



# DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD

## On Learnable Mechanism Design

The Harvard community has made this article openly available.  
[Please share](#) how this access benefits you. Your story matters.

<b>Citation</b>	Parkes, David C. 2004. On learnable mechanism design. In Collectives and the design of complex systems, ed. K. Tumer and D. Wolpert, 107-131. New York, N.Y.: Springer.
<b>Accessed</b>	February 18, 2015 2:09:08 PM EST
<b>Citable Link</b>	<a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:4039782">http://nrs.harvard.edu/urn-3:HUL.InstRepos:4039782</a>
<b>Terms of Use</b>	This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA">http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA</a>

*(Article begins on next page)*

# Contents

<b>1</b>	<b>On Learnable Mechanism Design</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	Mechanism Design . . . . .	4
1.2.1	Mechanism Design: Preliminaries . . . . .	5
1.2.2	Example: Efficient Mechanism Design . . . . .	7
1.2.3	Computational Considerations . . . . .	8
1.2.4	Dynamic Considerations . . . . .	9
1.3	Collective Intelligence . . . . .	11
1.3.1	Collective Intelligence: Preliminaries . . . . .	12
1.3.2	Informative Local Utilities . . . . .	13
1.3.3	Back to VCG Mechanisms . . . . .	15
1.3.4	Example: A Congestion Game . . . . .	15
1.4	Learnable Mechanism Design for Episodic Systems . . . . .	16
1.4.1	The Groves-WLU Mechanism . . . . .	17
1.4.2	Example: Auction Game . . . . .	19
1.4.3	Example: Congestion Game . . . . .	22
1.5	Summary: Towards Learnable Mechanism Design . . . . .	24
1.6	References . . . . .	26



# 1

## On Learnable Mechanism Design

**David C. Parkes, Division of Engineering and Applied Sciences, Harvard University,  
parkes@eecs.harvard.edu**

**ABSTRACT** Computation is increasingly distributed across open networks, and performed by self-interested autonomous agents that represent individuals and businesses. Given that these computational agents are often in situations of strategic interaction, it is natural to turn to economics for ideas to control these systems. Mechanism design is particularly attractive in this setting, given its focus on the design of optimal rules to implement good system-wide outcomes despite individual self-interest. Yet, these rich computational environments present new challenges for mechanism design, for example because of system dynamics and because the computational cost of implementing particular equilibrium outcomes is also important. We discuss some of these challenges and provide a reinterpretation of the mathematics of collective intelligence in terms of *learnable mechanism design* for bounded-rational agents.

### 1.1 Introduction

Far from confined to a desktop machine, computation is now ubiquitous, and performed by heterogenous and networked computers, many of which are operated by and used on behalf of self-interested users. As such, computational systems increasingly exhibit the properties of economies, and it is natural to turn to incentive-based methods to construct systems with useful system-wide behaviors. Indeed, as an alternative to traditional cooperative or adversarial assumptions in computer science, a reasonable design principle for many distributed systems is that computational devices will be programmed to follow selfish, self-interested, behaviors [13, 52].

The framework of *economic mechanism design* (e.g. [27]) provides a rich theoretical backdrop for an economically-motivated approach to the control of decentralized computational systems. Mechanism design was first proposed as a method to implement social choices (e.g. [7, 23]). More recently, mechanism design has been adapted to implement outcomes that are individually-optimal for a single agent, for example in the setting of *optimal auction design* [41].

The key problem addressed within mechanism design is the *information problem*, in which agents have private information that the mechanism must elicit in order to implement a good system-wide outcome. Participants must receive ap-

appropriate incentives to reveal information truthfully. This incentive-compatibility requirement limits the outcomes that can be implemented in a game-theoretic equilibrium (e.g. [42]). However, in certain contexts it has been possible to propose *strategyproof* mechanisms in which no participant can manipulate an outcome to their own benefit whatever the strategy of any other agent [56].

In this chapter, we introduce some of the basic methods of mechanism design, and briefly discuss some of the new challenges in the application of ideas from mechanism design to distributed computational systems. We focus here on the particular problem of mechanism design in the presence of bounded-rational agents that employ simple learning methods to adjust towards an equilibrium. We suggest a collective-intelligence inspired approach for mechanism design in repeated games with these simple adaptive agents. No background is assumed in either mechanism design or collective intelligence.

We first review the mathematics of mechanism design and the mathematics of collective intelligence. Then, we reinterpret the *factoredness* and *informative* methods of collective intelligence in the context of *incentive-compatibility* (truth-revelation in equilibrium) and *learnable* mechanism design, in which payments are selected to enable computational agents to learn equilibrium strategies. In particular, we introduce a VCG-WLU mechanism, which is a hybrid of the Groves [23] family of mechanisms and the *wonderful life* local utility [58] of collective-intelligence. We present simple experimental results, for a simple auction problem and a simple congestion problem, to demonstrate the advantages of this approach to mechanism design. In closing, we consider a broader agenda of *learnable mechanism design* in which mechanisms are explicitly constructed to maximize their performance with respect to a model of limited agent-rationality.

## 1.2 Mechanism Design

We begin with a formal introduction to mechanism design theory, and then outline some problems with the approach as applied to distributed and dynamic computational systems. The problems arise because of practical limits on the amount of computation and communication that is reasonable within a system. For a more complete review of mechanism design, see Jackson [27], Chapter 23 of MasColell et al. [37], or Chapter 2 of Parkes [46].

The mechanism design (MD) approach to solving problems in decentralized systems with self-interested agents is to formulate the design problem as an optimization problem. The parameters of the design are the kinds of strategies that will be available to agents, specified for example by the bidding language in an auction, and the rules used to determine an outcome based on agent actions, for example rules to determine the winning bids in an auction.

Perhaps the most successful application of MD has been to the theory of auctions. In recent years auction theory has been applied to the design of a number of real-world markets [38].

The fundamental assumption made in MD is that participants in a mechanism will follow a game-theoretic equilibrium strategy. Simply stated, each agent is expected to select actions that maximize its expected utility, given the strategies of the other agents and the rules of the game, as specified by the mechanism. As such, MD makes a strong behavioral assumption about agents and about the information available to agents, and then selects the mechanism that will maximize some system-wide performance measure, such as seller revenue, with respect to the behavioral assumption.

As an example, suppose that a number of users are competing for access to the data staging capabilities on a data server located in Times Square. In MD the valuations of users for different amounts of disk space are private, and the problem is to design incentives, for example through the calculation of appropriate payments, to allocate disk space to maximize the total value across users. One cannot simply request that users report their valuations and then make an allocation based on reported valuations, because self-interested agents can be expected to overstate their values. Rather, a useful mechanism must design the rules of the game to provide incentives for agents to announce truthful information about their preferences for different outcomes.

### 1.2.1 Mechanism Design: Preliminaries

A mechanism defines a set of feasible *strategies*, that restrict the kinds of messages that agents can send to the mechanism. A mechanism also fixes a particular *outcome rule*, which selects an outcome based on agent strategies.<sup>1</sup> Game theoretic methods are used to analyze the properties of a mechanism, under the assumption that agents are rational and will follow expected-utility maximizing strategies in equilibrium, given beliefs about the strategies of other agents and knowledge of the rules of the mechanism.

Formally, a mechanism,  $\mathcal{M} = (S, x)$ , defines a strategy space,  $S$ , and an *outcome rule*,  $x : S^N \rightarrow \mathcal{O}$ . The outcome rule selects outcome  $x(s')$  in some abstract set  $\mathcal{O}$ , given strategy profile  $s' = (s_1(\theta_1), \dots, s_N(\theta_N))$ . A strategy defines an action for an agent in all possible states of the mechanism. For example, a strategy in an ascending-price auction defines when an agent will bid, and what price an agent will bid, for all possible states of the auction. The outcome rule in the auction takes these strategies, and then implements the choice (an allocation and payments) based on the strategies.

An agent,  $i \in \mathcal{I}$ , has a *type*,  $\theta_i \in \Theta$ . This captures all the *private* information about an agent that is relevant to its preferences for different outcomes. Formally, a type,  $\theta_i$ , defines an agent's utility,  $u_i(o, \theta_i)$ , for outcome  $o \in \mathcal{O}$ . A complete strategy description,  $s_i : \Theta_i \rightarrow \mathcal{S}$ , defines a strategy,  $s_i(\theta'_i)$ , for each possible

---

<sup>1</sup>A mechanism must be able to make a commitment to use these rules. Without this commitment ability the equilibrium of a mechanism can quickly unravel. For example, if an auctioneer in a second-price auction cannot commit to selling the item at the second-price than the auction looks more like a first-price auction [50].

type,  $\theta'_i \in \Theta_i$ . Given a strategy profile,  $s' = (s'_1, \dots, s'_N)$  and outcome rule  $x'$ , then agent  $i$ 's utility is  $u_i(x(s'), \theta_i)$ .

