to review.

3.2.1 Smart markets for domain-specific applications

There are many computational markets in use. Most research, with some exceptions, concerned designs that have not (yet) been implemented. One important exception has been a recent surge in matching markets for solving various social problems. The best known is the medical resident matching market in the U.S. [Roth and Peranson, 1999]. Related field work is underway, though not yet complete, for markets to match pairs of potential kidney donors [Roth et al., 2005] and to match students to public schools [Abdulkadiroglu et al., 2005]. More often, due to the high cost of implementing test markets, empirical research to evaluate performance is carried out through human subject laboratory experiments on stylized instances of the designs, or through numerical computer simulations.

In the remainder of this section we discuss a number of computational negotiation mechanisms—only some of which have been implemented in the field—designed for specific domains. We call these "smart markets", following Vernon Smith [McCabe et al., 1991], because nearly all of these mechanisms involve a nontrivial computation on submitted offer messages to determine the outcome. Thus, we do not discuss the negotiation mechanisms that underlie markets such as eBay, because they are simple enough to not require any special computational capabilities. Indeed, such negotiation mechanisms are notable for mimicking non-computational auctions and other market forms that have been common for centuries.[5]

Allocating computational and communication network resources. Given that computer scientists directly confront allocation problems involving computational resources (e.g., sharing bandwidth, CPU cycles, file space), it is perhaps unsurprising that much research in computational market mechanisms has targeted such problems. This reflexive phenomenon has been important for development of the research community. Over time, a number gravitated towards principles from economics: the discipline most focused on the analysis of resource allocation questions. More or less contemporaneously, economists interested in computationally-intensive mechanisms began picking up ideas from computational science. Mechanism design for network and computational resources became an early meeting ground for economists and computer scientists, and much of the research began to exhibit cross-disciplinary approaches, often supported by cross-disciplinary collaboration. These early efforts resulted in important learning about the interaction between incentives theory and computational method that informed much of the more recent negotiation design research in other domains.

Several computer scientists in the 1980s focused on the possibility of applying market-mediated transactions to allocate computational resources.[6] These projects drew attention to the problem of defining the goods over which a computational market negotiates. There are many levels of abstraction and aggregation at which computing resources and services could be specified; to create an automated market it is necessary to explicitly specify the set of goods. Among these early studies were investigations of the problems of specifying markets for file space [Kurose and Simha, 1989], communications channels [Kurose and Simha, 1986], and CPU loads [Ferguson et al., 1988].

[5]Other features of eBay and similar online auctions, such as search facilities and reputation management, *do* make innovative use of Instead, we focus on negotiation mechanisms that for the most part are infeasible to operate without computer automation.

[6]Ironically, an early market for time-sharing computer resources was implemented at Harvard without computational support, with bids and schedules posted by hand on a bulletin board [Sutherland, 1968].

In a novel approach to allocating scarce computing resources, Brewer [1999] proposes a "computation procuring clock auction" which addresses the challenge at the level of a market for problem solutions, rather than a market for problem-solving resources. In Brewer's mechanism a mediator poses a computationally costly problem and agents offer approximate solutions. Thus, the computational market effectively creates a decentralized "computer" out of the participating agents. At any instant the market displays the current best solution to the problem of interest. Agents can then submit improved solutions; they are paid some fraction of the improvement in the objective function. The auction ends when a defined interval passes without new solution submissions. Brewer obtained positive results in human subject experiments, using a complex train scheduling from another smart market as the problem to be solved.

The academic research Internet rapidly grew and made the transition to the commercial Internet in the early 1990s. For several years, usage (traffic) doubled approximately annually, outstripping (physical and technical) increases in the network. Congestion became a significant problem, and engineers were concerned that with continued growth the Internet would collapse. From these conditions emerged a quite large literature on designing computational markets for allocating bandwidth. The early work focused on characterizing the economics of bandwidth congestion and the potential benefits from a designed market [Cocchi et al., 1993; Shenker, 1994; MacKie-Mason and Varian, 1994a, 1995a]. Congestion is an externality: that is, a given user putting a load on the network does not directly bear the cost of additional congestion experienced by others. Thus in general the allocation of bandwidth resources by a market will be inefficient unless the market is specifically designed to internalize the congestion externality.

Internet traffic is transported using packet-switching; by contrast, voice networks switch circuits. MacKie-Mason and Varian [1994a, 1995b, 1996] explored the implications of the Internet's architecture for good market design, and proposed a mechanism designed specifically for packet networks that would allocate congested bandwidth to packets. Their mechanism charges a positive price for packets when there is congestion (and zero otherwise), respects agents' autonomy and private information, and obtains an efficient allocation despite the congestion externality. This mechanism is a smart market that necessarily depends on a high degree of automation to process agent messages, determine the allocation, and implement the allocation. This computational market is a Generalized Vickrey Auction [MacKie-Mason and Varian, 1994b], which is a feasible instance of a Vickrey-Clark-Groves (VCG) mechanism designed specifically to handle externalities. This is the first proposal for a VCG mechanism we have found for computational markets; the later literature on combinatorial markets extensively explores VCG mechanisms, as we discuss below.

Other mechanisms proposed for congestion priority allocation include [Cocchi et al., 1993; Gupta et al., 1996; Korilis et al., 1995]. These and the Generalized Vickrey Auction have various difficulties with the matching of the domain of allocations offered in the market to the domain of agent preferences. Some proposed mechanisms are specific to allocating packets, but generally users have preferences defined over sessions or flows with many (sometimes many thousand) packets. Further, all of these proposals were for static allocation markets, but user preferences generally encompass schedule and other time dependencies.

Well over one hundred papers were published about computational markets for network bandwidth in the ensuing decade. One important topic addressed early was the design of markets to allocate multiple qualities of service (rather than merely congestion priority); see, e.g., [Cocchi et al., 1991; Shenker, 1995; MacKie-Mason et al., 1996b]. Mechanisms were designed for networks with virtual circuits [MacKie-Mason et al., 1996a; Thomas et al., 2002; Kelly et al., 1998].[7] Others developed computational mechanisms for cost-sharing network services that generate joint costs, such as multicasting [Moulin and Shenker, 2001; Feigenbaum et al., 2001]. Chen [2003] tested some of these mechanisms with human subjects. Some work addressed additional problems that arise in markets for network services that support mobile users [Mullen and Breese, 1998]. Recent work in "distributed algorithmic mechanism design" obtains results for a mechanism to assign interdomain routing that is constrained to be backwards compatible with existing Internet communication protocols [Feigenbaum et al., ming].

Recently there has been renewed interest in computational markets for other computational resources. In particular, in the late 1990s several authors explored markets for CPU resources. This research responded to the observation that most CPU cycles available from desktop computers and workstations are unused. For a price, computer owners might be willing to let others run programs on their machines. Researchers explored market designs for CPU markets on networks of workstations [Amir et al., 2000; Gagliano et al., 1995; Waldspurger et al., 1992], as well as the broader Internet [Amir et al., 1998; Regev and Nisan, 1998]. Recent work introduced market models for peer-to-peer [Gupta and Somani, 2004; Cox and Noble, 2003] and grid computing [Wolski et al., 2001].

The other significant strand of computational market design for computational resources focused on providing file system services. Specific applications include markets for distributed databases [Stonebraker et al., 1996]; Web servers and web caching [Karaul et al., 1998; Kelly et al., 2005, 1999]; and data replication [Anastadiadi et al., 1998].

Energy markets. Computational mechanisms have been employed for electric generation in England, California, France, New England, and other locations. In an important study, given the paucity of empirical evaluations of implemented markets, Wolfram [1998] studies the behavior of (non-automated) bidders in the automated daily generating capacity auction in England. This is a multi-unit uniform-price mechanism; Wolfram finds that bidders strategically manipulate their bids in accordance with theoretical predictions about this mechanism design, resulting in less than optimal social efficiency. Cameron and Cramton [1999] analyze some of the institutional details of market implementations in California, and their implication for efficiency. Nicolaisen et al. [2001] develop a simulation model of electricity markets, relating efficiency and market power to mechanism microstructure. Ygge and Akkermans [1996] design a computational market mechanism for power load management, where agents representing individual devices present demands, responding competitively to price changes. See Marks [2005] for an extended discussion of agent-based simulation models applied

[7] Virtual circuits are a blend of packet- and circuit-switching technology of which asynchronous transfer mode (ATM) is the best known example.

to energy markets.

A joint market for natural gas supplies and transportation was designed and evaluated with human subject experiments by McCabe et al. [1989]. The market calls for sealed, one-shot bids. Wholesale buyers and wellhead producers submit location-specific offers, and pipeline owners submit link-specific capacity offers. The smart market solves a linear programming problem for the network, sets uniform prices and assigns a consistent allocation of gas and transport that maximizes social surplus (given the constraints of the market design). In experiments the market achieved 90% or higher efficiency, and marginal bids were approximately truthful (thus fulfilling a price discovery role). However, inframarginal bids were substantially below truth values, and the authors point out that the theoretical literature predicts an equilibrium for this market that is not truth-revealing and thus is less than fully efficient.

Scheduling. Resource allocation with time contingencies is known as a *scheduling* problem. There is a huge research literature on the centralized solution of scheduling problems. A simple keyword search yields over 1500 references, covering many varieties of scheduling problems distinguished by constraints, objectives, and information available. Recently a few authors have started to develop market solutions to scheduling problems, addressing the decentralized structure of many scheduling environments.

In traditional scheduling problems, agents submit their bids for time-indexed resources in advance, the mediator applies the mechanism allocation function, and the schedule is announced, then implemented [Nisan and Ronen, 2001]. Time dependencies almost always lead to complementarities in preferences; Wellman et al. [2001a] analytically compare three designs along a spectrum of matching the domain of allocations to the domain of preferences (separate markets, restricted package markets, and fully combinatorial VCG mechanism). Train scheduling is one application for which specific markets have been designed and tested (with human subject experiments) [Brewer and Plott, 1996, 2002].

Another interesting category for computational markets contains *online* scheduling problems: the inputs arrive sequentially, and allocations are made dynamically, before all of the inputs are known. At any given moment there is a set of jobs that want to use the resource. One difference from offline scheduling is that some or all of the resource may already be in use, facing the mediator with a decision whether to preempt a running task. Another difference is that new bids for service may arrive in the future, creating an option cost of committing current resources to current job requests.

Online scheduling problems highlight a problem caused by uncertainty. The economic objective in an online problem usually involves some sort of expected value maximization. In deterministic problems, it is relatively straightforward to evaluate the performance of a particular allocation rule given the agents' (static) private information. With uncertainty, the outcome also depends on the future evolution of these state variables. Some of these stochastic processes themselves may be endogenous to the problem: for example, the arrival of new requests may depend on the current allocation decisions by the mediator. This only complicates what is already typically an intractable (NP-hard) optimization problem.

Due to the complexity, there are few results on markets that maximize expected value for online scheduling problems. The smart markets proposed typically implement heuristic allocation rules, for instance pre-empting a currently running job if a new request has an estimated expected value greater than some threshold. Two recent contributions provide some hope for traditional mechanism designs (that maximize a social objective function) in online scheduling problems. Friedman and Parkes [2003] define a class of problems for which a "delayed Vickrey-Clarke-Groves mechanism" has a dominant strategy equilibrium. Parkes and Singh [2003] show that an online mechanism design problem can be formulated and solved as a Markov Decision Process (MDP) problem, and they define a mechanism in which there is an approximately efficient (though computationally intractable) Bayes-Nash equilibrium.

Most of the online scheduling literature has avoided the complexity problems by focusing on minimax optimization, that is, reaching lower bounds for worst case performance. Two teams established that the best ratio achievable for worst case online scheduling performance (in centralized (non-strategic) problems) relative to full-information (offline) scheduling is $(1+\sqrt{k})^2$, where k is the maximum ratio between the value per time unit of any two jobs [Baruah et al., 1992; Koren and Shasha, 1995]. These authors also provide algorithms that reach these bounds. Two recent approaches construct market solutions for strategic agents. In one the worst case ratio is increased to only $((1+\sqrt{k})^2+1)$ [Porter, 2004]; the other addresses a somewhat different question, but also provides constructive results [Hajiaghayi et al., 2004].

Belief discovery and aggregation. One of the benefits of market allocations is the discovery of value information. Of particular interest, markets for securities whose value depends on the future realization of a random variable will aggregate beliefs about the outcome, and thus provide a predictor. For example, it has long been known that well-functioning financial markets provide excellent predictors of the underlying asset values [Forsythe and Lundholm, 1990; Plott and Sunder, 1988]. Forsythe et al. [1992] implemented and studied the long-running Iowa Electronic Market, in which agents bid for securities that pay off on the results of political events (e.g., presidential primaries) and other well-defined events such as corporate earnings announcements. This market has routinely forecast political outcomes more accurately than professional polling organizations.

Standard financial markets introduce independent auctions for each security, which presents scaling problems when there are a large number of uncertain propositions. Pennock and Wellman [2000] establish conditions under which probabilistic dependence structure can or cannot reduce the number of securities needed for an operationally complete market. Hanson [2003] addresses the problem by defining a hybrid between pure markets and the evaluation methods sometimes used to score probability assessors. His *market scoring rules* exhibit properties of a market when there exists sufficient activity, reverting to the properties of scoring rules in cases of low liquidity. This market was implemented as a DARPA experiment to aggregate public information relevant to national security concerns [Polk et al., 2003], but days before trading began it was halted due to political uproar. Inspired by market scoring rules, Pennock [2004] introduced a dynamic pari-mutuel market for information aggregation that exhibits guaranteed liquidity, no risk to the mediator, and continuous updating of information.

3.2.2 Combinatorial markets

Problems with complementarities. Complementarities in demand are one of the more common causes, at least in the research literature, for the complexity that calls for smart markets. Goods are complements when acquisition of one increases demand for the other. In such a case, an agent's willingness to pay for one good will depend on whether or not the other can also be obtained. Many problems have this feature. For example, a take-off slot is worth little if the airline cannot also secure a landing slot. One hour of job-shop time may be worth zero if the firm cannot obtain the second hour necessary to complete the job. Fast delivery of the first packet in a file or email delivery is worth little if the remaining packets are delayed.