A mechanism is said to *implement* a social choice function,  $f : \Theta^N \rightarrow \mathcal{O}$ , if

$$x(s_1^*(\theta_1), \dots, s_N^*(\theta_N)) = f(\theta), \quad \forall \theta \in \Theta^N$$

where  $s^*$  denotes an equilibrium strategy profile, given an outcome rule,  $x$ , and strategy space,  $S$ . A number of different equilibrium solution concepts are typically considered, including *Bayesian-Nash* (BNE), *ex-post Nash*, and *Dominant Strategy*.<sup>2</sup> Dominant strategy implementations are particularly robust because they do not require agents to have correct beliefs about the types of other agents, or beliefs that other agents will play an equilibrium strategy. Ex post Nash equilibrium also have useful robustness properties.

Two important classes of mechanisms are *direct revelation* and *indirect revelation* mechanisms. A mechanism is a direct revelation mechanism (DRM) if the strategy space available to an agent is restricted to its space of types, i.e.  $S_i = \Theta_i$ . In words, an agent can only make a claim to the mechanism about its preferences. An indirect mechanism is any mechanism in which the strategy space is something other than the type space of an agent. Computational concerns aside, the *revelation principle* [21, 22] states that it is possible to restrict attention to DRMs, with no loss in implementation power. In particular, anything that can be implemented in some complex mechanism  $\mathcal{M}$  can also be implemented in a DRM. The intuition is that if mechanism  $\mathcal{M}$  implements SCF,  $f(\theta)$ , in equilibrium, then we can construct a DRM,  $\mathcal{M}'$ , that will implement the same SCF by simulating agent equilibrium strategies and mechanism  $\mathcal{M}$  within mechanism  $\mathcal{M}'$ . Moreover, the revelation principle states that it is sufficient to consider mechanisms in which agents will choose to announce truthful information about their preferences in equilibrium. These truthful and direct-revelation mechanisms are referred to as *incentive-compatible* mechanisms.

The revelation principle allows an analytic solution to the MD problem in a number of interesting design problems. Examples of solved MD problems include: the *bargaining problem* [42, 6] in which there is a single buyer and a single seller; the single-item *optimal auction* design problem [41] in which a seller wishes to maximize her expected revenue; and the *efficient mechanism design problem* in which the objective is to allocate resources to maximize the total value across agents.

---

<sup>2</sup>A strategy profile,  $s^*$ , is a Bayesian-Nash equilibrium if it maximizes the expected utility to every agent, given the strategies of the other agents and given beliefs about the distribution over agent types. Every agent is assumed to have probabilistic information about the types of the other agents. All agents have the same information, and this is common knowledge. A strategy profile,  $s^*$ , is an *ex post* Nash equilibrium, strategy if it is a utility-maximizing strategy for every agent given the strategies of the other agents, and *whatever* the types of the other agents. In a dominant strategy equilibrium, strategy  $s_i^*(\theta_i)$ , for every  $i$ , is a utility-maximizing strategy whatever the strategies and whatever the types of the other agents.

### 1.2.2 Example: Efficient Mechanism Design

It is illustrative to present the well known Vickrey-Clarke-Groves (VCG) mechanism [56, 7, 23]. The VCG mechanism is an incentive-compatible DRM for the *efficient allocation problem*. In this setting, the outcome space,  $\mathcal{O}$ , is defined in terms of a choice,  $k$ , from a discrete set of possible choices,  $\mathcal{K}$ , and payments,  $p = (p_1, \dots, p_N)$ , where  $p_i$  is the payment *by* the mechanism to each agent. Agents are assumed to have quasilinear utility functions, such that  $u_i(k, p_i) = v_i(k, \theta_i) + p_i$ , given choice  $k$ , payment  $p_i$ , and type  $\theta_i$ . In an auction setting, an agent's type,  $\theta_i$ , defines its valuation function. The system-wide objective is to implement a choice,  $k^* \in \mathcal{K}$ , that maximizes the total value across all agents. We can imagine that a choice,  $k$ , specifies an allocation of resources across agents. Given this, an efficient mechanism must always select the choice to maximize the total value.

By the revelation principle, we can focus on incentive-compatible DRMs. The outcome rule is defined in terms of an *allocation rule*,  $g : \Theta \rightarrow \mathcal{K}$ , and a *payment rule*,  $p : \Theta \rightarrow \mathbb{R}^N$ . Given reported types,  $\hat{\theta}$ , then choice  $g(\hat{\theta})$  is implemented and agent  $i$  makes payment  $p_i(\hat{\theta})$ .

The goal of efficiency, combined with incentive-compatibility (IC), pins down the allocation rule:

$$g_{\text{eff}}(\theta) = \arg \max_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} v_i(k, \theta_i) \quad (\text{EFF})$$

for all  $\theta \in \Theta$ . The remaining problem is to choose a payment rule that provides incentive-compatibility.

A particularly strong version of incentive-compatibility is that of *strategyproofness*, in which truth-revelation is a dominant strategy, i.e. utility-maximizing whatever the strategies or preferences of other agents. Formally, strategyproofness requires that the allocation and payment rules satisfy the following constraints:

$$v_i(g(\theta_i, \theta_{-i}), \theta_i) - p(\theta_i, \theta_{-i}) \geq v_i(g(\hat{\theta}_i, \theta_{-i}), \theta_i) - p(\hat{\theta}_i, \theta_{-i}), \quad \forall i, \forall \theta_i, \forall \hat{\theta}_i, \forall \theta_{-i} \quad (\text{SP})$$

Strategyproofness is a useful property because agents can play their equilibrium strategy without game-theoretic modeling or counterspeculation about other agents.

The Groves [23] mechanisms completely characterize the class of efficient and strategyproof mechanisms [22]. The payment rule in a Groves mechanism is defined as:

$$p_{\text{groves}, i}(\hat{\theta}) = \sum_{j \neq i} v_j(g_{\text{eff}}(\hat{\theta}), \hat{v}_j) - h_i(\hat{\theta}_{-i})$$

where  $h_i : \Theta_{-i} \rightarrow \mathbb{R}$  is an arbitrary function on the reported types of every agent except  $i$ , or simply a constant. The Groves payment rule internalizes the externality placed on the other agents in the system by the reported preferences of agent  $i$  and aligns each agent's incentives with the system-wide goal of allocative-efficiency.



To understand the strategyproofness of the Groves mechanisms, consider the utility of agent  $i$ ,  $u_i(\hat{\theta}_i)$ , from reporting type  $\hat{\theta}_i$ , given  $g_{\text{eff}}$  and  $p_{\text{groves}}$ , and fix the reported types,  $\theta_{-i}$ , of the other agents. Then,  $u_i(\hat{\theta}_i) = v_i(g_{\text{eff}}(\hat{\theta}_i, \theta_{-i}), \theta_i) + p_{\text{groves},i}(\hat{\theta}_i, \theta_{-i})$ , and substituting for  $p_{\text{groves}}$ , we have  $u_i(\hat{\theta}_i) = v_i(g_{\text{eff}}(\hat{\theta}_i, \theta_{-i}), \theta_i) + \sum_{j \neq i} v_j(g_{\text{eff}}(\hat{\theta}_i, \theta_{-i}), \theta_j) - h_i(\theta_{-i})$ . Truth revelation maximizes the sum of the first two terms by construction, and the final term is independent of the reported type. This holds for all reported types from other agents, and strategyproofness follows.

From within the class of Groves mechanisms, the Vickrey-Clarke-Groves (VCG) mechanism is especially important because it minimizes the expected total payments by the mechanism to the agents, across all incentive-compatible, efficient, and *individual-rational* (IR) mechanisms [33]. An IR mechanism is one in which participation is voluntary and agents can choose not to participate. In the VCG mechanism, the payment  $p_{\text{vcg},i}$ , is computed as:

$$p_{\text{vcg},i}(\hat{\theta}) = \sum_{j \neq i} v_j(g_{\text{eff}}(\hat{\theta}_{-i}), \hat{\theta}_j) - \sum_{j \neq i} v_j(g_{\text{eff}}(\hat{\theta}), \hat{\theta}_j)$$

where  $g_{\text{eff}}(\hat{\theta}_{-i})$  is the efficient allocation as computed with agent  $i$  removed from the system. For a single item allocation problem, this VCG mechanism reduces to the well-known Vickrey [56] auction, which is a second-price sealed-bid auction.

The VCG mechanism has received considerable attention within computer science and operations research in recent years, in particular in application to distributed optimization problems with self-interested agents (e.g. [43, 46]). Later, in Section 1.4, we will draw some connections between the payments in the VCG mechanism and the mathematics of collective intelligence.

### 1.2.3 Computational Considerations

Computational and informational considerations are largely missing from economic mechanism design. Simply stated, mechanism design assumes that *all equilibrium have the same cost*. In fact, a pair of otherwise equivalent mechanisms can have equilibrium solutions with vastly different computational properties.

Briefly, some computational considerations that can impact the choice of a mechanism in practice, include:

- The *computational cost*, both to agents to compute an equilibrium strategy and to the mechanism to compute the outcome based on agent strategies. Strategyproof mechanisms are one compelling class of mechanisms that do not place an unreasonable game-theoretic burden on participants. Approximations to strategyproof mechanisms that are intractable to implement have been suggested for some problems, where the goal is to retain strategyproofness but also simplify the problem facing the mechanism infrastructure (e.g. [35, 43]).

- The *informational cost* of computing an equilibrium. Direct revelation mechanisms require that agents submit complete and exact information about their types, to enable the mechanism to compute optimal outcomes for all possible reports from other agents. This can be unreasonable, for example when the preference elicitation problem involves collecting information from users, or solving hard optimization problems to evaluate different outcomes, of which there can be exponentially many. In comparison, agents can often compute equilibrium strategies in indirect mechanisms with approximate information about their own types [45, 49, 8]. For example, in an ascending-price auction an agent can bid with upper- and lower-bounds on its value for the item. A number of recent studies consider the design of mechanisms in settings with costly preference elicitation [25, 47, 9], and the communication complexity of mechanisms [24, 44].

The basic idea in all of the aforementioned approaches to computational MD is to reduce the complexity of the proposed solution until the equilibrium of the mechanism is computable.

A fundamentally different approach is to perform MD with respect to a model of the bounded-rationality of an agent. For example, to determine the optimal mechanism given a model of *satisficing* agent behavior, such as myopic best-response to prices, or simple reinforcement-learning behavior. Although some studies have considered the performance of different mechanisms for models of bounded-rational agents [49, 8, 48], there are currently no theories for the design of optimal mechanisms with respect to an explicit model of bounded-rational agent behavior. It is an intriguing challenge, to develop an analytically tractable, yet meaningful, model of agent bounded-rationality.

The mathematics of collective intelligence provides some useful insights into the challenging issues that agent bounded-rationality introduced into mechanism design. In particular, the methods of collective intelligence suggest a third approach, in which mechanisms are designed to explicitly support *convergence* towards equilibrium outcomes by *simple adaptive agents*. We pick up on this in more detail in Section 1.4.

#### 1.2.4 Dynamic Considerations

Another place where traditional MD breaks down is in its assumption that all agents are present in a system, and able to reveal their type information simultaneously within the mechanism. This assumption is often unreasonable in dynamic systems, for example in open network environments such as the Internet, where multi-period and asynchronous interactions across anonymous and shifting agent populations is a more reasonable model [17].

It is interesting to consider the *online mechanism design problem*, in which agents are continually entering and leaving a system and the goal is to make both truth-revelation of valuations and truthful announcements of arrival an equilibrium of the system (see Friedman & Parkes [18]). Online MD is interesting, for

example, in the context of *ad hoc* network formation across peer-to-peer WiFi networks in which services maintain robust *overlay* networks as nodes enter, leave, and fail.

Once an agent's arrival and departure time is introduced into its type, the revelation principle continues to apply and reduces the problem to the space of mechanisms in which agents truthfully announce their type, and therefore also their arrival time, in equilibrium. However, standard approaches, such as the family of Groves mechanisms, only provide strategyproof solutions in combination with optimal *online* algorithms that are able to make optimal sequential allocation decisions as agents arrive and announce their type. Relaxing to Bayesian-optimal online algorithms corresponds to Bayesian-Nash incentive-compatible mechanisms [18]. A simpler special case of the online MD problem occurs when it is reasonable to assume *truthful arrivals*, which can be justified if agents are myopic and react to the current state of the system as soon as they arrive. The online/truthful problem remains interesting because the sequential-decision aspect of the problem is retained.

This problem of online mechanism design with myopic agents is closely related to the problem of endogenous network and group formation in economics (e.g. [11, 26, 10, 28]). In this work, agents arrive into a system and choose an action to maximize their myopic utility.<sup>3</sup> The focus is on identifying which network structures can emerge from a sequence of myopic decisions by self-interested agents. A typical setting is one of *network formation*, in which agents choose how to add a new link into a network. Recently, there has also been some attention to a *stochastic* model of network formation, in which agents arrive by some random arrival process and then take a myopic decision (e.g. [57]). Dutta & Jackson [11] identify the mechanism design approach as an important future direction for the study of network formation, where payoff division rules are imposed by a designer in order to promote the emergence of useful networks.

Finally, there is an interesting comparison to be drawn between the online mechanism design problem and the recent literature on utility-based models to explain the existence and generation of complex networks with particular statistical properties (see e.g. [1]). On one hand, one can view the structure and statistics of complex networks from the perspective of solving a constrained global optimization problem [5]. On the other hand, recent models have provided local optimization-based models to explain the existence of complex networks [12]. It is interesting to ask whether there is an opportunity to leverage mechanism design in the positioning of appropriate local incentives to align local agent decisions with optimal system-wide performance, in order to encourage the emergence of complex networks with desirable topologies and statistical properties.

---

<sup>3</sup>A few studies which have considered an extensive form game model of the sequential decisions by a fixed set of agents, see for example Aumann & Myerson [3].

### 1.3 Collective Intelligence

In this section we review the mathematical theory of collectives (see Chapter ??Wolpert?? for an introduction to collective intelligence). We provide a simple treatment that will enable a useful comparison to be drawn with the mathematics of economic mechanism design. Although there are a number of important differences between collectives and the traditional setting of mechanism design, it will nevertheless be useful to draw parallels between mechanism design and collective intelligence.

The main distinction between collective intelligence (COIN) and MD is that the approach of collective intelligence is fundamentally an *indirect* paradigm. Rather than focusing on direct-revelation mechanisms in which agents reveal private information to the mechanism, that implements a particular outcome, the COIN approach implements an optimal system-wide outcome through the direct actions of agents.

Another distinction between COIN and MD is that in COIN it is assumed to be possible to *directly* set the utility functions of agents in order to guide their choices. In comparison, MD is more constrained because a designer can only indirectly influence agent payoffs via her choice of the payment and outcome rules.

Additional distinctions between COIN and MD include:

- **No private information.** The center in COIN is assumed to have enough information about local agent preferences to compute the system-wide value for a particular state.
- **Limited rationality.** The theory of COIN allows agents to have limited-rationality, including agents that follow simple best-response dynamics in response to an evolving multiagent system.

Finally, COIN has typically been implemented in *multiple period* problems, in which agents are able to adjust their strategies across time. In comparison, MD has traditionally been applied to static environments.<sup>4</sup>

The ability to adjust payoffs directly, coupled with the absence of private information, would render the COIN problem trivial from a MD perspective, because there are no incentive issues to solve. However, COIN focuses instead on *computational goals*, such as providing payoffs to enable a system of bounded-rational agents to quickly adjust to good system-wide outcomes. This focus in COIN nicely complements the obsession with incentives, but relative complacency about computational issues, within MD.

---

<sup>4</sup>One exception within mechanism design is the work on Repeated Implementation [30], in which the center aims to implement an outcome in the long-run as agent strategies converge towards a Nash equilibrium.

### 1.3.1 Collective Intelligence: Preliminaries

We define in this section a standard and simple COIN model. To maintain notational consistency with MD, our notation will necessarily depart from the notation in Chapter ??Wolpert??.