When goods are complements, a standard competitive price equilibrium may not exist [Bikhchandani and Mamer, 1997].[8] Even when one does, standard price-formation protocols are not guaranteed to find it [Scarf, 1973]. The fundamental problem is that when markets operate by separately forming prices for each good, agents cannot directly express information concerning value complementarities. Using the language from our conceptual framework, the domain of goods allocated by the mechanism does not match the domain of goods over which agents have preferences. For example, consider two simultaneous sealed-bid auctions, one each for goods A and B. An agent who jointly values the goods at \$3, but who values each separately at \$0, might be willing to pay \$1 for good A if it can also purchase good B for no more than \$2, but not otherwise. However, in this auction market the agent can bid for A at \$1, but cannot ensure that if it wins it can simultaneously purchase B for \$2 or less.

A direct response to this mismatch between the agent's preference domain and mechanism's allocation domain is to design mechanisms that allocate a domain of goods better aligned with agent the domain of agent preferences. Many authors pursue this through the design of combinatorial mechanisms.

Aligning the scopes of mechanism allocations and agent preferences does not, it turns out, solve the design problem. There are two types of difficulties. First, as shown by Myerson and Satterthwaite [1981], for a surprisingly broad set of problems, it is impossible to design mechanisms that satisfy minimally desirable constraint sets. Then, though all else equal some combinatorial mechanisms may outperform non-combinatorial options, the problem remains of choosing among the possible second-best combinatorial mechanisms, which may be unbounded in number. The second difficulty is that all else is not equal: when we take into account the computational and other costs of combinatorial mechanisms, non-combinatorial mechanisms may better achieve the designer's objective. We shall discuss these two problems, and then some highlights of the literature that developed around them.

The first problem is that in a broad class of problems there exists no Bayesian-Nash mechanism that is efficient, individually rational, and budget balanced (see Section 2), but generally two of these can be satisfied at the expense of the third. Therefore, designers typically choose which property to sacrifice, and then try to limit loss on that

[8] A "standard" competitive price equilibrium is a vector of unit prices and corresponding feasible allocation such that each agent receives the quantities it desires taking these prices as given.

dimension. As an alternative, a designer might give up one of these criteria but offer a mechanism that satisfies the other two plus some other desiderata. Thus, the intuition to design combinatorial mechanisms when agents exhibit complementary preferences is only the first search step through a vast design space: the quality of a design depends in a strong way on the designer's objective and desired constraints. There may be many or zero combinatorial mechanisms that are best.

The second problem is that mechanisms implemented for actual use inevitably incur transaction, computation, and cognitive costs that are often ignored in theoretical analyses. Computational costs include most directly the complexity of solving combinatorial optimization problems, but also the communication complexity of transmitting offers over many possible bundles. Cognitive costs include the burden of constructing offers over such bundles. Transaction costs include delays, coordination effort, and other costs of addressing multi-dimensional allocation domains in a single overarching mechanism. These implementation costs create standard economic tradeoffs (largely ignored by mechanism design economists) between the advantages of combinatorial mechanisms and their inherent diseconomies of scope. The potential benefits of aligning mechanism allocation domains with agent preference domains, along with the computational challenges, motivated a surge in mechanism design research by computer scientists [Dash et al., 2003; Nisan and Ronen, 2001; Papadimitriou, 2001; Rosenschein and Zlotkin, 1994]. This line of work has begun to address some of these additional costs, however we are unaware of any work that presents a reasonably complete and explicit model of the overall design tradeoffs.

Combinatorial market design. A combinatorial auction specifies rules for permissible messages that express values over combinations of goods, and an allocation function over these messages that assigns combinations. See de Vries and Vohra [2003] for a good survey; Cramton et al. [2005] collect articles by many of the leading researchers on this topic, presenting an in-depth review of technical issues. We can only briefly introduce this huge literature. We highlight crucial issues for computational market design and open research questions.

Combinatorial mechanisms are motivated in part by the Arrow-Debreu theorem, which establishes that if markets span the complete domain of agent preferences, a competitive equilibrium exists and is efficient [Arrow, 1964; Debreu, 1954]. However, a full set of Arrow-Debreu markets, including markets for all bundles of interest to agents, is not sufficient for two reasons. First, when preferences exhibit complementarities, the conditions of the Arrow-Debreu theorem are not met and a competitive equilibrium may not exist [Bikhchandani and Mamer, 1997]. Second, designers are often concerned with strategic (non-competitive) situations as well. The most important motivation for computational market design when agents are strategic is a result due to Vickrey, Clarke, and Groves: a direct revelation mechanism that guarantees an efficient, individually rational allocation [Vickrey, 1961; Clarke, 1971; Groves, 1973]. In a direct revelation mechanism, agents announce to the mediator their preferences over allocations; in the VCG family of mechanisms, the scope of allocations is the same as the scope of agent preferences. For our discussion of combinatorial mechanisms we focus on VCG-based mechanisms.[9]

Based on the number of papers solving implementation design problems for VCG mechanisms, it might appear that researchers view the VCG as an ideal form. In general, it is not. First, VCG does not overcome the Myerson and Satterthwaite [1981] impossibility result: a VCG mechanism that is guaranteed to be efficient and individually rational will not in general be budget balanced. Indeed, in bad cases, for N agents the VCG can require a subsidy on the order of $N-1$ times the total surplus of the final allocation.[10] Second, although individual rationality and efficiency is a plausible set of minimally desirable criteria, other criteria may be desirable for some allocation problems. For example, VCG payments are typically "discriminatory": different agents likely make (or receive) different payments for the same allocation. In some settings social norms or other goals may impose a non-discriminatory constraint.[11] Third, there are substantial concerns about the computational feasibility of VCG mechanisms in moderately complex problems. The practical problems proved to be so numerous, and thus far, sufficiently intractable, that almost no VCGs are implemented in observed practice. We now discuss these feasibility concerns.

One computational design issue is the *winner determination* problem: how to compute the allocation function $g(S_1, S_2, \ldots, S_N)$ (see Section 2.2)? For a general combinatorial problem, the VCG computation requires $N - 1$ separate solutions of an NP-hard set-packing problem [Rothkopf et al., 1998]. Known algorithms for NP-hard problems have worst-case exponential runtimes: the computational cost effectively doubles with each additional good.[12] One line of research focused on developing algorithms with good average-case performance on representative problem classes [Leyton-Brown, 2003; Sandholm and Suri, 2003]. A second logical approach is to find an algorithm that is guaranteed to find an *approximate* solution to the VCG allocation in polynomial time. However, Nisan and Ronen [2000] demonstrate that approximate (non-optimal) but polynomial (computationally feasible) VCG-based mechanisms that are truthful have arbitrarily bad performance in the worst case. Yet a third approach is to impose sufficient restrictions on agent rationality (or permissible strategies) to enable mechanisms that implement the VCG outcomes exactly with feasible computations [Parkes and Ungar, 2002]. Another line of research studied problems in which

[9] VCG mechanisms maximize Marshallian social welfare, which is the unweighted sum of surpluses (value net of any payments) for all buying and selling agents, measured in some common unit such as dollars. Another common design goal is to maximize the seller's revenue. Most of the points we make about VCG mechanisms are qualitatively true for revenue-maximizing mechanisms as well, though of course the details are different. The literature on revenue maximizing mechanisms over complementary goods is much less developed than that for VCG mechanisms.

[10] Roughly speaking, the VCG pays to each agent the value of the surplus that the agent's value creates by its participation in the final allocation. Consider a problem in which the participation of all agents is necessary for any positive value to be created (a coordination, or joint production problem). In this case, if a total value of S is created, the VCG pays NS in total, of which only S is financed by the surplus created through the allocation.

[11] Much has been written over the years about the social ethics of discriminatory prices. In practice they are common: for example, students generally pay less for movie tickets than do their professors. Nonetheless, non-discriminatory pricing is sometimes imposed, particularly for public projects. For example, in designing a computational market for the provision of evaluations, such as product reviews, Avery et al. [1999] require that the same action (timing of an evaluation) must be paid the same price.

[12] For example, the FCC simultaneously auctioned 1472 licenses in one 1996 auction. The total number of possible combinations to consider in a fully combinatorial allocation function would have been $2^{1472} - 1$.

there is a structure on the space of goods that provides sufficient simplification to make the winner determination problem tractable [Rothkopf et al., 1998; Wellman et al., 2001a].

A symmetric problem is that of *preference elicitation*: extracting value information from agents without imposing an undue or infeasible burden. Given a fully combinatorial allocation space, agents must determine and express an exponential number of valuations. For example, with only 30 distinct goods, there are $2^N - 1$ (over a billion) possible bundles for which to bid.

A number of authors investigate the communication complexity of various resource allocation mechanisms. For a convex economy, the Walrasian mechanism is the unique individually rational mechanism that is informationally efficient (minimizes the dimensionality of the message space necessary to verify a Pareto efficient allocation) [Jordan, 1982; Hurwicz, 1960; Mount and Reiter, 1974].[13] Among other things, for an economy to be convex preferences must be sub-additive (which rules out complementarities between goods), and continuous (which rules out integer constraints), and thus many interesting problems cannot be treated as convex. Unfortunately, the results are somewhat negative for non-convex economies. Nisan and Segal [2004] show that any efficient mechanism must communicate at least as much information as a full revelation of one agent's preferences, which will in general be exponential when agents have preferences over combinations of goods. They further prove that even approximately efficient allocations are hard: To guarantee an improvement over the approximation represented by selling all of the items as a single bundle requires communication that is exponentially increasing in the number of goods. This is true in a worst-case analysis, and also in expectation for at least some probability distributions over agent valuations.

Although the preference elicitation problem is provably hard, a number of authors have worked on pragmatic approaches to making it manageable for some problems. For example, some researchers address this problem by designing iterative, or *progressive* combinatorial auctions [Ausubel and Milgrom, 2002; Parkes, 1999; Parkes and Ungar, 2000; Wurman and Wellman, 2000], in which agents are expected to bid on each iteration only on bundles that appear best given the current information. Recently, some proposed methods based on explicit queries [Conen and Sandholm, 2002], where agents are asked their values for particular bundles based on the auction's defined query policy for its current state. There are a variety of related approaches to the elicitation problem [e.g., Faratin and de Walle, 2002; Conen and Sandholm, 2001; Parkes, 2004]. One is to develop bidding languages that are natural and concise for human agents [Boutilier and Hoos, 2001].

A different approach is to identify special problem classes that require less complete expressions of preferences. For example, Bikhchandani et al. [2002] focus on settings where "agents are substitutes": the contribution to problem value of a group of agents is more than the sum of their individual contributions. In such cases, agents can describe their preference over a smaller number of bundles, and communication and computation are polynomial (requiring the solution of two linear programs). Another class of examples are problems in which valuations satisfy the gross substitutes property [Kelso and Crawford, 1982]: a Walrasian equilibrium exists [Gul and Stacchetti,

[13] A *Walrasian* mechanism is one that yields a competitive equilibrium; see footnote 8.

1999] and it can be found with polynomial communication [Nisan and Segal, 2004].[14]

Another pragmatic concern for VCG mechanisms (as well as many others) is their susceptibility to the often unenforceable assumption that agents do not collude. In our conceptual framework this assumption is represented by limiting communications to the links between agents and the mediator (see Figure 2). Specifying mechanism rules that forbid collusion does not necessarily prevent it. VCG mechanisms perform arbitrarily badly when agents can collude [Ausubel and Milgrom, 2002]. A related concern is their vulnerability to "false name" bids (one agent splitting package bids between multiple pseudonyms to change the allocation or associated payments) [Sakurai et al., 1999].

Despite the known problems with combinatorial mechanisms, they have been tested in a number of laboratory experiments in which the space of goods was small enough for the computations to remain tractable. For example, Rassenti et al. [1982] developed a sealed-bid combinatorial auction to allocate airport runway time slots. Their specification of goods allowed for agents to express preferences over packages of multiple slots to accommodate complementarities (for example, needing a landing slot to combine with every take-off slot). They implemented an algorithm to determine the allocation that maximized system surplus, then awarded packages at prices guaranteed to be no more than the amounts bid. They tested this smart market negotiation mechanism in a laboratory setting with cash-motivated human subjects, where it obtained about 10% higher efficiency than a mechanism of independent auctions for each slot.

NASA funded a team of Caltech economists to study various computational market designs to allocate payload space, power, and other resources for commercial experiments in the space station program. Banks et al. [1989] report on several designs and human subject experimental tests of their performance. As in Rassenti et al. [1982], the designs were driven by the specification of the goods over which negotiations were defined. They addressed problems with multiple resources (space, power), uncertainties in demand and supply (for example, some shuttle launches are cancelled), unresponsive supply (no inventories and fixed capacities), and demand indivisibilities. They tested two smart market negotiation mechanisms: one an iterative approximation to a Vickrey-Clarke-Groves mechanism, and the second a simpler iterative package bidding process. Traditional markets averaged only 66% efficiency; the iterative VCG averaged 78%, and the package bidding mechanism averaged 81% efficiency.

Another Caltech experiment tested a combinatorial design for the FCC spectrum auctions [Bykowsky et al., 2000]. The FCC did not use combinatorial markets for its spectrum auctions despite the well-known complementarities, due to concerns with computational costs and bidding strategy issues.[15]

Combinatorial mechanisms directly address the problem we have identified many times in this chapter: the performance of negotiation mechanisms will depend crucially on the quality of the match between the mechanism's domain of goods and the domain

[14]Two goods are *gross substitutes* when the Marshallian demand for one increases as the price of the other increases. The Marshallian demand is the "ordinary" demand; that is, it reflects how a consumer's demand changes with price changes, without any income compensation to hold the consumer's level of overall utility constant. See, e.g., Mas-Colell et al. [1995].

[15]We discuss the FCC auctions further in Section 4.3.2 below, when we address agent bidding strategies for simultaneous ascending auctions.

of agent preferences. To date few combinatorial mechanisms have been implemented, but the very active research on each of the design problems we identify offers hope that this approach to computational negotiation will become more usable in the future.

3.3 Exchanging: Transaction services

Once a deal is negotiated, it remains for the parties to execute the agreed-upon exchange. Many online marketplaces support transaction services to some extent, recognizing that integrating "back-end" functions—such as logistics, fulfillment, and settlement—can reduce overall transaction costs and enhance the overall value of a marketplace [Woods, 2002].

A critical component of market-based exchange, of course, is *payment*, the actual transfer of money as part of an overall transaction. The online medium enables the automation of payment in new ways, and indeed, the 1990s saw the introduction of many novel *electronic payment mechanisms* [O'Mahony et al., 1997], offering a variety of interesting features [MacKie-Mason and White, 1997], including many not available in conventional financial clearing systems. For example, some of the schemes supported anonymity [Chaum, 1992], micropayments [Manasse, 1995], or atomic exchange of digital goods with payment [Sirbu and Tygar, 1995].

As it turned out, none of the innovative electronic payment mechanisms really caught on. There are several plausible explanations [Crocker, 1999], including inconvenience of special-purpose software, network effects (i.e., the need to achieve a critical mass of buyers and sellers), the rise of advertising-supported Internet content, and decreases in credit-card processing fees. Nevertheless, some new payment services proved complementary with marketplace functions, and thrived. The most well-known example is PayPal, which became extremely popular among buyers and sellers in person-to-person auctions, who benefited greatly from simple third-party payment services. PayPal's rapid ascension was in large part due to an effective "viral marketing" launch strategy, in which one could send money to any individual, who would then be enticed to open an account [Jackson, 2004]. PayPal is still not economical for micropayments, however, and new schemes—most notably, Peppercoin [Micali and Rivest, 2002]—have emerged aiming to provide such services.

4 Automating market participants

Part of automating markets is automating the behavior of participants in those markets. Of course, computerized trading has been a reality almost as long as we have had computers. What is relatively new is the proliferation of electronic markets on networks, and their potential to dramatically expand the opportunities for automating trading functions in a broad variety of domains. Conversely, automating traders can shape the automation of markets, for example, by rendering feasible some market designs too complex for manual traders.

As in most realms of computerization, there is no sharp line between automated and non-automated trading. Virtually all trading in financial markets is mediated by computers at some stage of the process, and the same is true by definition for markets that themselves operate electronically. Consider the communication flow of the generic trading system diagrammed in Figure 3. The human trader issues “instructions” to the computer, resulting in bids submitted to the market. The instructions may be direct, for example, “buy 100 shares XYZ at $20”, in which case the computer is merely serving a communication interface function, and the trading is essentially manual. To the extent instructions are indirect, such as “balance my portfolio”, or “liquidate my holdings in sector S in an orderly manner”, we would characterize the trading as automated to a correspondingly greater degree. Similarly, to the extent the computer processes the market information for presentation to the human—summarizing, identifying patterns, even recommending specific trading actions—we would have to credit the machine with a share of the overall decision process.

<Figure 3 about here>

The upshot is that automated trading exists on a continuum, and it is futile to attempt a precise binary classification of human and computer trading activity. We merely observe that the computer's role is often significant, and growing over time in sophistication and complexity. Of course, the trading is ultimately on behalf of humans (or organizations operated by and for humans), and so the humans will continue to exert ultimate control and influence over their *trading agents*. As the machines prove increasingly worthy of trust in their competence to execute decreasingly direct instructions, it is inevitable that a significant fraction of trading activity will become fully automated for all practical purposes.

Recognizing that overall trading activity is the product of manual and automated components, we henceforth apply the term “trading agent” to the combined entity interacting with the market, enclosed by the dashed box in Figure 3.

The phenomenon of automated trading raises several interesting questions.

- How should trading agents behave? How can we design effective strategies for a range of environments? How can we construct agents capable of incorporating new information and objectives, and adapting to changing circumstances?

- How will automated traders change the character and behavior of markets? Will additional stabilization, security, or other safety-related mechanisms be required?

- How can we design markets to cater to or exploit the capabilities of automated trading agents?

In the sections below we address these questions, in the course of surveying some significant threads in trading agent research. Our focus is on works that study the behavior or potential of trading agents themselves, as opposed to efforts that use computational agents as a way to model human trading behavior. The latter is the domain of much research in agent-based computational economics, addressed in several other chapters of this handbook.

4.1 Program trading

As noted above, automation of trading in financial markets is a well-established practice. The term “program trading” (or “programmed trading”) is sometimes applied generically to any initiation of trade activity based on procedural rules (typically implemented by computer programs), but more frequently refers to a particular form of trading based on *index arbitrage* [Brennan and Schwartz, 1990] or other standard portfolio trading strategies. Index-arbitrage programs monitor the price of index futures contracts (e.g., for the Standard & Poor's 500), as well as the basket of underlying securities, and triggers trades whenever the futures price deviates from the underlying price by some pre-specified threshold dependent on the interest rate. Academic interest in program trading focused on the effect of this activity on price volatility, including much investigation of its relation to the October 1987 stock market crash [Baldauf and Santoni, 1991].

The New York Stock Exchange (NYSE) requires its members to report trades involving fifteen or more stocks with aggregate value of a million dollars or more. This definition is designed to capture the common pattern of program trading for index or other derivative-based arbitrage, portfolio insurance, and other portfolio-based actions. According to NYSE, such trades account for a large fraction of overall volume: 51.2%, for a typical example, in the last week of January 2005. Of this, 11.4% was attributed to index arbitrage specifically.

It is of course possible to implement any systematic trading strategy in a computer program, and many such programs have been marketed to investors as "black-box" or "gray box" trading systems (so-called because their specific trading rules are secret or only partially revealed). As the availability of financial market data via the Internet has increased, so have the offerings of software packages providing analysis and monitoring tools, some providing interfaces for user-specified trading rules. Whereas it is doubtful that retail investors can profit substantially through such means, major brokerages reportedly devoted significant resources to computational modeling and automated trading strategies for internal use by their trading units. For proprietary reasons, little is publicly known about the nature and extent of these computerized trading activities. Bass [1999] presents an unusually forthcoming story of the Prediction Company's efforts in automated trading, but even this account stops short of technical and strategic precision.

4.2 Market interfaces

Automated agents interact with electronic markets according to standardized interfaces. Program trading in financial markets is facilitated by ECNs (electronic crossing networks) such as Island and Instinet, which support specified network protocols for submitting stock orders. The Small Order Execution System provides an analogous standard interface to Nasdaq market makers.

Online marketplaces inherently provide a window to automated traders, as a side effect of supporting standard web protocols. For example, eBay cannot necessarily distinguish a bid submitted by a human user through a browser from one generated by a program constructing the same web posting. Users have taken advantage of this opportunity, for example by employing programs to submit bids at prespecified times, typically seconds before the scheduled auction close. This practice, called *sniping*, is quite common on eBay, and can be supported by several auction-theoretic arguments [Roth and Ockenfels, 2002]. Services such as eSnipe also provide rudimentary facilities to condition bids on auction events, such as the success or failure of related bids in specified auction groups.

Definition of a market interface is part of the overall task of market design. Bidding rules comprise a dimension of market design space, governing the language of allowable bids as well as the policy for admitting bids over time [Wurman et al., 2001]. Choice of a bidding language often entails addressing rich tradeoffs, for example in the complexity of bidding or evaluating bids [Nisan, 2000]. In some cases, a designer might intentionally restrict bidding rules in order to simplify the interface implementation or to bias toward simple negotiation strategies [Cranor and Resnick, 2000].

To fully support automated trading, market interfaces would provide machine-readable specification of bidding rules, as well as other market policies. This would facilitate deployment and testing of mechanisms, promote transparency, and ultimately support automatic adaptation of trading strategies. Although sufficiently flexible and formal standards for specifying markets are not yet available, special-purpose languages for specifying auctions [Lochner and Wellman, 2004] and reasoning about negotiation protocols [Guerin and Pitt, 2002] constitute steps in this direction.

4.3 Agent strategies

How a trading agent should behave depends, of course, on the market mechanism and other agents in its environment. Studies of trading agent strategy typically focus on a particular environment; there have been few attempts thus far to distill general cross-cutting principles. In this section we examine research on strategies for two canonical market environments: individual continuous double auctions and collections of simultaneous auctions. In Section 4.4 we consider a more complex market game combining several different market mechanisms.

4.3.1 Continuous double auction strategies

One of the most basic trading scenarios is an abstract market based on the continuous double auction (CDA) mechanism [Friedman, 1993]. The CDA is a simple and well-studied auction institution, employed commonly in commodity and financial markets. The “double” in its name refers to the fact that both buyers and sellers submit bids, and it is “continuous” in the sense that the market clears instantaneously on receipt of compatible bids.[16]

The CDA has also been widely employed in experimental economic studies, and notably in an open research competition conducted at a Santa Fe Institute workshop in 1990 [Friedman and Rust, 1993; Rust et al., 1994]. The winning trader in this competition held back until most of the other agents revealed their valuations through bidding behavior, then “stole the deal” by sniping at an advantageous price. Agents employing more elaborate reasoning failed to make such sophistication pay off. This is consistent with observations that even extremely naive strategies—exhibiting what Gode and Sunder [1993] dubbed “zero intelligence” (ZI)—achieve virtually efficient outcomes

[16] In the computer science literature “continuous” mechanisms are usually called “on-line”; we discussed some theoretical results for on-line scheduling market design in section 3.2.1, above.

in this environment. Such results suggested a strong limit on the potential returns to positive smarts.

Over the last fifteen years, CDA markets served as a basis for many further studies of artificial trading agents. The simplicity and familiarity of the abstract CDA framework presents some distinct advantages as the basis for trading agent research. These include ease of explanation and simulation, low barriers to entry, consensus understanding of market rules, predictability of behavior, opportunity to build on prior work (on design of both mechanism and agents), and analyzability of outcomes. Given the ubiquity of the CDA institution, there is even a potential to incorporate real-world market data of various kinds.

Cliff [1998] provides an extensive bibliography covering much of this work, including his own evolutionary studies of "ZI plus" agents. One particularly influential trading strategy was proposed by Gjerstad and Dickhaut [1998], later revised and termed the "heuristic belief learning" (HBL) model [Gjerstad, 2004]. An HBL agent maintains a belief state over acceptance of hypothetical buy or sell offers, constructed from historical observed frequencies. It then constructs optimal offers with respect to these beliefs and its underlying valuations. The timing of bid generation is stochastic, controlled by a *pace* parameter, which may depend on absolute time and the agent's current position. Gjerstad [2004] demonstrates that pace is a pivotal strategic variable, and that indeed there is surprisingly large potential advantage to strategic dynamic behavior despite the eventual convergence to competitive prices and allocations.

In extensive simulated trials, Tesauro and Das [2001] found that a modified version of HBL outperformed a range of other strategies, including ZI, ZI plus, and the sniping strategy that won the original Santa Fe tournament. The strategy also compared favorably with human traders [Das et al., 2001].

Because CDAs or close variants are widely employed in financial markets, models from the finance literature that account for details of the trading mechanism, or *market microstructure* [Garman, 1976], are also highly relevant to trading agent strategy.[17] Much of this literature addresses the trading problem from a market maker's perspective, explaining price spreads and the potential for dealer profit by way of transaction costs and inventory management, information asymmetries, or strategic opportunities [O'Hara, 1995].

Availability of real-time market information has recently begun to enable higher-fidelity modeling of financial trading environments. The Penn Exchange Simulator [Kearns and Ortiz, 2003] merges bids from automated trading agents with actual limit-order streams, providing realistic volume and volatility patterns, whether or not these would emerge naturally from the artificial agent strategies. Competitions based on this simulator enabled comparison of a wide variety of CDA bidding policies [Sherstov and Stone, 2004], including some that may use information from the entire order book [Kearns and Ortiz, 2003].

[17] Agent-based finance models, as discussed by Hommes [2005] and LeBaron [2005], are primarily directed at explaining aggregate behavior, but may also prove useful for strategic studies.

4.3.2 Simultaneous ascending auction strategies

A *simultaneous ascending auction* (SAA) allocates a set of M related goods among N agents via separate English auctions for each good. Each auction may undergo multiple rounds of bidding. At any given time, the *bid price* on good m is β_m, defined to be the highest agent bid $\max_{1 \leq j \leq N}\{b_j^m\}$ received thus far, or zero if there have been no bids. To be admissible, a new bid must meet the bid price plus a bid increment (which we take to be one w.l.o.g.), $b_j^m \geq \beta_m + 1$. If an auction receives multiple admissible bids in a given round, it admits the highest (breaking ties arbitrarily). An auction is *quiescent* when a round passes with no new admissible bids.

The auctions proceed concurrently. When all are simultaneously quiescent, the auctions close and allocate their respective goods per the last admitted bids. Because no good is committed until all are, an agent's bidding strategy in one auction cannot be contingent on the outcome for another. Thus, an agent j desiring a bundle of goods inherently runs the risk—if it bids at all—that it will purchase some but not all goods in the bundle. This is the well-known *exposure problem*, and arises whenever agents have complementarities among goods allocated through separate markets. The exposure problem is perhaps the pivotal strategic issue in SAAs.

As noted above, dealing with complementarities was a prime motivation for the development and exploration of combinatorial auctions in recent years. Although such mechanisms may provide an effective solution in many cases, there are often significant barriers to their application. Most significantly, conducting a combinatorial auction requires the existence of a competent authority to coordinate the allocation of interdependent resources, and incurs costs and delays associated with such coordination. It is a simple fact that today we see many markets operating separately, despite apparent strong complementarities for their respective goods. Whereas automation will very likely increase the prevalence of combinatorial markets, we expect that the issue of trading in separate dependent markets will remain for the foreseeable future.