Let  $t \geq 1$  denote the time period, and  $\theta_i^t$  denote the *type* of agent  $i$ . The type in COIN is used analogously to the type in MD, and is intended to capture any information that is relevant to the behavior and preferences of an agent. Since agents in COIN may be bounded-rational, and not necessarily game-theoretic, the type in COIN captures information that relates to an agent's computational and belief state, in addition to its preferences. Furthermore, the type is indexed with time period,  $t$ , and allowed to change across periods to reflect changing preferences and changing computational state. Finally, in the standard COIN model the type can be directly controlled by the system designer.

An agent follows a *strategy*,  $s_i^t$ , which captures information about the actions taken by the agent in period  $t$ . Taken together, the strategy and the type of an agent, defines its *state*, which is denoted  $\xi_i^t = (s_i^t, \theta_i^t)$ . Notation,  $\xi_i = (\xi_i^1, \xi_i^2, \dots)$ , denotes the time sequence of states for agent  $i$ ,  $\xi^t = (\xi_1^t, \dots, \xi_N^t)$ , denotes the state of all agents in period  $t$ , and  $\xi = (\xi^1, \xi^2, \dots)$  denotes the total state across all periods. The individual utility,  $u_i(\theta_i^t, \xi) \geq 0$ , of agent  $i$ , is defined over the total state,  $\xi$ , of the system, and evaluated here with respect to its type at time  $t$ . Shorthand,  $u_i^t(\xi)$ , is used to denote the restriction to states up to and including period  $t$ .

The designer's goal is to maximize the system-wide utility across all states, denoted  $G(\xi)$ . Shorthand,  $G^t(\xi)$ , is used to denote the restriction to the social value for time periods up to and including  $t$ . The system-wide utility can depend on both the strategies and types of agents, and need not be the sum over the individual agent utilities, but can be quite general.

The central solution concept in COIN is that of a *factored system*. A system is factored if the individual utility of every agent is always aligned with the system-wide utility:

$$u_i^t(\xi) = G^t(\xi) - \Gamma_i(f_{-i}(\xi^1, \dots, \xi^t)) \quad (\text{FAC})$$

for all agents  $i$  and all time periods  $t$ . The function,  $f_{-i}(\xi^1, \dots, \xi^t)$ , takes only the components of states,  $\xi^1, \dots, \xi^t$ , that are *independent* of any state information related to agent  $i$ , and  $\Gamma_i$  is an arbitrary function over that state information. In words, a collective is factored if and only if a change at time  $t$  to the state of agent  $i$  alone, when propagated across time, will result in an increased utility to agent  $i$  if and only if it results in an increase to the system-wide utility.

The factoredness of a system is a stronger concept than the game-theoretic implementation concept in indirect MD. The main difference is that every agent should always want to select its own strategy to maximize the social value,  $G(\xi)$ , *whatever* the state of the system. In comparison, an incentive-compatible mechanism that implements a particular social-choice function must only ensure that an agent's incentives are aligned with the social good *when other agents also play an*

*equilibrium strategy*. In this aspect, the factoredness solution concept of COIN is reminiscent of the strategyproof solution concept of MD (although applied here to indirect mechanisms), in that it does not require on-equilibrium play of other agents.

We can readily verify that factoredness is sufficient to implement the choice that maximizes the system-wide utility,  $G(\xi)$ , in a dominant-strategy equilibrium. Let  $\xi^*$  denote the optimal social choice. Consider agent  $i$ . The states,  $\xi_i^*$ , represent a best-response choice because

$$\begin{aligned} \xi^* = \arg \max_{\xi} G(\xi) &\Rightarrow \xi_i^* = \arg \max_{\xi_i'} G(\xi_i', \xi_{-i}^*) - \Gamma_i(f_{-i}(\xi^*)) \\ &\Rightarrow \xi_i^* = \arg \max_{\xi_i'} u_i(\xi_i', \xi_i^*) \end{aligned}$$

However, we observe that full factoredness is a strong property, and is not necessary for a system of agents to maximize the value of  $G$  in equilibrium. As an example, suppose that each agent selects an action  $s_i \in [0, 1]$ , and that  $G(s_1, s_2) = f(s_1)f(s_2)$  where  $f(x)$  is a Gaussian function with mean 0.5 and variance 1. Now, suppose that the system is factored for agent 1, with  $u_1(s_1, s_2) = f(s_1)f(s_2)$ , but not factored for agent 2, with

$$u_2(s_1, s_2) = \begin{cases} f(s_2), & \text{if } s_1 \in [0.5 - \epsilon, 0.5 + \epsilon] \\ 1 - f(s_2), & \text{otherwise} \end{cases}$$

for some small,  $\epsilon > 0$ . Despite this failure of factoredness, the system is factored in the neighborhood of the optimal system-wide outcome of  $s^* = (0.5, 0.5)$ , and this is a stable outcome.

Factoredness is a useful property for a dynamic system because it implies that a rational agent will always play to maximize the social-value of the choice whatever the actions of other agents, and therefore gives useful robustness and stability properties in environments with bounded-rational, or faulty, agents.

### 1.3.2 Informative Local Utilities

The key challenge in COIN is to select factored local utilities to promote good convergence to a desired state. COIN agents are modeled as simple bounded-rational agents, rather than as traditional game-theoretic agents. In particular, the standard COIN model assumes that agents play a myopic best-response strategy as the state of a system evolves, given their current beliefs about payoffs for different strategies.

Given a particular system-wide utility,  $G(\xi)$ , the designer in COIN retains some flexibility in her choice of factored utility functions for individual agents. Specifically, the designer can choose the functions  $\Gamma_i$  and  $f_{-i}$  in (FAC). The theory of collective intelligence demonstrates that utility functions that are better able to isolate the effect of an individual agent's strategy on the system-wide utility are more *informative* for simple learning agents, and improve the rate of convergence to optimal system-wide agent strategies.

An example of a factoring choice with *poor* information properties is to set-up a *team game* [40], with

$$u_{\text{TG},i}^t(\xi) = G^t(\xi)$$

This is clearly factored, because every agent’s utility is exactly that of the over-all system. However, team game utilities are not very informative because the marginal effect of an agent’s own action on the system utility is likely to be masked by the effect of the actions of the other agents.

In developing a factored utility, it is important that the function,  $f_{-i}$ , leaves only components of states  $(\xi^1, \dots, \xi^t)$  that are *independent* of the states corresponding to agent  $i$ . It is not, in general, sufficient to remove only those states,  $(\xi_i^1, \dots, \xi_i^t)$ , that pertain directly to agent  $i$ , because an earlier strategy of agent  $i$  can affect the strategy and beliefs of another agent in a later round. Instead, it is also necessary to remove any states for other agents that are potentially affected by the state of agent  $i$  in some earlier round. This is referred to as the *effect set* of agent  $i$ , and denoted  $\text{eff}(i)$ . The independent states are denoted  $\xi \setminus \text{eff}(i)$ , with shorthand  $\xi_{-i}$ .

Another idea for a factored local utility is presented by the *aristocratic utility* (AU), in which:

$$u_{\text{AU},i}^t(\xi) = G^t(\xi) - \mathbb{E}_{\xi' \in \text{eff}(i)} [G^t(\xi_{-i}, \xi')] \quad (\text{AU})$$

The expectation is defined with respect to current beliefs over the probability of different states in the effect set of an agent.<sup>5</sup> This aristocratic utility is maximally-informative, with respect to a reasonable definition of the information provided by a reinforcement signal [58].

In the special case of a repeated single-stage game the effect-set of agent  $i$  is simply its own state, and the aristocratic utility simplifies to

$$u_{\text{AU},i}(\xi) = G(\xi) - \mathbb{E}_{s'_i \in S_i} [G(\xi_{-i}, s'_i)]$$

where the time index,  $t$ , is dropped, set  $S_i$  denotes the space of legal actions for agent  $i$ , and the expectation is taken with respect to the distribution of actions played by agent  $i$ .

Another idea for a factored local utility, is presented by the *wonderful life utility* (WLU). This can be a useful approximation to AU, in particular when AU is itself difficult to compute. In WLU the states in the effect set of agent  $i$  are replaced with a single “clamped” set of states:

$$u_{\text{WLU},i}^t(\xi) = G^t(\xi) - G^t(\xi_{-i}, \text{CL}_i) \quad (\text{WLU})$$

where  $\text{CL}_i$ , is the *clamping factor*. This clamping defines a fixed strategy for every agent in the effect set of agent  $i$  across all periods.