Perhaps the most natural baseline for SAAs is a strategy called *straightforward bidding* (SB).[18] A straightforward bidder takes a vector of *perceived prices* for the goods as given, and bids those prices for the bundle of goods that would maximize the agent's surplus if it were to win all of its bids at those prices.

Let $v_j(X)$ denote the value to agent j of obtaining the set of goods X. Given that it obtains X at prices $\vec{p}$, the agent's *surplus* is its value less the amount paid, $\sigma(X, \vec{p}) = v_j(X) - \sum_{m \in X} p_m$. When agent j is winning the set of goods X_{-1} in the previous bidding round, we define the current perceived prices to be $\hat{p}_m = \beta_m$ for $m \in X_{-1}$, and $\hat{p}_m = \beta_m + 1$ otherwise. Then, under SB, agent j bids $b_j^m = \hat{p}_m$ for $m \in X^*$ such that $X^* = \arg\max_X \sigma(X, \vec{\hat{p}})$.

The straightforward bidding strategy is quite simple, involving no anticipation of other agents' strategies. For the single-unit problem, such anticipation is unnecessary, as the agent would not wish to change its bid even after observing what the other agents did [Bikhchandani and Mamer, 1997]. This is called the *no regret* property [Hart and Mas-Colell, 2000], and means that from the agent's perspective, no bidding policy would have been a better response to the other agents' bids.

[18] We adopt the terminology introduced by Milgrom [2000]. The same strategy concept is also referred to as "myopic best response", or "myopically optimal", or even "myoptimal" [Kephart et al., 1998].

For a *single-unit value function*, the value of a set of goods is just that of its most valuable included singleton. When all agents have single-unit value, and value every good equally, the situation is equivalent to a problem in which all buyers have an inelastic demand for a single unit of a homogeneous commodity. For this problem, Peters and Severinov [2001] showed that straightforward bidding is a perfect Bayesian equilibrium. Up to a discretization error, the allocations from SAAs are efficient when agents follow straightforward bidding. It can also be shown [Bertsekas, 1992; Wellman et al., 2001a] that the final price vector will differ from the minimum unique equilibrium price by at most $\kappa \equiv \min(M, N)$. The value of the allocation, defined to be the sum of the bidder surpluses, will differ from the optimal by at most $\kappa(1 + \kappa)$.

Unfortunately, the very nice properties for straightforward bidding with single-unit value do not carry over to multiple-unit problems. Indeed, the resulting price vector can differ from the minimum equilibrium price vector, and the allocation value can differ from the optimal, by arbitrarily large amounts [Wellman et al., 2001a]. However, whereas the case against SB is quite clear, auction theory [Krishna, 2002] to date has relatively little to say about how one *should* bid in simultaneous markets with complementarities. In fact, determining an optimal strategy even when it is known that other agents are playing SB turns out to be an unsolved and surprisingly difficult problem, sensitive to the smallest details of preference distributions [Reeves et al., 2005].

Our gap in knowledge about SAA strategy is especially striking given the ubiquity of simultaneous auctions in economically significant settings. Indeed, markets for interdependent goods operating simultaneously and independently represents the normal or default state of affairs. Even for some markets that are expressly designed, most famously the US FCC spectrum auctions starting in the mid-1990s [McAfee and McMillan, 1996], a variant of the SAA is deliberately adopted, despite awareness of strategic complications [Milgrom, 2000]. Simulation studies of scenarios based on the FCC auctions shed light on some strategic issues [Csirik et al., 2001], as have accounts of some of the strategists involved [Cramton, 1995; Weber, 1997], but the general game is still too complex to admit definitive strategic recommendations.

In our own work, we explored SAA strategies in the context of a simple market-based scheduling scenario [MacKie-Mason et al., 2004; Reeves et al., 2005]. In the scheduling game, agents need to complete a job requiring a specified duration of resource, by acquiring the resource over individual time slots. The value for completing a job depends on when it is finished. Complementarities arise whenever jobs require more than a single time slot.

We investigated a family of possible strategies for this game, employing an empirical methodology discussed in some detail in section 5 below. Our basic approach was to start with SB as a baseline, and evaluate parametric variations through extensive simulation and analysis. In particular, we considered two extensions of SB designed to mitigate the exposure problem. First, we modify SB to account for sunk costs to some degree, recognizing that goods an agent is already winning will pose no marginal costs if other agents do not submit additional bids. The strategy is implemented in terms of a "sunk awareness" parameter ranging over [0,1], with zero treating all winning bids as sunk costs and one corresponding to unmodified SB. Perhaps it should not be surprising that the equilibrium settings of this parameter are quite sensitive to the distribution of agent job characteristics (length, deadline values). We identified qualitatively distinct equilibria corresponding to different job distributions.

The second alternative we considered attempts to explicitly predict the closing prices for each slot, and selects bundles based on these price predictions [MacKie-Mason et al., 2004]. Our overall finding is that this approach is quite effective compared to SB or employing a global sunk-awareness parameter. Performance, of course, depends on the prediction vector employed by the agent, as well as the distribution of job characteristics. Since prices are observable, however, it is perhaps plausible to glean the prediction vectors directly from experience (real or simulated). The structure of the prediction methods surviving in equilibrium appear relatively robust to changing the agent job distributions.

4.4 Case study: Trading agent competition

Inspired by success of Santa Fe double auction tournament and other research competitions, a community of trading-agent researchers established an annual competition event designed to focus effort on a common problem, thus enabling researchers to compare techniques and build on each others' ideas [Wellman et al., 2001b]. Working on a shared problem coordinates attention on particular issues (among the many of interest in the trading domain), and facilitates communication of methods and results by fixing a set of assumptions and other environment settings.

The multi-year Trading Agent Competition (TAC) series offers the further prospect of learning from shared experience over time. As a case study of trading agent research, we examine the experience of the first four years of TAC, and some of the research results spawned from that activity. The first TAC was held in 2000, followed by annual sequels, each attracting approximately twenty participant teams. In 2003, TAC introduced a second game, in the domain of supply chain management [Arunachalam and Sadeh, 2005], which also produced significant interest and research activity. In this case study we focus on the original travel-shopping market game.

4.4.1 TAC travel-shopping rules

The TAC travel-shopping market game presents a travel-shopping task, where traders assemble flights, hotels, and entertainment into trips for a set of eight probabilistically generated clients. Clients are described by their preferred arrival and departure days, the premium they are willing to pay to stay at the nicer hotel, and their respective values for three different types of entertainment events. The agents' objective is to maximize the value of trips for their clients, net of expenditures in the markets for travel goods. The three categories of goods are exchanged through distinct market mechanisms.

Flights. A feasible trip includes round-trip air, which consists of an inflight day i and outflight day j, $1 \leq i < j \leq 5$. Flights in and out each day are sold independently, at prices determined by a stochastic process. The initial price for each flight is distributed uniformly, following a random walk thereafter with an increasingly upward bias.

Hotels. Feasible trips must also include a room in one of the two hotels for each night of the client's stay. There are 16 rooms available in each hotel each night, and these are sold through ascending 16th-price auctions. Agents submit bids for various quantities, specifying the price offered for each additional unit. Each minute, the hotel auctions issue *quotes*, indicating the 16th- (*ASK*) and 17th-highest (*BID*) prices among the currently active unit offers. To ensure ascending prices, hotel bidders are subject to a "beat-the-quote" rule [Wurman et al., 2001], requiring that any new bid offer to purchase at least one unit at a price of $ASK + 1$, and at least as many units at $ASK + 1$ as the agent was previously winning at *ASK*. Also each minute, starting at minute four, one of the hotel auctions is selected at random to close, with the others remaining active and open for bids. When the auction closes, the units are allocated to the 16 highest offers, with all bidders paying the price of the lowest winning offer.

Entertainment. Agents receive an initial random allocation of entertainment tickets (indexed by type and day), which they may allocate to their own clients or sell to other agents through CDAs. The entertainment auctions issue *BID* and *ASK* quotes representing the highest outstanding buy and lowest sell offer, respectively, and remain open for buying and selling throughout the 12-minute game duration.

A feasible client trip is defined by inflight and ouflight days, rooms in the same hotel for all nights in the interim, and a set of entertainment tickets. The client's utility for this trip is given by a constant base value, minus penalties for deviating from preferred dates, plus (if applicable) bonuses for staying in the premium hotel and attending entertainment. At the end of a game instance, the TAC server calculates the optimal allocation of trips to clients for each agent, given final holdings of flights, hotels, and entertainment. The agent's game score is its total client trip utility, minus net expenditures in the TAC auctions.

4.4.2 TAC experience

As we can see, the TAC travel-shopping game scenario presents a challenging trading problem, involving multiple interdependent goods allocated over time, through three distinct market mechanisms. Flights are sold through take-it-or-leave-it offers, hotels through multiunit SAAs (with stochastic termination), and entertainment through CDAs. Each of these poses open strategic problems.

The TAC record is well documented, including accounts of particular tournaments [Wellman et al., 2001b, 2003b; Lanzi and Strada, 2002; Eriksson and Janson, 2002], and summary descriptions of competing agents [Greenwald and Stone, 2001; Greenwald, 2003]. We also investigated behavior across years [Wellman et al., 2003a], finding that over time the allocation of travel resources in TAC play has become increasingly efficient. Since the TAC market appears to be quite competitive (as discussed below), this provides indirect evidence of general progress in agent performance.

One of the first findings to emerge from TAC was simply that a diverse set of research groups (ranging from individual students or employees to teams of senior researchers) were capable of constructing competent agents to play a complex game. By and large, most participants recognized the key strategic issues, and solved relevant subproblems accurately. For example, two key subproblems identified and solved by many perticipants were determining the optimal allocation of a given set of goods to clients, and evaluating the marginal utility of a particular good [Greenwald and Boyan, 2001; Stone et al., 2001]. Techniques for such core problems are generally disclosed by participants after the competition, and often incorporated and extended by other entrants in the next year's event.

In some cases, work on challenging TAC problems spurred research on techniques applicable much more generally in automated reasoning and decision making. For example, Stone et al. [2003] extended boosting techniques from machine learning to estimate conditional densities, driven by the pivotal TAC problem of estimating future hotel prices given current and historical price information, as well as other features.

Sophisticated learning of price distributions was undoubtedly a major ingredient in the success of ATTac-2001, which finished in a virtual two-way tie for first place in the 2001 TAC tournament. Its precise monitoring and reaction to prices was in stark contrast with the other first-place agent, livingagents [Fritschi and Dorer, 2002], which implemented a comparatively simple strategy of predicting optimal trips at the beginning and then taking hotel prices however they turned out. That such open-loop behavior could work so well was initially surprising. Indeed, if all agents played the livingagents strategy, hotel prices would skyrocket to unprofitable levels. But in the actual tournament, stabilizing agents like ATTac-2001 were the norm, effectively removing the risk to blind price-taking behavior.

An interesting lesson from this 2001 outcome was that interactions among the strategies are indeed important in TAC. The success of price-taking in the finalist pool also suggests that the market was fairly competitive. In the 2002 tournament, Walverine [Cheng et al., 2005] took the competitiveness assumption seriously, modeling the TAC hotel market as a perfectly competitive system. Specifically, Walverine derived the Walrasian equilibrium for hotel prices given the initial flight prices and expected demand based on the known distribution of client preferences. This proved to be quite accurate as an initial prediction for hotel prices, performing on par with the sophisticated machine learning method employed by ATTac-2001 [Stone et al., 2003], and significantly better than all other approaches in the TAC-02 finals [Wellman et al., 2004]. This is perhaps surprising, given that Walverine was the only agent that did not employ historical data in its prediction method. Subsequent analysis indicated that a key determinant of success was taking into account the effect of flight prices on clients' choices of travel dates (and therefore hotel demands on different days). This relationship was pivotal in Walverine's competitive equilibrium analysis, and was empirically learned by ATTac-2001 as well as kavayaH [Putchala et al., 2002], which predicted prices based on a neural-network model.

Predictions are of course uncertain, and TAC participants have identified several approaches to using probability distribution information in their bidding strategies. ATTac-2001 made decisions based on sampling from the price distribution, but its developers found in subsequent experiments that deciding directly based on distribution means was more effective [Stone et al., 2003]. Similar results in the context of other agents were reported by the developers of RoxyBot and Walverine. Greenwald and Boyan [2004] performed a careful study of the general problem of bidding under uncertainty, comparing the problem as it arises in TAC to simpler models of purely sequential and simultaneous auctions. Hotel auctions in TAC are a hybrid, as agents bid simultaneously in all of them, after which one closes, and the agents have an opportunity to revise bids in the rest based on the results. Their study found that TAC hotel auctions strategically resemble simultaneous more than sequential auctions, which suggests that insights from research on SAAs (Section 4.3.2) may prove applicable to this problem.

Overall, success in TAC requires putting together solutions to the several subproblems comprising the game. The top scorer in the 2002 tournament was whitebear, whose developers tuned to victory through a process of extensive simulation experiments, performed systematically over a set of key control parameters [Vetsikas and Selman, 2003]. The 2003 tournament proved to be the tightest competition yet, with less than 100 points separating the top five agents: ATTac-2001, PackaTAC, whitebear, Thalis, and UMBCTAC.

5 A computational reasoning methodology for analyzing mechanisms and strategies

To conduct descriptive and explanatory research, economists traditionally rely heavily on the specification of stylized models that abstract from many real-world details in order to obtain formal results. One of our themes is that less formalism is reasonable when economics is practiced as a normative science applied to the design of computational markets and agents. Implementation details, problem complexity, and context matter in a fundamental way.