---

<sup>5</sup>It is important to notice in this construction that the dynamics of the combined state  $(\xi_{-i}, \xi^t)$  do not need to be consistent with the dynamical laws of the system. They are purely used as a counterfactual operator [54].

In the special case of an iterated single-stage game the wonderful life utility simplifies to:

$$u_{\text{WLU},i}(\xi) = G(\xi) - G(\xi_{-i}, \text{CL}_i)$$

where the time index,  $t$ , is dropped, and  $\text{CL}_i \in S_i$  is some fixed strategy for agent  $i$ . This is factored for any choice of clamping factor.

### 1.3.3 Back to VCG Mechanisms

Interestingly, the WLU factored local utility function brings us full circle, back to the Vickrey-Clarke-Groves mechanism that was introduced in Section 1.2 as a canonical example of a strategyproof solution in economic mechanism design.

Consider the special-case of a WLU with clamping factor,  $\text{CL}_i = \emptyset$ , to represent a null move by agent  $i$ . In this case an agent's utility for some strategy  $\xi$  is equal to the marginal contribution of its own strategy to the system-wide utility. Thus, WLU with a null clamping factor, implements the same equilibrium outcome as the VCG mechanism in the special case of a revelation game in which agent strategies define the revelation of type information and the system-wide utility is the sum of individual agent utilities.

Looking at COIN, both theoretical and experimental results suggest that it is better to use a WLU with a clamping factor that is equal to an agent's *expected* move. This provides a mean-field approximation to the AU, which itself has provably optimal informative properties. This suggests that the informativeness of the information provided by payments in an economic mechanism will be of interest in environments in which simple, but self-interested agents, are learning optimal strategies through best-response dynamics, and should be considered in selecting the appropriate mechanism from the class of Groves mechanisms. We pick up this theme in Section 1.4, where we introduce a COIN-inspired approach to MD for repeated games with simple adaptive agents.

### 1.3.4 Example: A Congestion Game

In this section, we present an example of COIN in the setting of a congestion game and review experimental results from Wolpert & Tumer [58]. Consider a simple variation on the *El Farol bar problem* [2]. Each player is deciding which night in the week to attend a bar, and if too few people attend the bar is boring but if too many people attend the bar is too crowded. This is a simple *coordination game*, in which the system-wide goal is to spread the attendance evenly across the nights.

Formally, there are  $N$  players and 7 nights, and every player wants to attend for the same number of nights,  $l$ . The value of  $l$ , which is selected from set  $\{1, \dots, 6\}$ , defines different variations of the game. In particular, as  $l$  increases, the bar will become more crowded. In each period,  $t$ , every player chooses a strategy,  $s_t^i$ . This defines a vector of 0's and 1's, e.g. (0110000), to indicate which nights the player will attend.



Given the joint strategies,  $s^t = (s_1^t, \dots, s_N^t)$ , then the system-wide utility for a particular week,  $t$ , is

$$G(s^t) = \sum_{k=1}^7 \phi(x_k(s^t))$$

where  $x_k(s^t)$  is the total attendance on night  $k$ . The congestion function,  $\phi(x) = x e^{-x/c}$  for some  $c \geq 0$ , and attendance  $x$ , captures the idea that the bar should not be too empty or too crowded on any single night. The function is maximized for an attendance of exactly  $c$ . The optimal system-wide outcome has players making a coordinated decision, with the same number of players attending on each night.

Notice that this problem has characteristics that are consistent with the COIN approach, but different from the MD approach. First, the system-wide utility function is known to the mechanism. Second, the protocol is an indirect mechanism in which the outcome is determined directly by the strategies of the players.

Experimental results [58] compare the performance of different WLUs, for different clamping factors. This is an iterated single-stage game, so the effect set for player  $i$  is simply its strategy in period  $t$  and the clamping factor defines a static strategy. Clamping factors  $\text{CL}_i = \vec{0}$ , and  $\text{CL}_i = \vec{1}$  and  $\text{CL}_i = \vec{1}/7$  are considered. Notice that it is not necessary that the clamping factors are legal moves. Let  $k_i(s_i)$  denote the nights that player  $i$  chooses to attend with strategy  $s_i$ . Then, WLU for clamping factor  $\text{CL}_i = (\text{CL}_{i1}, \dots, \text{CL}_{i7})$ , is defined as:

$$u_{\text{WLU},i}(s_i) = \sum_{k=1}^7 \phi(x_k(s)) - \left[ \sum_{k \notin k_i} \phi(x_k(s) + \text{CL}_{ik}) + \sum_{k \in k_i} \phi(x_k(s) - 1 + \text{CL}_{ik}) \right]$$

The mean-field approximation clamping factor,  $\text{CL}_i = \vec{1}/7$  (i.e. with each player assuming that the nights are well-balanced) is shown to outperform the other WLUs, with the  $\vec{0}$  clamp outperforming the  $\vec{1}$  clamp except when the  $\vec{1}$  clamp is a good approximation to the mean-field clamp. This can occur as congestion increases and the number of nights that a player wants to attend approaches 6. The experimental results also demonstrate that all *difference* methods, with  $u_i(\xi) = G(\xi) - \Gamma(f_{-i}(\xi))$ , are more effective than team game methods, with  $u_i(\xi) = G(\xi)$ .

## 1.4 Learnable Mechanism Design for Episodic Systems

One approach to the problem of mechanism design with bounded-rational agents is to consider agents with simple learning algorithms, that can adjust towards Nash equilibrium strategies [51, 17]. Indeed, there is an established literature that considers the ability of agents to *learn* to play equilibria in games [19, 20, 31, 53]. The

emphasis is on *simple* learners that can adjust, for example through myopic best-response, towards equilibrium strategies. A useful *learnable mechanism* would provide information, for example via price signals, to maximize the effectiveness with which individual agents can learn equilibrium strategies.

Given the focus on methods to address agents with limited-rationality in COIN, it is interesting to reinterpret the methods of COIN in terms of designing learnable mechanisms for episodic systems, such as iterated single-stage games. The methods of COIN suggest a new criteria for selecting a mechanism from within the Groves family of mechanisms, namely to provide informative signals to agents that are using simple learning methods to adjust towards equilibrium play.

### 1.4.1 The VCG-WLU Mechanism

Consider the following multi-period MD problem. Each period,  $t$ , is a single-stage game, that is repeated across periods. There is a set of choices,  $\mathcal{K}$ , to implement in each period, and the goal is to implement the efficient choice,  $k^*$ , which maximizes  $\sum_i v_i(k, \theta_i)$ , where  $v_i(k, \theta_i) \geq 0$  is the value of agent  $i$  for choice  $k$  and  $\theta_i$  is the type of agent  $i$ . Consider a direct-revelation mechanism, with strategy space  $S_i = \Theta_i$  for each agent, and choice rule,

$$g_{\text{eff}}(\hat{\theta}) = \max_{k \in \mathcal{K}} \sum_i (v_i(k, \hat{\theta}_i))$$

for types  $\hat{\theta} = (\hat{\theta}_1, \dots, \hat{\theta}_N)$ . Given this choice rule, the system-wide utility function from COIN is

$$G(\hat{\theta}) = \sum_i v_i(g_{\text{eff}}(\hat{\theta}), \theta_i)$$

Of course, in MD, there is uncertainty about this system-wide utility because types,  $\theta_i$ , are private to agents. Consider a Groves mechanism, with payment

$$p_{\text{groves},i}(\hat{\theta}) = \sum_{j \neq i} v_j(g_{\text{eff}}(\hat{\theta}), \hat{\theta}_j) - h_{-i}(\hat{\theta}_{-i})$$

to agent  $i$  given reported types  $\hat{\theta}$ , and some arbitrary function,  $h_{-i}$ , on the reported types of all agents except  $i$ .