Direct application of analytic (usually game) theory quickly becomes infeasible as problem complexity grows, as reflected (informally) in size of strategy space, number of agents, degree of incomplete and imperfect information, and dynamism. Despite recent advances in game computation [Koller et al., 1996; McKelvey and McLennan, 1996; Kearns et al., 2001; Porter et al., 2004], even moderate size coupled with uncertainty and dynamics suffices to place modest but interesting market designs beyond the range of currently available solution methods.[19] As one well-known example, consider the FCC spectrum auctions. These multi-billion dollar auctions were designed by some of the best auction theory researchers alive, and major bidders hired most of the rest of the top auction researchers to help them devise strategies. Yet neither the market or agent strategy designers were able to analytically solve the game induced by the auction rules.[20] The outlook is more bleak for the numerous other markets that are at least as complex but less rich with potential gains from analytical solution.

When analytic methods are infeasible, what other tools are available for market and agent designers? One standard method is to statistically study quasi-experimental evidence from real-world market implementations to test generalizable hypotheses. Of course, for computational market design there are few implementations in the field, especially if we wish to test new ideas. In this case, a variant is to design markets based on heuristics when theory is not complete, implement them in the field, and test their performance. This process, unfortunately is both slow and extremely expensive.

A related approach that has been used from the earliest days of computational market design is to test implementations in human subject laboratory experiments. Some

[19] Although the theoretical complexity of various game-computation problems [Conitzer and Sandholm, 2003; Papadimitriou, 2001] is to some extent unsettled, the practical unsolvability of many games of interest—now and for the foreseeable future—is an uncontroversial proposition.

[20] Nor, apparently, did the designers anticipate and prevent certain collusive strategies; see, e.g., Weber [1997].

of the earliest computational market designers were also among the pioneers of experimental lab methods in economics; in particular, the economists at the University of Arizona [Smith, 1962] and the California Institute of Technology [Plott, 1986]. This coincidence is not terribly surprising: to test any market, including non-computational, in a lab setting, researchers quickly found it expedient to build computational markets so that the experiment interface and instrumentation could be automated. However, although laboratory experiments are often more practical than field trials, they are still expensive. Further, mechanism and strategy complexity is limited by reliance on non-expert human participants.

We describe an emerging methodology that uses computational experiments to systematically investigate agent strategies and the performance of market mechanisms. The method begins with an explicit formulation of the resource allocation problem, and proceeds through at least five distinct tasks (we elaborate on these and provide references in the ensuing subsections):

1. **Specify a computational mechanism** (or several). Designs can be generated from innovation to existing forms, creative speculation, or through directed search (say, with a genetic program [Cliff, 2003; Phelps et al., 2002]).

2. **Generate candidate strategies.** As with mechanisms, candidate strategies can be generated in several ways. One promising idea is to search systematically or randomly through some encoding of strategy space. Another is to specify a strategy family parameterized to address important tradeoffs, perhaps based on a previously studied strategy. In any case, it is necessary to reduce dimensionality by restricting the strategy space, in order to employ numeric analysis methods.

3. **Estimate the "empirical game".** Simulation and sampling converts the extensive form game of incomplete information into a normal form with expected payoffs associated with each possible strategy profile.

4. **Solve the empirical game.** Methods such as replicator dynamics exploit symmetry or other available structure to efficiently solve large games for their equilibria.

5. **Analyze the results.** Attempt to extract generalizable regularities, and employ sensitivity analysis to drive further sampling and search.

These methods are emerging in the work of several authors [Reeves et al., 2005; MacKie-Mason et al., 2004; Armantier et al., 2003; Kephart and Greenwald, 2002; Walsh et al., 2002]. They are related in some respects to the generative social science methods used elsewhere in agent-based research [Epstein, 2005].

In the remainder of this section we discuss most of the main steps in this methodology. We do not devote any further attention to the first step of specifying a mechanism: this was the subject of Section 3 (especially Section 3.2).

5.1 Generate candidate strategies

One important source of intractability in market mechanisms is the enormousness of the strategy space. For example, in the market-based scheduling problem we studied, agent strategies include all functions from preferences (job length and deadline values) and price-quote histories to current-round bid vectors. The strategy domain includes all preferences ($M + 1$ - dimensional, when there are M time slots), plus all price-quote histories up to the current time T (MT-dimensional). Partial or full combinatorial mechanisms have even higher dimensionality. Exploring all possible mappings from an $M(T + 1)$ to an M dimensional space is clearly not feasible. The traditional approach is to impose a rationality assumption (usually Bayes-Nash) and solve analytically for optimal strategies, but as we noted above the problem is not tractable for most complex mechanisms.

To render computational analysis feasible, the researcher restricts the strategy space to a manageable set. Typically, the researcher will specify a few "interesting" strategies, generated by intuition or experience, and analyze their performance against each other. Selten et al. [1997] implement a strategy generation method first proposed by Selten in 1967: humans play strategies in a laboratory setting to gain experience with the game and the mechanism, and then program those strategies so the researchers could analyze them further. In a similar vein, Axelrod [1984] solicited programmed strategies. Another approach is to implement a directed search strategy, such as a genetic algorithm, to select candidates from the full strategy space [Miller, 1988; Koza, 1991; Ünver, 2001].

A different approach for identifying such strategies that we explored is to specify a reasonable skeletal structure augmented with control parameters addressing key tradeoffs, and then to vary the parameters. For example, as discussed in Section 4.3.2, straightforward bidding (SB) is a natural candidate for a baseline strategy in any simultaneous ascending auction situation [Milgrom, 2000]. An SB agent determines which subset of goods (including the null set) would be most profitable at currently available prices, and places incremental bids on those it is not currently winning. For our scheduling problem we considered variants of SB that admit deviations from its myopic behavior. One variant was to introduce a "sunk awareness" parameter to account for exposure risk when an agent is already high bidder on some but not all slots it needs to complete its package [Reeves et al., 2005]. Parameters need not be limited to scalar quantities. We recently investigated bidding strategies that use explicit price prediction [MacKie-Mason et al., 2004], similar to many of the trading agents in the TAC competition (see Section 4.4). The parameters in this case may be vectors of expected prices, full belief distributions, or more generally, methods for price prediction that may be plugged in to the broader bidding strategy.

Although sometimes for a different purpose, many investigations of bidding agents include simulations of what are essentially restricted strategy profiles [Csirik et al., 2001; Goldman et al., 2001; Wellman et al., 2005, 2003a; Stone et al., 2001, 2003; Vetsikas and Selman, 2003].

5.2 Estimate the "empirical game"

Given a restricted set of candidate strategies, the cross-product of these sets across agents induces a space of *strategy profiles*, defining a restricted game. The payoff to each agent in a given profile is defined as the expected payoff for playing its corresponding strategy, where expectations are taken with respect to the distribution over the agents' private information, and any other stochastic factors.

For shorthand, refer to the joint probability function over these random variables as the *type distribution*. Given a specification of the type distribution, the expected payoffs with respect to this distribution can be estimated via sampling. The researcher draws randomly from the type distribution, and simulates play for a given profile. In the limit, the sample average payoff vector will approach the true expected payoffs for this profile if the mild conditions hold to support a weak law of large numbers. We refer to the mapping of strategy profiles to their estimated payoff vectors as an *empirical game*. This mapping has also been termed a *heuristic strategy payoff matrix* [Walsh et al., 2003].

For example, we investigated a task-allocation problem in an information-collection domain [Cheng et al., 2003]. The game has five agents, and we restricted the agents to choose among three available strategies (A, B, C). The game is symmetric, which means that each agent receives the same payoff from a given strategy when it faces a given profile of strategies played by the other agents (in payoff matrix terms, the matrix is symmetric). Agent types represent resources and tasks assigned in a particular game instance. Figure 4 depicts the empirical game matrix. We constructed similar empirical games for many other scenarios, including several configurations of the scheduling problem, with varying numbers of agents and strategies.

<Figure 4 about here>

5.3 Solve the empirical game

With a normal form expression of the empirical game, the next step is to solve for one or more of the Nash equilibria. Because it is based on a restricted strategy set, a Nash equilibrium of the empirical game—termed a *constrained strategic equilibrium* (CSE) [Armantier et al., 2003]—does not correspond to an equilibrium of the full original game (even ignoring sampling error). Moreover, because the strategies already dictate how agents choose their actions based on private information, the CSE is not even a Bayes-Nash equilibrium (BNE) of the game where agents may play any of the strategies conditional on this private information. For this reason, Walsh et al. [2003] refer to the derived solution profile as an *ex ante* Nash equilibrium. In the limit as we relax strategy restrictions, a CSE converges to a BNE [Armantier et al., 2003].

There are a variety of tools for finding a CSE in the restricted empirical game. The state-of-the-art solver for finite games is GAMBIT [McKelvey et al., 1992]. But GAMBIT fails to exploit key structure in many games, such as symmetry. Converting the compact, symmetric representation of a payoff matrix into the more general form often renders the problem of finding equilibria intractable. For example, we have had GAMBIT fail on games with five agents choosing among five strategies. For this reason, we used two other solution methods that do exploit symmetry, described below.

In his original exposition of the concept, Nash [1950] suggested an evolutionary interpretation of the Nash equilibrium. We used the related replicator dynamics formalism [Taylor and Jonker, 1978; Schuster and Sigmund, 1983] in service of computing equilibria. Friedman [1991] proves that if the probabilities in a mixed strategy are cast as proportions of a large population of agents playing the corresponding pure strategies, then an agent population that reaches a fixed point with respect to the replicator dynamics will be a symmetric mixed-strategy Nash equilibrium. This definition suggests an evolutionary algorithm in which population proportions are iteratively updated in successive generations.

We illustrate this evolutionary process in Figure 5 for a version of our scheduling game [MacKie-Mason et al., 2004]. In this particular example, agents in a five-player game are drawn from a population in which the indicated fractions play one of five strategies from the set labeled by $\{16, 17, 18, 19, 20\}$. These strategy labels refer to a parameter we call "sunk awareness"; when zero the strategy treats all winning bids as sunk costs, and when the parameter is one the strategy completely ignores sunk costs. In the figure, the population is converging to the mixed strategy of playing strategy 16 with probability 0.745, and 17 with probability 0.255. In our experience the replicator dynamics method converges quickly, however the theory only guarantees convergence to a Nash equilibrium is the number of generations approaches infinity. We are unaware of any literature that systematically analyzed the performance of this method for solving matrix form games.

<Figure 5 about here>

Another solution method for symmetric games characterizes (symmetric) Nash equilibrium as the global minimum of a function mapping mixed strategies to the reals. For our experiments, we used an adaptation of a Nelder and Mead [1965] nonlinear function minimizer developed by Walsh et al. [2002].

Although, in part by exploiting symmetry and using replicator dynamics, we have been able to solve moderately large games faster and more successfully than GAMBIT, the problem is still computationally burdensome. The number of strategy profiles in a game with N agents and S strategies is the binomial coefficient $\binom{N+S-1}{N}$. For example, we recently studied a problem with five agents and 53 possible strategies, which has over four million unique strategy profiles to evaluate and hand to a game solving tool to find the equilibrium strategy set [Osepayshvili et al., 2005]. We have not come close to estimating empirically all of the cells in the payoff matrix.

5.4 Analyze the results

Once an equilibrium strategy set is obtained, all of the usual analyses can be performed (subject to the caveat that the equilibria hold with respect to a restricted set of permissible strategies). For example, the equilibrium strategies can be analyzed to discover critical features that explain their strategic robustness, or to measure their performance under various conditions. Or, the equilibrium set can be calculated for each of several candidate mechanisms, and then the performance of the mechanisms compared under equilibrium play as part of the design loop to obtain better mechanisms.

As we noted above, fully solving a large game may be infeasible. However, partial empirical data offers opportunities for analysis as well. For example, in our game with five agents and 53 possible strategies, we have empirically evaluated only 4916 of the more than four million possible strategy profiles (and that only for a single assumed preference distribution). However, we have been able to establish that a particular strategy, s^*, forms a pure symmetric Nash equilibrium when all five agents play it. We did this by selectively estimating the payoff submatrix for 53 profiles: one with all agents playing s^*, and 52 with four agents playing s^* and one agent unilaterally deviating to an alternate strategy. None of the deviations was successful, so all-s^* is a Nash equilibrium [Osepayshvili et al., 2005].

5.5 Discussion

The automation of markets and agents that trade in them opens up many new opportunities in market design and deployment. It also raises many new issues for strategic analysis: extending attention to challenging new market environments, and accounting for the wide strategic options available to computational agents. It may seem ironic (particularly for a chapter in the Handbook of Agent-Based Computational Economics) that we conclude by sketching a methodology that uses agent-based simulations in service of game-theoretic analysis. Indeed, many of the works in the agent-based economic literature [Tesfatsion, 2003] aim expressly to overcome the limitations of overly stylized analyses of markets abstracted from their microstructure. We share this aim, but emphasize the possibility of addressing issues of model fidelity without necessarily discarding the underlying theoretical framework. Computational modeling will likely prove just as valuable in service of game-theoretic analyses, as it can be as an alternative.

Acknowledgments The authors gratefully acknowledge the helpful suggestions made by the editors, by five hard-working, generous, but anonymous reviewers, and by the other authors in this volume and additional participants at a book workshop in May 2004 at which we presented a first draft. This work was supported in part by grants CSS-9988715 and IIS-0414710 from the National Science Foundation.

References

Abdulkadiroglu, A., Pathak, P., Roth, A., and Sönmez, T. (2005). The boston public school match. *American Economic Review Papers and Proceedings*.

Amir, Y., Awerbuch, B., Barak, A., Borgstrom, R. S., and Keren, A. (2000). An opportunity cost approach for job assignment in a scalable computing cluster. *IEEE Transactions on Parallel and Distributed Systems*, 11:760–768.

Amir, Y., Awerbuch, B., and Borgstrom, R. S. (1998). The Java market: Transforming the Internet into a metacomputer. Technical Report CNDS-98-2, The Johns Hopkins University.

Anastadiadi, A., Kapidakis, S., Nikolaou, C., and Sairamesh, J. (1998). A computational economy for dynamic load balancing and data replication. In *First International Conference on Information and Computation Economies*, pages 166–180, Charleston, SC.

Armantier, O., Florens, J.-P., and Richard, J.-F. (2003). Empirical game theoretic models: Constrained equilibrium and simulations. Technical report, SUNY Stony Brook.

Arrow, K. (1964). On the role of securities in the optimal allocation of risk bearing. *Review of Economic Studies*, 31(2):91–96.

Arunachalam, R. and Sadeh, N. M. (2005). The supply chain trading agent competition. *Electronic Commerce Research and Applications*, 4:63–81.

Ausubel, L. and Milgrom, P. (2002). Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1(1):Article 1.

Avery, C., Resnick, P., and Zeckhauser, R. (1999). The market for evaluations. *AER*, 89:564–584.

Axelrod, R. (1984). *The Evolution of Cooperation*. Basic, New York.

Baldauf, B. and Santoni, G. J. (1991). Stock-price volatility: Some evidence from an ARCH model. *Journal of Futures Markets*, 11:191–200.

Banks, J. S., Ledyard, J. O., and Porter, D. P. (1989). Allocating uncertain and unresponsive resources: An experimental approach. *RAND Journal of Economics*, 20:1–25.

Baruah, S., Koren, G., Mao, D., Mishra, B., Raghunathan, A., Rosier, L., Shasha, D., and Wang, F. (1992). On the competitiveness of on-line real-time task scheduling. *Real-Time Systems*, 4(2):125–144.

Bass, T. A. (1999). *The Predictors*. Henry Holt and Company.

Belew, R. K. (2000). *Finding Out About*. Cambridge University Press.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.

Bertsekas, D. P. (1992). Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications*, 1:7–66.

Bikhchandani, S., de Vries, S., Schummer, J., and Vohra, R. V. (2002). Linear programming and Vickrey auctions. In *Mathematics of the Internet: E-Auction and Markets*, volume 127 of *IMA Volumes in Mathematics and its Applications*, pages 75–116. Springer.

Bikhchandani, S. and Mamer, J. W. (1997). Competitive equilibrium in an exchange economy with indivisibilities. *Journal of Economic Theory*, 74:385–413.

Boutilier, C. and Hoos, H. H. (2001). Bidding languages for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 1211–1217, Seattle, Wash.

Brennan, M. J. and Schwartz, E. S. (1990). Arbitrage in stock index futures. *Journal of Business*, 63:S7–S31.

Brewer, P. J. (1999). Decentralized computation procurement and computational robustness in a smart market. *Economic Theory*, 13(1):41–92.

Brewer, P. J. and Plott, C. R. (1996). A binary conflict ascending price (Bicap) mechanisms for the decentralized allocation of the right to use railroad tracks. *International Journal of Industrial Organization*, 14(6):857–886.

Brewer, P. J. and Plott, C. R. (2002). A decentralized, smart market solution to a class of back-haul transportation problems: Concept and experimental test beds. *Interfaces*, 32(5):13–36.

Brooks, C. H. and Durfee, E. H. (2003). Congregation formation in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 7:145–170.

Bykowsky, M. M., Cull, R. J., and Ledyard, J. O. (2000). Mutually destructive bidding: The FCC auction design problem. *Journal of Regulatory Economics*, 17:205–228.

Cameron, L. and Cramton, P. (1999). The role of the ISO in U.S. electricity markets: A review of restructuring in California and PJM. *Electricity Journal*, 12(3):71–81.

Chaum, D. (1992). Achieving electronic privacy. *Scientific American*, 267(2):96–101.

Chen, Y. (2003). An experimental study of the serial and average cost pricing mechanisms. *Journal of Public Economics*, 87(9-10):2305–2335.

Cheng, S.-F., Leung, E., Lochner, K. M., O'Malley, K., Reeves, D. M., Schvartzman, L. J., and Wellman, M. P. (2005). Walverine: A Walrasian trading agent. *Decision Support Systems*, 39:169–184.

Cheng, S.-F., Wellman, M. P., and Perry, D. G. (2003). Market-based resource allocation for information-collection scenarios. In *IJCAI-03 Workshop on Multi-Agent for Mass User Support*, Acapulco.

Clarke, E. (1971). Multipart pricing of public goods. *Public Choice*, 8:19–33.

Cliff, D. (1998). Evolving parameter sets for adaptive trading agents in continuous double-auction markets. In *Agents-98 Workshop on Artificial Societies and Computational Markets*, pages 38–47, Minneapolis, MN.

Cliff, D. (2003). Explorations in evolutionary design of online auction market mechanisms. *Electronic Commerce Research and Applications*, 2:162–175.

Cocchi, R., Estrin, D., Shenker, S. J., and Zhang, L. (1991). A study of priority pricing in multiple service class networks. In *ACM SIGCOMM*, pages 123–130.

Cocchi, R., Estrin, D., Shenker, S. J., and Zhang, L. (1993). Pricing in computer networks: Motivation, formulation and example. *IEEE Transactions on Networking*, 1(6):614–627.

Cohen, A. (2002). *The Perfect Store: Inside eBay*. Little, Brown, and Company.

Conen, W. and Sandholm, T. (2001). Minimal preference elicitation in combinatorial auctions. In *IJCAI-2001 Workshop on Economic Agents, Models, and Mechanisms*, pages 71–80, Seattle, WA.

Conen, W. and Sandholm, T. (2002). Partial-revelation VCG mechanisms for combinatorial auctions. In *Eighteenth National Conference on Artificial Intelligence*, pages 367–372, Edmonton.

Conitzer, V. and Sandholm, T. (2003). Complexity results about Nash equilibria. In *Eighteenth International Joint Conference on Artificial Intelligence*, pages 765–771, Acapulco, Mexico.

Cox, L. P. and Noble, B. D. (2003). Samsara: Honor among thieves in peer-to-peer storage. In *Nineteenth ACM Symposium on Operating Systems Principles*, Bolton Landing, NY.

Cramton, P. (1995). Money out of thin air: The nationwide narrowband PCS auction. *Journal of Economics and Management Strategy*, 4:267–343.

Cramton, P., Shoham, Y., and Steinberg, R., editors (forthcoming 2005). *Combinatorial Auctions*. MIT Press.

Cranor, L. F. and Resnick, P. (2000). Protocols for automated negotiations with buyer anonymity and seller reputations. *Netnomics*, 2:1–23.

Crocker, S. (1999). The siren song of Internet micropayments. *iMP: The Magazine on Information Impacts*.

Csirik, J. A., Littman, M. L., Singh, S., and Stone, P. (2001). FAucS: An FCC spectrum auction simulator for autonomous bidding agents. In *Second International Workshop on Electronic Commerce*, volume 2232 of *Lecture Notes in Computer Science*, pages 139–151. Springer-Verlag.

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., and Weerawarana, S. (2002). Unraveling the web services web: An introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing*, 6:86–93.

Das, R., Hanson, J. E., Kephart, J. O., and Tesauro, G. (2001). Agent-human interactions in the continuous double auction. In *Seventeenth International Joint Conference on Artificial Intelligence*, pages 1169–1176, Seattle, WA.

Dash, R. K., Jennings, N. R., and Parkes, D. C. (2003). Computational-mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):40–47.

de Vries, S. and Vohra, R. (2003). Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3):284–309.

Debreu, G. (1954). Valuation equilibrium and Pareto optimum. In *Proceedings of the National Academy of Sciences U.S.A.*, volume 40, pages 588–592.

Dellarocas, C. (2003). The digitization of word-of-mouth: Promise and challenges of online reputation mechanisms. *Management Science*, 49:1407–1424.

Di Noia, T., Di Sciascio, E., Donini, F. M., and Mongiello, M. (2004). A system for principled matchmaking in an electronic marketplace. *International Journal of Electronic Commerce*, 8:9–37.

Doorenbos, R. B., Etzioni, O., and Weld, D. S. (1997). A scalable comparison-shopping agent for the world-wide web. In *First International Conference on Autonomous Agents*, pages 39–48.

Epstein, J. M. (2005). Remarks on the foundations of agent-based generative social science. This Handbook.

Eriksson, J. and Janson, S. (2002). The Trading Agent Competition: TAC 2002. *ERCIM News*, 51.

Faratin, P. and de Walle, B. V. (2002). Agent preference relations: Strict, equivalent and incomparables. In *Autonomous Agents and Multi-Agent Systems*, pages 1317–1324, Bologna,Italy. AAAI Press.

Feigenbaum, J., Papadimitriou, C., Sami, R., and Shenker, S. (forthcoming). A BGP-based mechanism for lowest-cost routing. *Distributed Computing*.

Feigenbaum, J., Papadimitriou, C., and Shenker, S. (2001). Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63:21–41.

Ferguson, D., Yemini, Y., and Nikolaou, C. (1988). Microeconomic algorithms for load balancing in distributed computer systems. In *Eighth International Conference on Distributed Computing Systems*, pages 491–499.

Finholt, T. A. (2003). Collaboratories as a new form of scientific organization. *Economics of Innovation and New Technology*, 12:5–25.

Finholt, T. A. and Olson, G. M. (1997). From laboratories to collaboratories: A new organizational form for scientific collaboration. *Psychological Science*, 8:28–36.

Forsythe, R. and Lundholm, R. (1990). Information aggregation in an experimental market. *Econometrica*, 58(2):309–347.

Forsythe, R., Nelson, F., Neumann, G. R., and Wright, J. (1992). Anatomy of an experimental political stock market. *American Economic Review*, 82(5):1142–1161.

Friedman, D. (1991). Evolutionary games in economics. *Econometrica*, 59(3):637–666.

Friedman, D. (1993). The double auction market institution: A survey. In Friedman and Rust [1993], pages 3–25.

Friedman, D. and Rust, J., editors (1993). *The Double Auction Market*. Addison-Wesley.

Friedman, E. and Parkes, D. (2003). Pricing wifi at starbucks: Issues in online mechanism design. In *Proc. Fourth Conf. on Electronic Commerce (EC'03)*. ACM.

Fritschi, C. and Dorer, K. (2002). Agent-oriented software engineering for successful TAC participation. In *First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Bologna.

Fudenberg, D. and Tirole, J. (1991). Perfect Bayesian equilibrium and sequential equilibrium. *Journal of Economic Theory*, 53:236–260.

Gagliano, R., Fraser, M., and Schaefer, M. (1995). Auction allocation of computing reserves. *Communications of the ACM*, 38(6):88–99.

Garman, M. B. (1976). Market microstructure. *Journal of Financial Economics*, 3:257–275.

Gjerstad, S. (2004). The impact of bargaining pace on double auction dynamics. Technical report, University of Arizona.

Gjerstad, S. and Dickhaut, J. (1998). Price formation in double auctions. *Games and Economic Behavior*, 22:1–29.

Gode, D. K. and Sunder, S. (1993). Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101:119–137.

Goldman, C. V., Kraus, S., and Shehory, O. (2001). Equilibria strategies for selecting sellers and satisfying buyers. In Klusch, M. and Zambonelli, F., editors, *Cooperative Information Agents V*, volume 2182 of *Lecture Notes in Artificial Intelligence*, pages 166–177. Springer-Verlag.

Golle, P., Leyton-Brown, K., Mironov, I., and Lillibridge, M. (2001). Incentives for sharing in peer-to-peer networks. In *Second International Workshop on Electronic Commerce*, volume 2232 of *Lecture Notes in Computer Science*. Springer-Verlag.

Greenwald, A. (2003). The 2002 trading agent competition: An overview of agent strategies. *AI Magazine*, 24(1):83–91.

Greenwald, A. and Boyan, J. (2001). Bidding algorithms for simultaneous auctions: A case study. In *Third ACM Conference on Electronic Commerce*, pages 115–124, Tampa, FL.

Greenwald, A. and Boyan, J. (2004). Bidding under uncertainty: Theory and experiments. In *Twentieth Conference on Uncertainty in Artificial Intelligence*, pages 209–216, Banff.

Greenwald, A. and Stone, P. (2001). The First International Trading Agent Competition: Autonomous bidding agents. *IEEE Internet Computing*, 5:52–60.

Groves, T. (1973). Incentives in teams. *Econometrica*, 41:617–631.

Guerin, F. and Pitt, J. (2002). Guaranteeing properties for e-commerce systems. In *Agent-Mediated Electronic Commerce IV*, volume 2153 of *Lecture Notes in Computer Science*, pages 253–272. Springer-Verlag.

Gul, F. and Stacchetti, E. (1999). Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87:95–124.

Gupta, A., Stahl, D. O., and Whinston, A. B. (1996). An economic approach to network computing with priority classes. *Journal of Organizational Computing and Electronic Commerce*, 6:71–95.

Gupta, R. and Somani, A. K. (2004). CompuP2P: An architecture for sharing of compute power in peer-to-peer networks with selfish nodes. In *Second Workshop on the Economics of Peer-to-Peer Systems*, Cambridge, MA.

Hajiaghayi, M. T., Kleinberg, R., and Parkes, D. C. (2004). Adaptive limited-supply online auctions. In *Fifth ACM Conference on Electronic Commerce*, pages 71–80.

Hanson, R. (2003). Combinatorial information market design. *Information Systems Frontiers*, 5(1):107–119.

Hart, S. and Mas-Colell, A. (2000). A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150.

Hill, W., Stead, L., Rosenstein, M., and Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. In *Conference on Human Factors in Computing Systems*, pages 194–201, Denver. ACM.

Hofreiter, B., Huemer, C., and Klas, W. (2002). ebXML: Status, research issues, and obstacles. In *Twelfth International Workshop on Research Issues in Data Engineering*, pages 7–16, San José, CA.

Hommes, C. (2005). Heterogeneous agent models in economics and finance. This Handbook.

Hurwicz, L. (1960). Optimality and informational efficiency in resource allocation processes. In Arrow, K., Karlin, S., and Suppes, P., editors, *Mathematical Methods in the Social Sciences*, pages 27–46. Stanford University Press, Stanford, CA.