These payments, combined with outcome rule,  $g_{\text{eff}}$ , *induce* individual utility functions on agents:

$$u_i(\hat{\theta}_i, \hat{\theta}_{-i}) = v_i(g_{\text{eff}}(\hat{\theta}), \theta_i) + \sum_{j \neq i} v_j(g_{\text{eff}}(\hat{\theta}), \hat{\theta}_j) - h_{-i}(\hat{\theta}_{-i})$$

These individual agent utility functions are not always factored. However, they are factored when other agents,  $j \neq i$ , follow equilibrium strategies and  $\hat{\theta}_{-i} = \theta_{-i}$ . This follows from the incentive-compatibility of Groves mechanisms:

$$\begin{aligned} u_i(\hat{\theta}_i, \hat{\theta}_{-i}) &= v_i(g_{\text{eff}}(\hat{\theta}_i, \theta_{-i}), \theta_i) + \sum_{j \neq i} v_j(g_{\text{eff}}(\hat{\theta}_i, \theta_{-i}), \theta_j) - h_{-i}(\theta_{-i}) \\ &= G(\hat{\theta}_i, \hat{\theta}_{-i}) - h_{-i}(\hat{\theta}_{-i}) \end{aligned}$$

Although the Groves payments do not make the agent utilities factored out of equilibrium, they are still sufficient to provide convergence towards equilibrium. For any strategy,  $\hat{\theta}_{-i}$ , from agents  $j \neq i$ , the best-response,  $s_{br,i}(\hat{\theta}_{-i})$ , of agent  $i$  is truth-revelation:

$$\begin{aligned} s_{br,i}(\hat{\theta}_{-i}) &= \arg \max_{\theta'_i \in \Theta_i} u_i(\theta'_i, \hat{\theta}_{-i}) \\ &= \arg \max_{\theta'_i \in \Theta_i} \left[ v_i(g_{\text{eff}}(\theta'_i, \hat{\theta}_{-i}), \theta_i) + \sum_{j \neq i} v_j(g_{\text{eff}}(\theta'_i, \hat{\theta}_{-i}), \hat{\theta}_j) - h_{-i}(\hat{\theta}_{-i}) \right] \\ &= \theta_i \end{aligned}$$

This property follows directly from the strong dominant-strategy truth-revelation properties of Groves mechanisms.

Economic MD suggests using additional properties, such as individual-rationality and revenue-maximizing to select the function  $h_{-i}$  and choose a particular Groves mechanism. As an example, the VCG mechanism, with  $h_{-i}(\hat{\theta}_{-i}) = \sum_{j \neq i} v_j(g_{\text{eff}}(\hat{\theta}_{-i}))$ , is popular because it maximizes the expected revenue across all individual-rational and efficient solutions [34].

In comparison, the theory of COIN suggests that a better choice than VCG payments for a mechanism in which agents are bounded-rational, and adjusting towards an equilibrium, is to set:

$$h_{-i}(\hat{\theta}_{-i}) = v_i(g_{\text{eff}}(\bar{\theta}_i, \hat{\theta}_{-i}), \bar{\theta}_i) + \sum_{j \neq i} v_j(g_{\text{eff}}(\bar{\theta}_i, \hat{\theta}_{-i}), \hat{\theta}_j) \quad (\text{VCG-WLU})$$

where  $\bar{\theta}_i$  is the *ex ante* average type of agent  $i$ . This is analogous to setting a WLU utility, in which the clamping parameter set to the average action of player  $i$ . We will call this hybrid the *VCG-WLU* mechanism.

This payment rule forfeits *ex post* individual-rationality, because the equilibrium utility to an agent is:

$$u_i(\theta) = \sum_i v_i(g_{\text{eff}}(\theta), \theta_i) - \left( v_i(g_{\text{eff}}(\bar{\theta}_i, \theta_{-i}), \bar{\theta}_i) + \sum_{j \neq i} v_j(g_{\text{eff}}(\bar{\theta}_i, \theta_{-i}), \theta_j) \right)$$

This equilibrium utility is negative, unless  $\theta_i \geq \bar{\theta}_i$ .

However, the following is sufficient to retain *ex ante* individual-rationality, such that the expected utility to agent  $i$  is non-negative:

$$\mathbb{E}_{\theta_i} \left[ \sum_j v_j(g_{\text{eff}}(\theta_i, \theta_{-i}), \theta_j) \right] \geq v_i(g_{\text{eff}}(\bar{\theta}_i, \theta_{-i}), \bar{\theta}_i) + \sum_{j \neq i} v_j(g_{\text{eff}}(\bar{\theta}_i, \theta_{-i}), \theta_j)$$

for every  $\theta_{-i}$ . This requirement is explored in the following example.

*Example.*

Consider a Vickrey auction for a single item, in which agent the type,  $\theta_i \in [0, 1]$  specifies an agent's value for the item. Then, *ex ante* IR requires:

$$\mathbb{E}_{\theta_i} [\max(\theta_i, \theta_{-i})] \geq \max(\bar{\theta}_i, \theta_{-i}), \quad \forall \theta_{-i}$$

where  $\bar{\theta}_i = 0.5$ . Let  $x(\theta_{-i}) = \max_{j \neq i} \theta_j$ , then we require:

$$\mathbb{E}_{\theta_i} [\max(\theta_i, x)] \geq \max(\bar{\theta}_i, x), \quad \forall x$$

By case analysis on  $x$ , this holds trivially when  $x \geq 0.5$ , and when  $x < 0.5$  we have:

$$\mathbb{E}_{\theta_i} [\max(\theta_i, x)] = x + \int_{v=x}^1 v - x \, dv = x + 1/2 - x + x^2/2 \geq 1/2 = \max(\bar{\theta}_i, x)$$

Thus, the payments suggested by the WLU clamping  $\bar{\theta}_i$  satisfy *ex ante* IR in this simple example.

In the following sections we present experimental results to compare the effectiveness of the VCG-WLU mechanism with the VCG mechanism in an auction problem with simple adaptive agents, and also for a variant of the congestion game that was introduced in Section 1.3.4.

*1.4.2 Example: Auction Game*

Consider a simple allocation problem, in which there is a single item to allocate  $N$  agents each with type,  $\theta_i$ , uniformly distributed between 0 and 1. The type of an agent defines its value for the item. Let  $x_i = 1$  if agent  $i$  receives the item, and  $x_i = 0$  otherwise, with  $\sum_i x_i \leq 1$  for all feasible allocations. The utility of agent  $i$  for outcome  $x$  and payment  $p_i$  from the mechanism is

$$u_i(x, p_i, \theta_i) = \begin{cases} \theta_i + p_i & , \text{ if } x_i = 1 \\ p_i & , \text{ otherwise} \end{cases}$$

The social choice function is to maximize the allocative-efficiency of the outcome, with

$$G(x, \theta) = \sum_i x_i \theta_i$$

for feasible allocation  $x$ . A Groves mechanism for this single-item allocation problem first asks each agent to report its type, and then implements the outcome  $x(\hat{\theta})$  that maximizes  $G(x, \hat{\theta})$  such that  $\sum_i x_i \leq 1, x_i \in \{0, 1\}$ . Agent  $i$  receives payment

$$p_{\text{groves}, i}(\hat{\theta}) = \sum_{j \neq i} v_j(x(\hat{\theta}), \hat{\theta}_j) - h_{-i}(\hat{\theta}_{-i})$$

for some arbitrary function,  $h_{-i}$ , defined on the reported types  $\theta_{-i}$  for agents  $j \neq i$ . We consider the following choices, to complete the payment definitions:

$$h_{-i}(\hat{\theta}_{-i}) = \max_{j \neq i} \hat{\theta}_j \quad (\text{VCG})$$

$$h_{-i}(\hat{\theta}_{-i}) = 0 \quad (\text{Groves-TG})$$

$$h_{-i}(\hat{\theta}_{-i}) = \max(\bar{\theta}, \max_{j \neq i} \hat{\theta}_j) \quad (\text{VCG-WLU})$$

where  $\bar{\theta}$  is the *average* type, in this example equal to 0.5.

Putting this together, and assuming that agent 1 announces the highest type and agent 2 announces the second-highest type, then the payment rules simplify to:

$$p_{\text{vcg},i} = \begin{cases} -\hat{\theta}_2 & , \text{if } i = 1 \\ 0 & , \text{otherwise} \end{cases}$$

$$p_{\text{TG},i} = \begin{cases} 0 & , \text{if } i = 1 \\ \hat{\theta}_1 & , \text{otherwise} \end{cases}$$

$$p_{\text{WLU},i} = \begin{cases} -\max(\bar{\theta}, \hat{\theta}_2) & , \text{if } i = 1 \\ \hat{\theta}_1 - \max(\bar{\theta}, \hat{\theta}_1) & , \text{otherwise} \end{cases}$$

In our experiments we adopt the approach in Wolpert & Tumer [58], and use a simple reinforcement learning algorithm for individual agents. Each agent,  $i$ , considers a linear strategy  $\hat{\theta}_i = k\theta_i$ , for some  $k \in \{0.6, 0.7, 0.8, 0.9, 1, 1.1\}$ , and maintains a 6-dimensional vector to estimate the utility it receives for each strategy. Let  $\pi_i^t(k)$  denote the utility estimate by agent  $i$  in period  $t$  for strategy  $k$ . At the beginning of each period each agent picks the strategy to announce at random, using a Boltzmann distribution defined over the six components of the estimated utility vector. In particular, the probability of playing strategy  $k$  is proportional to  $e^{(\pi_i^t(k)/temp^t)}$ . The parameter,  $temp^t$ , controls the amount of exploration in period  $t$ .