Jackson, E. M. (2004). *The PayPal Wars*. World Ahead Publishing.

Jordan, J. S. (1982). The competitive allocation process is informationally efficient uniquely. *Journal of Economic Theory*, 28(1):1–18.

Karaul, M., Korilis, Y. A., and Orda, A. (1998). A market-based architecture for management of geographically dispersed, replicated web servers. In *First International Conference on Information and Computation Economies*, pages 158–165, Charleston, SC.

Kearns, M., Littman, M. L., and Singh, S. (2001). Graphical models for game theory. In *Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 253–260, Seattle.

Kearns, M. and Ortiz, L. (2003). The Penn-Lehman automated trading project. *IEEE Intelligent Systems*, 18(6):22–31.

Kelly, F., Maulloo, A., and Tan, D. (1998). Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252.

Kelly, T. P., Chan, Y. M., Jamin, S., and MacKie-Mason, J. K. (1999). Biased replacement policies for web caches: Differential quality-of-service and aggregate user value. In *Fourth International Web Caching Workshop*, San Diego, CA.

Kelly, T. P., Jamin, S., and MacKie-Mason, J. K. (forthcoming 2005). Variable QoS from shared web caches: User-centered design and value-sensitive replacement. In McKnight, L., editor, *Internet Service Quality Economics*. MIT Press.

Kelso, Jr., A. S. and Crawford, V. P. (1982). Job matching, coalition formation and gross substitutes. *Econometrica*, 50:1483–1504.

Kephart, J. O. and Greenwald, A. R. (2002). Shopbot economics. *Autonomous Agents and Multiagent Systems*, 5:255–287.

Kephart, J. O., Hanson, J. E., and Sairamesh, J. (1998). Price and niche wars in a free-market economy of software agents. *Artificial Life*, 4:1–23.

Koller, D., Megiddo, N., and von Stengel, B. (1996). Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14:247–259.

Koren, G. and Shasha, D. (1995). D-over: An optimal on-line scheduling algorithm for overloaded uniprocessor real-time systems. *SIAM Journal on Computing*, 24(2):318–339.

Korilis, Y. A., Lazar, A. A., and Orda, A. (1995). Architecting noncooperative networks. *IEEE Journal on Selected Areas in Communications*, 13(7):1241–1251.

Koza, J. R. (1991). Evolution and co-evolution of computer programs to control independently-acting agents. In *Proceedings of the first international conference on simulation of adaptive behavior: From animals to animats*, pages 366–375, Cambridge, MA. MIT Press.

Krishna, V. (2002). *Auction Theory*. Academic Press.

Krulwich, B. T. (1996). The BargainFinder agent: Comparison price shopping on the Internet. In Williams, J., editor, *Bots and Other Internet Beasties*, chapter 13, pages 257–263. Sams Publishing.

Kurose, J. F. and Simha, R. (1986). A microeconomic approach to optimal file allocation. In *Sixth International Conference on Distributed Computing Systems*, pages 28–35.

Kurose, J. F. and Simha, R. (1989). A microeconomic approach to optimal resource allocation in distributed systems. *IEEE Transactions on Computers*, 38(5):705–717.

Lanzi, P. L. and Strada, A. (2002). A statistical analysis of the trading agent competition 2001. *SIGecom Exchanges*, 3(2):1–8.

LeBaron, B. (2005). Agent-based computational finance. This Handbook.

Ledyard, J. and Szakaly, K. (1994). Designing organizations for trading pollution rights. *Journal of Economic Behavior and Organization*, 25:167–196.

Leyton-Brown, K. (2003). *Resource Allocation in Competitive Multiagent Systems*. PhD thesis, Stanford University.

Li, L. and Horrocks, I. (2004). A software framework for matchmaking based on semantic web technology. *International Journal of Electronic Commerce*, 8:39–60.

Lochner, K. M. and Wellman, M. P. (2004). Rule-based specification of auction mechanisms. In *Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 818–825, New York.

MacKie-Mason, J. K., Murphy, J., and Murphy, L. (1996a). Feedback and efficiency in ATM networks. In *International Conference on Communications*, volume 2, pages 1045–1049. IEEE.

MacKie-Mason, J. K., Murphy, J., and Murphy, L. (1996b). The role of responsive pricing in the internet. In Baily, J. and McKnight, L., editors, *Internet Economics*, pages 279–304. MIT Press, Cambridge.

MacKie-Mason, J. K., Osepayshvili, A., Reeves, D. M., and Wellman, M. P. (2004). Price prediction strategies for market-based scheduling. In *Fourteenth International Conference on Automated Planning and Scheduling*, pages 244–252, Whistler, BC.

MacKie-Mason, J. K. and Varian, H. R. (1994a). Economic FAQs about the internet. *Journal of Economic Perspectives*, 8:75–96.

MacKie-Mason, J. K. and Varian, H. R. (1994b). Generalized Vickrey auctions. Working paper, University of Michigan.

MacKie-Mason, J. K. and Varian, H. R. (1995a). Pricing congestible network resources. *IEEE Journal on Selected Areas of Communications*, 13(7):1141–1149.

MacKie-Mason, J. K. and Varian, H. R. (1995b). Pricing the Internet. In Kahin, B. and Keller, J., editors, *Public Access to the Internet*, pages 269–314. MIT Press.

MacKie-Mason, J. K. and Varian, H. R. (1996). Some economics of the Internet. In Sichel, W., editor, *Networks, Infrastructure and the New Task for Regulation*. University of Michigan Press, Ann Arbor, MI.

MacKie-Mason, J. K. and White, K. (1997). Evaluating and selecting digital payment mechanisms. In Rosston, G. L. and Waterman, D., editors, *Interconnection and the Internet: Selected Papers from the 24th Annual Telecommunications Policy Research Conference*. Lawrence Erlbaum.

Malone, T. W. (1987). Modeling coordination in organizations and markets. *Management Science*, 33:1317–1332.

Manasse, M. S. (1995). The Millicent protocols for electronic commerce. In *First USENIX Workshop on Electronic Commerce*, pages 117–123, New York.

Marks, R. (2005). Market design. This Handbook.

Mas-Colell, A., Whinston, M. D., and Green, J. R. (1995). *Microeconomic Theory*. Oxford University Press, New York.

Maskin, E. and Riley, J. (1989). Optimal multi-unit auctions. In Hahn, F., editor, *The Economics of Missing Markets, Information, and Games*, chapter 14, pages 312–335. Clarendon Press, Oxford.

McAfee, R. P. and McMillan, J. (1996). Analyzing the airwaves auction. *Journal of Economic Perspectives*, 10(1):159–175.

McCabe, K. A., Rassenti, S. J., and Smith, V. L. (1989). Designing 'smart' computer-assisted markets: An experimental auction for gas networks. *European Journal of Political Economy*, 5:259–283.

McCabe, K. A., Rassenti, S. J., and Smith, V. L. (1991). Smart computer-assisted markets. *Science*, 254:534–538.

McKelvey, R. D. and McLennan, A. (1996). Computation of equilibria in finite games. In *Handbook of Computational Economics*. Elsevier.

McKelvey, R. D., McLennan, A., and Turocy, T. (1992). Gambit game theory analysis software and tools.

Menczer, F., Street, W. N., and Monge, A. E. (2002). Adaptive assistants for customized e-shopping. *IEEE Intelligent Systems*, 17:12–19.

Micali, S. and Rivest, R. L. (2002). Micropayments revisited. In *Cryptographer's Track at the RSA Conference*, volume 2271 of *Lecture Notes in Computer Science*, pages 149–263, Heidelberg. Springer-Verlag.

Milgrom, P. (2000). Putting auction theory to work: The simultaneous ascending auction. *Journal of Political Economy*, 108:245–272.

Miller, J. H. (1988). The evolution of automata in the repeated prisoners dilemma. In *Two Essays on the Economics of Imperfect Information*, Ph.D. dissertation, Ann Arbor. University of Michigan.

Moulin, H. and Shenker, S. (2001). Strategyproof sharing of submodular access costs: Budget balance versus efficiency. *Economic Theory*, 18:511–533.

Mount, K. and Reiter, S. (1974). The informational size of message spaces. *Journal of Economic Theory*, 8:161–192.

Mullen, T. and Breese, J. (1998). Experiments in designing computational economies for mobile users. In *First International Conference on Information and Computation Economies*, pages 19–27, Charleston, SC.

Myerson, R. and Satterthwaite, M. (1981). Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29 (1983):265–281.

Nash, J. (1950). Non-cooperative games. Dept. of Mathematics, Princeton University.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, 7:308–313.

Nicolaisen, J., Petrov, V., and Tesfatsion, L. (2001). Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Transactions on Evolutionary Computation*, 5:504–523.

Nisan, N. (2000). Bidding and allocation in combinatorial auctions. In *Second ACM Conference on Electronic Commerce*, pages 1–12, Minneapolis, MN.

Nisan, N. and Ronen, A. (2000). Computationally feasible VCG mechanisms. In *Second ACM Conference on Electronic Commerce*, pages 242–252, Minneapolis, MN.

Nisan, N. and Ronen, A. (2001). Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196.

Nisan, N. and Segal, I. (2004). The communication requirements of efficient allocations and supporting prices. Unpublished manuscript, Stanford University.

O'Hara, M. (1995). *Market Microstructure Theory*. Blackwell.

O'Mahony, D., Peirce, M., and Tewari, H. (1997). *Electronic Payment Systems*. Artech House.

Osepayshvili, A., Wellman, M. P., Reeves, D., and MacKie-Mason, J. K. (2005). Self-confirming price prediction for bidding in simultaneous ascending auctions. Working paper, University of Michigan.

Papadimitriou, C. H. (2001). Algorithms, games, and the Internet. In *Thirty-Third Annual ACM Symposium on the Theory of Computing*, pages 749–753, Hersonissos, Greece.

Parkes, D. and Singh, S. (2003). An mdp-based approach to online mechanism design. In *Proc. 17th Annual Conf. on Neural Information Processing Systems (NIPS'03)*.

Parkes, D. C. (1999). iBundle: An efficient ascending price bundle auction. In *ACM Conference on Electronic Commerce*, pages 148–157.

Parkes, D. C. (2004). Auction design with costly preference elicitation. *Annals of Mathematics and AI*. Special Issue on the Foundations of Electronic Commerce, Forthcoming.

Parkes, D. C. and Ungar, L. H. (2000). Iterative combinatorial auctions: Theory and practice. In *Seventeenth National Conference on Artificial Intelligence*, pages 74–81.

Parkes, D. C. and Ungar, L. H. (2002). An ascending-price generalized Vickrey auction. Manuscript, Harvard University.

Pennock, D. M. (2004). A dynamic pari-mutuel market for hedging, wagering, and information aggregation. In *Fifth ACM Conference on Electronic Commerce*, pages 170–179.

Pennock, D. M. and Wellman, M. P. (2000). Compact securities markets for Pareto optimal reallocation of risk. In *Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 481–488, Stanford.

Peters, M. and Severinov, S. (2001). Internet auctions with many traders. Technical report, University of Toronto.

Phelps, S., Parsons, S., McBurney, P., and Sklar, E. (2002). Co-evolution of auction mechanisms and trading strategies: Towards a novel approach to microeconomic design. In *GECCO-02 Workshop on Evolutionary Computation in Multi-Agent Systems*, pages 65–72.

Plott, C. R. (1986). Laboratory experiments in economics: The implications of posted-price institutions. *Science*, 232:732–738.

Plott, C. R. and Sunder, S. (1988). Rational expectations and the aggregation of diverse information in laboratory security markets. *Econometrica*, 56(5):1085–1118.

Polk, C., Hanson, R., Ledyard, J., and Ishikida, T. (2003). Policy analysis market: An electronic commerce application of a combinatorial information market. In *Fourth ACM Conference on Electronic Commerce*, pages 272–273, San Diego, CA.

Porter, R. (2004). Mechanism design for online real-time scheduling. In *Fifth ACM Conference on Electronic Commerce*, pages 61–70, New York.

Porter, R., Nudelman, E., and Shoham, Y. (2004). Simple search methods for finding a Nash equilibrium. In *Nineteenth National Conference on Artificial Intelligence*, San Jose, CA.

Putchala, R. P., Morris, V. N., Kazhanchi, R., Raman, L., and Shekhar, S. (2002). kavayaH: A trading agent developed for TAC-02. Technical report, Oracle India.

Pynadath, D. V. and Tambe, M. (2002). The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:389–423.

Rassenti, S. J., Smith, V. L., and Bulfin, R. L. (1982). A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, 13:402–417.

Reeves, D. M., Wellman, M. P., MacKie-Mason, J. K., and Osepayshvili, A. (2005). Exploring bidding strategies for market-based scheduling. *Decision Support Systems*, 39:67–85.

Regev, O. and Nisan, N. (1998). The POPCORN market: An online market for computational resources. In *First International Conference on Information and Computation Economies*, pages 148–157, Charleston, SC.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW '94: Conference on Computer Supported Coorperative Work*, pages 175–186, Chapel Hill. ACM.

Resnick, P. and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

Resnick, P. and Zeckhauser, R. (2002). Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In Baye, M. R., editor, *The Economics of the Internet and E-Commerce*, volume 11 of *Advances in Applied Microeconomics*. Elsevier Science.

Resnick, P., Zeckhauser, R., Friedman, E., and Kuwabara, K. (2002). Reputation systems. *Communications of the ACM*, 43(12):45–48.

Riedl, J. and Konstan, J. A. (2002). *Word of Mouse: The Marketing Power of Collaborative Filtering*. Warner Books.

Rosenschein, J. S. and Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press.

Roth, A., Sönmez, T., and Unver, U. (2005). A kidney exchange clearinghouse in new england. *American Economic Review Papers and Proceedings*.

Roth, A. E. and Ockenfels, A. (2002). Last-minute bidding and the rules for ending second-price auctions: Evidence from eBay and Amazon auctions on the Internet. *American Economic Review*, 92:1093–1103.

Roth, A. E. and Peranson, E. (1999). The redesign of the matching market for American physicians: Some engineering aspects of economic design. *American Economic Review*, 89:748–780.

Rothkopf, M. H., Pekeč, A., and Harstad, R. M. (1998). Computationally manageable combinatorial auctions. *Management Science*, 44:1131–1147.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition.

Rust, J., Miller, J. H., and Palmer, R. (1994). Characterizing effective trading strategies: Insights from a computerized double auction tournament. *Journal of Economic Dynamics and Control*, 18:61–96.

Sakurai, Y., Yokoo, M., and Matsubara, S. (1999). A limitation of the generalized Vickrey auction in electronic commerce: Robustness against false-name bids. In *Sixteenth National Conference on Artificial Intelligence*, pages 86–92, Orlando, FL.

Sandholm, T. and Suri, S. (2003). BOB: Improved winner determination in combinatorial auctions and generalizations. *Artificial Intelligence*, 145:33–58.

Sandholm, T. W. and Lesser, V. R. (1997). Coalitions among computationally bounded agents. *Artificial Intelligence*, 94:99–137.

Scarf, H. (1973). *The Computation of Economic Equilibria*. Cowles Foundation Monograph No. 24. Yale University Press, New Haven.

Schafer, J. B., Konstan, J. A., and Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5:115–153.

Schuster, P. and Sigmund, K. (1983). Replicator dynamics. *Journal of Theoretical Biology*, 100:533–538.

Selten, R., Mitzkewitz, M., and Uhlich, G. R. (1997). Duopoly strategies programmed by experienced players. *Econometrica*, 65(3):517–555.

Shenker, S. (1994). Making greed work in networks: A game-theoretic analysis of switch service disciplines. In *ACM SIGCOMM*, pages 47–57.

Shenker, S. (1995). Service models and pricing policies for an integrated services Internet. In Kahin, B. and Keller, J., editors, *Public Access to the Internet*, pages 315–337. MIT Press.

Sherstov, A. and Stone, P. (2004). Three automated stock-trading agents: A comparative study. In *AAMAS-04 Workshop on Agent-Mediated Electronic Commerce*, New York.

Sirbu, M. and Tygar, J. D. (1995). NetBill: An Internet commerce system optimized for network delivered services. *IEEE Personal Communications*, 2(4):34–39.

Smith, V. L. (1962). An experimental study of competitive market behavior. *Journal of Political Economy*, 70:111–137.

Stone, P., Littman, M. L., Singh, S., and Kearns, M. (2001). ATTac-2000: An adaptive autonomous bidding agent. *Journal of Artificial Intelligence Research*, 15:189–206.

Stone, P., Schapire, R. E., Littman, M. L., Csirik, J. A., and McAllester, D. (2003). Decision-theoretic bidding based on learned density models in simultaneous, interacting auctions. *Journal of Artificial Intelligence Research*, 19:209–242.

Stonebraker, M., Aoki, P. M., Litwin, W., Pfeffer, A., Sah, A., Sidell, J., Staelin, C., and Yu, A. (1996). Mariposa: A wide-area distributed database system. *VLDB Journal*, 5:48–63.

Sutherland, I. E. (1968). A futures market in computer time. *Communications of the ACM*, 11(6):449–451.

Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124.

Taylor, P. and Jonker, L. (1978). Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 40:145–156.

Tesauro, G. and Das, R. (2001). High-performance bidding agents for the continuous double auction. In *Third ACM Conference on Electronic Commerce*, pages 206–209, Tampa, FL.

Tesfatsion, L. (2003). Agent-based computational economics. Technical report, Iowa State University.

Thomas, P., Teneketzis, D., and MacKie-Mason, J. K. (2002). A market-based approach to optimal resource allocation in integrated-services connection-oriented networks. *Operations Research*, 50:603–616.

Ünver, U. (2001). Backward unraveling over time: The evolution of strategic behavior in the entry-level British medical labor markets. *Journal of Economic Dynamics and Control*, 25(6–7):1039–1080.

Vetsikas, I. A. and Selman, B. (2003). A principled study of the design tradeoffs for autonomous trading agents. In *Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 473–480, Melbourne, Australia.

Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37.

Waldspurger, C. A., Hogg, T., Huberman, B. A., Kephart, J. O., and Stornetta, S. (1992). Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18(2):103–117.

Walsh, W. E., Das, R., Tesauro, G., and Kephart, J. O. (2002). Analyzing complex strategic interactions in multi-agent systems. In *AAAI-02 Workshop on Game-Theoretic and Decision-Theoretic Agents*, Edmonton.

Walsh, W. E., Parkes, D., and Das, R. (2003). Choosing samples to compute heuristic-strategy nash equilibrium. In *AAMAS-03 Workshop on Agent-Mediated Electronic Commerce*, Melbourne, Australia.

Weber, R. J. (1997). Making more from less: Strategic demand reduction in the FCC spectrum auctions. *Journal of Economics and Management Strategy*, 6:529–548.

Wellman, M. P. (2004). Online marketplaces. In Singh, M., editor, *Practical Handbook of Internet Computing*. CRC Press.

Wellman, M. P., Cheng, S.-F., Reeves, D. M., and Lochner, K. M. (2003a). Trading agents competing: Performance, progress, and market effectiveness. *IEEE Intelligent Systems*, 18(6):48–53.

Wellman, M. P., Estelle, J., Singh, S., Vorobeychik, Y., Kiekintveld, C., and Soni, V. (2005). Strategic interactions in a supply chain game. *Computational Intelligence*, 21:1–26.

Wellman, M. P., Greenwald, A., Stone, P., and Wurman, P. R. (2003b). The 2001 trading agent competition. *Electronic Markets*, 13:4–12.

Wellman, M. P., Reeves, D. M., Lochner, K. M., and Vorobeychik, Y. (2004). Price prediction in a trading agent competition. *Journal of Artificial Intelligence Research*, 21:19–36.

Wellman, M. P., Walsh, W. E., Wurman, P. R., and MacKie-Mason, J. K. (2001a). Auction protocols for decentralized scheduling. *Games and Economic Behavior*, 35:271–303.

Wellman, M. P., Wurman, P. R., O'Malley, K., Bangera, R., Lin, S.-d., Reeves, D., and Walsh, W. E. (2001b). Designing the market game for a trading agent competition. *IEEE Internet Computing*, 5:43–51.

Wolfram, C. (1998). Strategic bidding in a multi-unit auction: An empirical analysis of bids to supply electricity in England and Wales. *RAND Journal of Economics*, 29:703–725.

Wolski, R., Plank, J. S., Brevik, J., and Bryan, T. (2001). Analyzing market-based resource allocation strategies for the computational grid. *International Journal of High Performance Computing Applications*, 15:258–281.

Woods, W. W. A. (2002). *B2B Exchanges 2.0*. ISI Publications.

Wurman, P. R. and Wellman, M. P. (2000). AkBA: A progressive, anonymous-price combinatorial auction. In *Second ACM Conference on Electronic Commerce*, pages 21–29, Minneapolis, MN.

Wurman, P. R., Wellman, M. P., and Walsh, W. E. (2001). A parametrization of the auction design space. *Games and Economic Behavior*, 35:304–338.

Ygge, F. and Akkermans, H. (1996). Power load management as a computational market. In *Second International Conference on Multiagent Systems*, pages 393–400, Kyoto.

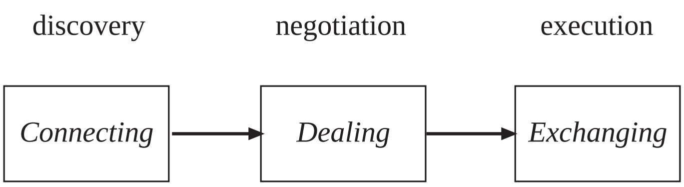

Figure 1: The fundamental steps of a market transaction.

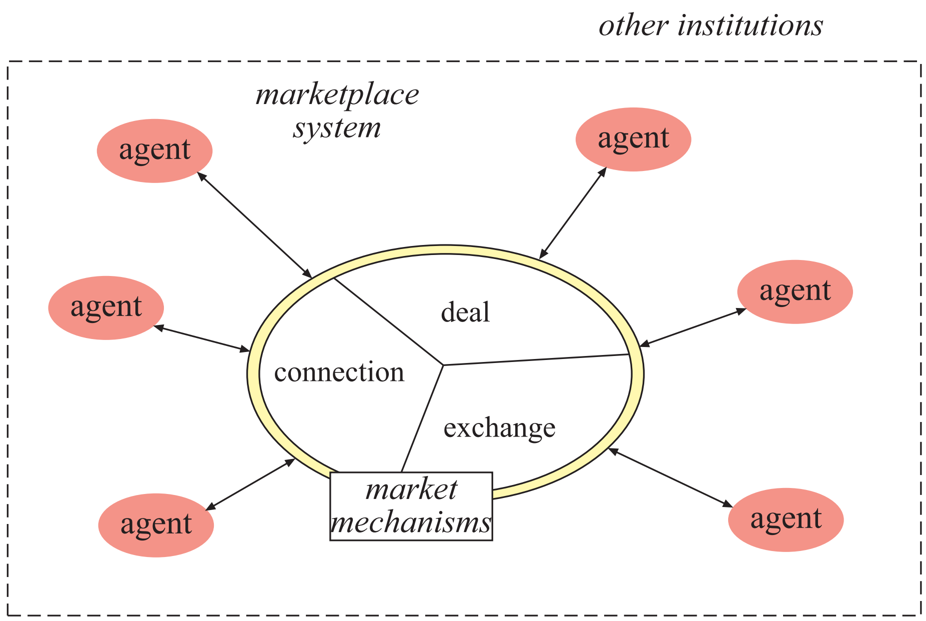

Figure 2: Schematic for a marketplace system.

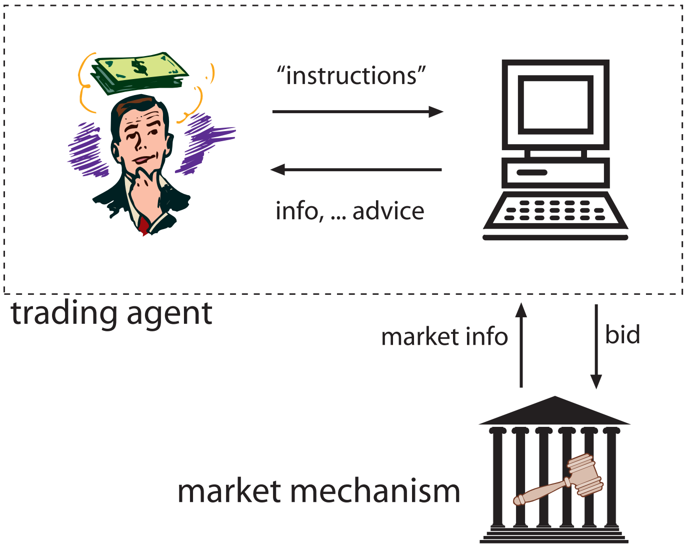

Figure 3: The trading agent interacts with a market mechanism by submitting bids in response to market information. The process can be automated to varying degrees, depending on the role of the computer in the process of translating instructions and market information.

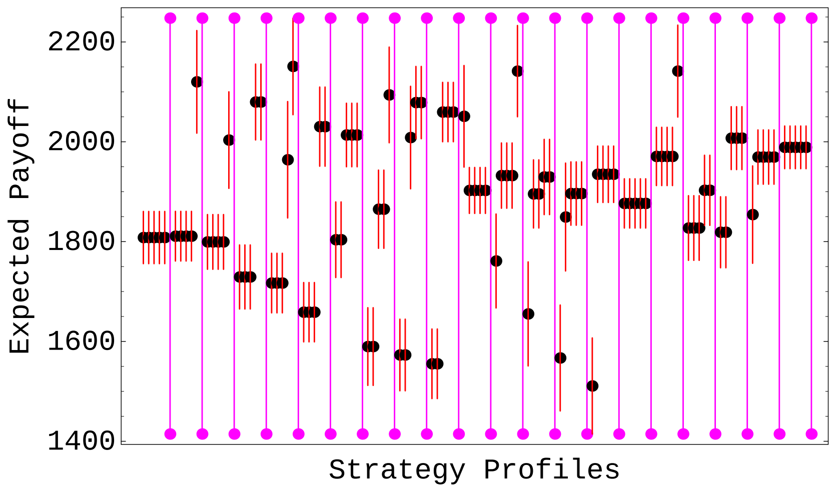

Figure 4: Payoff matrix for symmetric game with five agents choosing from strategies A,B,C. Each column corresponds to a strategy profile: {A,A,A,A,A} through {C,C,C,C,C} in lexicographic order. The jth dot within a column represents the mean payoff for the jth strategy in the profile. This payoff matrix is based on over 200 games simulated for each of the 21 distinct profiles. The error bars denote 95% confidence intervals.

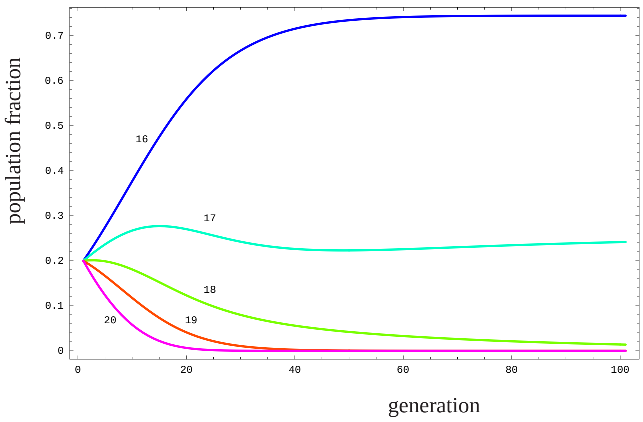

Figure 5: Replicator dynamics for a five-strategy version of a scheduling market