At the end of each period the mechanism reports the allocation,  $x$ , and the payment  $p_i$  to each agent and the agent computes its utility for the outcome and updates its estimates. For example, if agent  $i$  plays strategy  $k$  in period  $t$  and receives utility  $\pi$ , then its new estimate for strategy  $k$  is computed as  $\pi_i^{t+1}(k) = (1 - \lambda^t)\pi_i^t(k) + \lambda^t\pi$ . The parameter,  $\lambda^t$ , controls the amount of adjustment in period  $t$ .<sup>6</sup> An uninformative prior sets each component of the estimated utility vector to the average utility received during an initial training period in which all agents choose random strategies. Finally, the parameters,  $temp^t$ , and  $\lambda^t$ , are adjusted across periods according to decay factors  $0 < \alpha < 1$  and  $0 < \beta < 1$ , with  $temp^{t+1} = \alpha temp^t$  and  $\lambda^{t+1} = \beta \lambda^t$ .

In all experiments the initial temperature,  $temp^1$ , is equated to the average utility during the initial training period, and the decay rate,  $\beta$ , for the learning-rate

---

<sup>6</sup>The use of this weighted average over an exponential decay function reflects the fact that the environment is non-stationary.

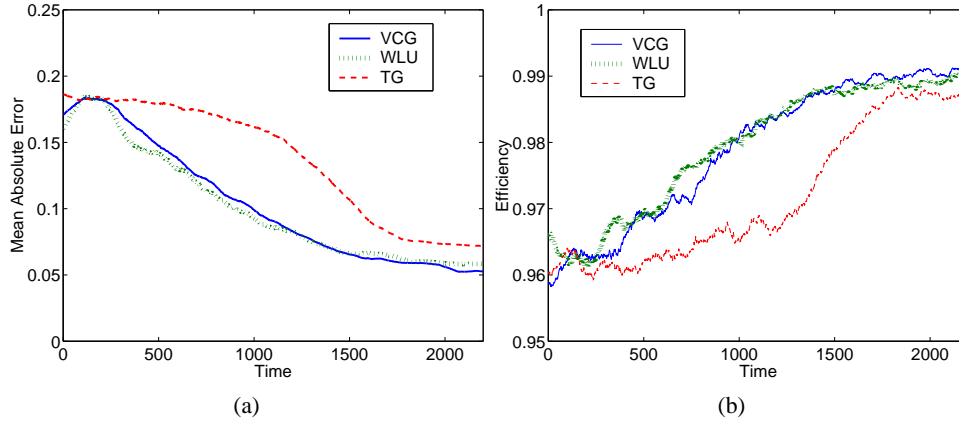


FIGURE 1.1. The Auction Problem with 5 Agents, comparing the VCG, VCG-WLU, and TG payment rules. Learning begins after 200 periods. (a) Moving average of the mean absolute error between agent strategies and truth-revelation. (b) Moving average of the efficiency of the auction.

is set to fix the learning-rate at the end of the final period to 0.001. We experimentally optimized the choice of the initial learning-rate,  $\lambda^1$ , and the temperature decay rate,  $\alpha$ , for each different choice of design with a logarithmic search in parameter space to select parameters that maximized the *average* efficiency and minimized the *average* distance to the equilibrium strategy, across all periods.

#### Experimental Results.

In the first set of experiments we considered an auction with 5 agents, and set the initial training period to 200 periods and the learning period to 2000 periods. All results are averaged over 40 runs. In the second set of experiments we considered an auction with 3 agents, having noticed that the payments in VCG approximate those in VCG-WLU, and are different only when the average value of an agent is greater than the second-highest reported value. For these experiments the initial training period was 200, and the learning period was 1000. As with 5 agents, the results are averaged over 40 runs.<sup>7</sup>

Figures 1.1 and 1.2 compare the performance of the mechanisms in the 5 agent and 3 agent settings. First, we plot the convergence of agent strategy to the equilibrium strategy, which is truth-revelation in all auctions. Given  $N$  agents, we compute the *mean absolute error* in each period, as  $1/N \sum_i |k_i - k^*|$ , where  $k_i$  is the strategy selected by agent  $i$  and  $k^* = 1$  is the equilibrium strategy. Second, we plot the *efficiency* of the outcome in each period, which is the ratio of the value of the item to the agent that receives the item to the value of the agent with the

<sup>7</sup>With 5 agents, the parameters  $(\alpha, \lambda^1)$  were set to  $(0.9973, 0.05)$ ,  $(0.9978, 0.2)$  and  $(0.9969, 0.1)$  in auctions VCG, VCG-WLU, and TG respectively. With 3 agents, the parameters  $(\alpha, \lambda^1)$  were set to  $(0.9944, 0.2)$ ,  $(0.9955, 0.1)$  and  $(0.9932, 0.1)$ .

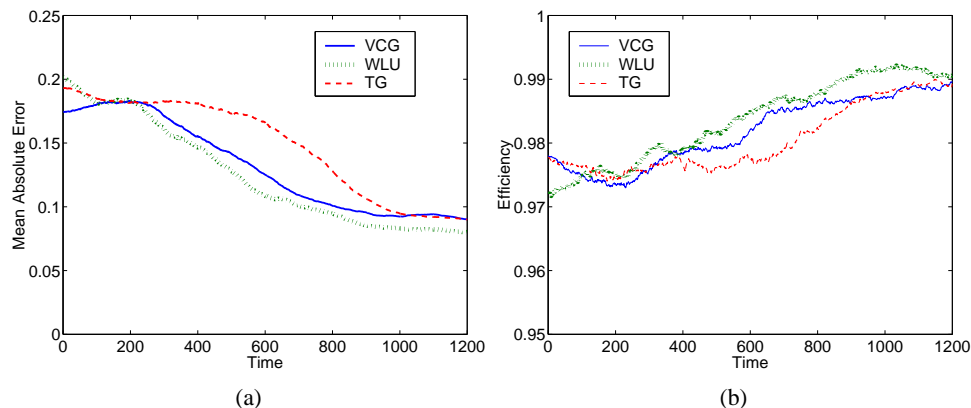


FIGURE 1.2. The Auction Problem with 3 Agents, comparing the VCG, VCG-WLU, and TG payment rules. Learning begins after 200 periods. (a) Moving average of the mean absolute error between agent strategies and truth-revelation. (b) Moving average of the efficiency of the auction.

maximum value. In both cases, we find it convenient to plot the *moving-averages* (with a window size of 100). This smoothes out random fluctuations from period-to-period due to Boltzmann learning. Also, the first 200 periods represent the initialization period, in which agents select random strategies.

As expected, the performance of the VCG and VCG-WLU auctions dominates that of the TG auction with simple learning agents. Moreover, the VCG-WLU method appears to slightly outperform the VCG method during the early learning periods with 5 agents, both in terms of the accuracy of agent strategies and the average efficiency. In addition, the effect of VCG-WLU is more striking with 3 agents, with the COIN-inspired VCG payments providing a clear performance advantage over the regular VCG payments.

### 1.4.3 Example: Congestion Game

We illustrate the VCG-WLU mechanism on a direct-revelation variation of the congestion game. Consider a simple variation on Arthur's El Farol bar problem. There are  $N$  players, one bar, and one night. The problem is interesting because the *type* of each player defines its *tolerance* for congestion, and is private to each player. Moreover, it is not certain which players will attend the bar.

On any night, the problem is to decide which players attend the bar. Let  $y_i \in \{0, 1\}$  denote whether or not player  $i$  attends the bar (with  $y_i = 1$  for attendance). Then, given a solution  $y = (y_1, \dots, y_N)$ , player  $i$  with type  $\theta_i$  has value:

$$v_i(y, \theta_i) = \begin{cases} x e^{x/\theta_i} & , \text{if } y_i = 1 \\ 0 & , \text{otherwise} \end{cases}$$

where  $x = \sum_i y_i$ .

Given types,  $\theta = (\theta_1, \dots, \theta_N)$ , the system-wide goal is to implement a solution in which the attendance,  $y^*$ , maximizes the total value:

$$\max_y \sum_i v_i(y, \theta_i) \quad (1.1)$$

The problem is to strike a compromise between the different preferences of players for the level of crowdedness in the bar. For example, if enough players prefer a crowded bar than it can be beneficial from the system-wide perspective to make players attend even if that makes it more crowded than desirable for other players.

The difference from the earlier congestion game (in Section 1.3.4) is that the mechanism *itself* implements a particular attendance profile, and must elicit information about player types to implement an optimal solution.

The Groves mechanism for this problem first asks players to report their type,  $\hat{\theta}_i$ , and then implements the outcome,  $y(\hat{\theta})$ , that solves (1.1) given the reported types. Player  $i$  receives payment

$$p_{\text{groves},i}(\hat{\theta}) = \sum_{j \neq i} v_j(y(\hat{\theta}), \hat{\theta}_j) - h_{-i}(\hat{\theta}_{-i})$$

where, as before,  $h_{-i}$  is some arbitrary function on the reported types of the other players. We again consider the particular variations that implement the VCG, VCG-WLU, and TG payments. The VCG payments require that the mechanism computes an alternative solution without each player in attendance, while the VCG-WLU payments require that the mechanism computes an alternative solution with the type of each player replaced, in turn, with the average player type.

#### Experimental Results.

First, we consider the congestion game with 8 players, and possible types  $\theta_i$  selected uniformly at random from  $\{1, 2, 3, 4, 5, 6\}$ . Second, we consider this congestion game with 4 players, and possible types  $\{1, 2, 3\}$ . We use the same Boltzmann learning method as in the auction example. Each player considers a linear strategy  $\hat{\theta}_i = k\theta_i$ , for some  $k \in \{0.6, 0.7, 0.8, 0.9, 1, 1.1\}$  and maintains an estimate of the utility for each strategy. This induces a Boltzmann distribution to define a probability with which the player selects a particular strategy. Utility estimates are again initially set to an uninformative prior, during an initial training period in which all players follow random strategies. We experimentally optimized the choice of initial learning-rate and temperature decay rate for each different choice of design.<sup>8</sup>

In the 8-player variation, we set the initial training period to 200 and the learning period to 1000. In the 4-player variation, we set the initial training period to

---

<sup>8</sup>For 8 players, the parameters  $(\alpha, \lambda^1)$  were set to  $(0.9947, 0.2)$ ,  $(0.9978, 0.05)$ , and  $(0.9932, 0.05)$  in the VCG, VCG-WLU, and TG mechanisms respectively. For 4 players, the parameters were set to  $(0.9865, 0.1)$ ,  $(0.9991, 0.02)$ , and  $(0.9861, 0.2)$ .



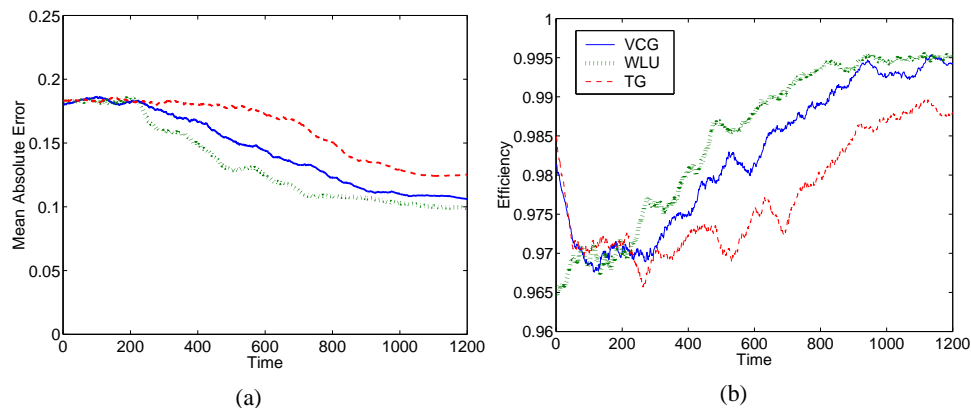


FIGURE 1.3. The Congestion Problem with 8 Players, comparing the VCG, VCG-WLU, and TG payment rules. Learning begins after 200 periods. (a) Moving average of the mean absolute error between player strategies and truth-revelation. (b) Moving average of the efficiency of the outcome.

100 and the learning period to 500. The 8-player results are averaged over 20 runs and the 4-player results are averaged over 60 runs. Again, we were interested to compare the 8-player and the 4-player variations, because we expected the VCG-WLU to VCG comparison to be more noticeable with less players.

Figures 1.3 and 1.4 compare the performance of the mechanisms in the 8 player and 4 player settings. We plot the mean absolute error between the player strategies and truthful strategies in each period, and the efficiency of the outcome in each period, which in this problem is measured as the ratio between the total value of the implemented outcome to the total value of the optimal outcome. We plot the moving averages (window size of 50), to smooth out the random fluctuations due to Boltzmann learning.

Just as in the auction example, the performance of the TG mechanism with simple learning agents, in terms of both the speed of convergence towards an equilibrium strategy and the overall efficiency across periods, is dominated by both the VCG and the VCG-WLU mechanisms. Most striking in this congestion game example, is that the performance of VCG-WLU mechanism itself dominates that of the VCG mechanism, both for the 8 player and the 4 player variations. This provides some experimental justification for a collective-intelligence inspired approach to mechanism design in the presence of bounded-rational agents.

## 1.5 Summary: Towards Learnable Mechanism Design

The integration of methods from COIN into methods in mechanism design can be viewed as a first step towards learnable mechanism design. Learnable mechanism design is a natural direction to take mechanism design in complex decentralized

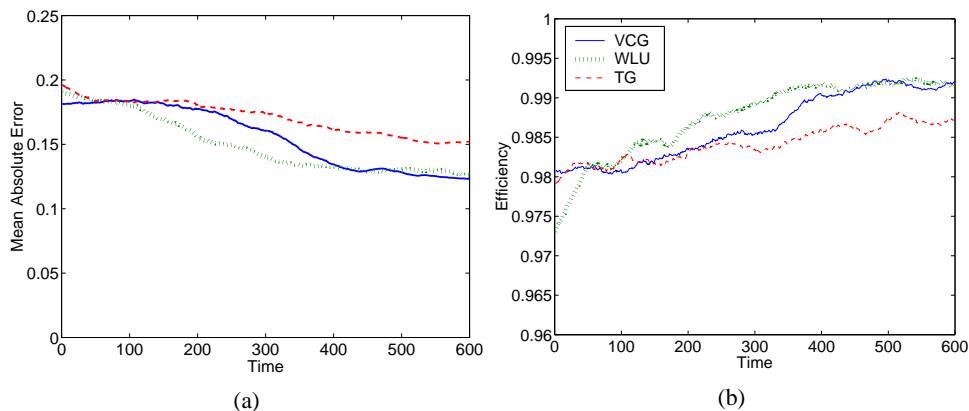


FIGURE 1.4. The Congestion Problem with 4 Players, comparing the VCG, VCG-WLU, and TG payment rules. Learning begins after 100 periods. (a) Moving average of the mean absolute error between player strategies and truth-revelation. (b) Moving average of the efficiency of the outcome.

settings. Classic mechanism design formulates an explicit *normative* model of the equilibrium behavior of an agent, and selects mechanism rules that are optimal with respect to that model. In particular, the Myerson program formulates the problem as a constrained optimization problem, in which one selects an outcome rule that maximizes a set of desiderata subject to incentive-compatibility constraints. In contrast, the idea presented in learnable mechanism design is to design a mechanism that is optimal with respect to a *behavioral* model of bounded-rational agents, and in particular to worry about the performance along the path towards equilibrium as well as in equilibrium itself.

The methods in this chapter assume simple Boltzmann learners, and adopt the idea of informative utilities from COIN to select an instance of the Groves family of mechanisms in which payments to agents are especially informative in the feedback they provide about the effect of an agent's choice of strategy on her individual utility. But, this opens up many interesting questions. In particular, there has been an explosion of research into algorithms to compute Nash equilibrium (and special-classes of equilibrium such as correlated equilibrium) in game-theoretic settings (e.g. [32, 55, 29]), and also to identify tractable special-cases of the equilibrium-computation problem (e.g. [36]). Many of the algorithms have a best-response/learning flavor (e.g. [16, 14, 15, 39, 31]). A very natural question arises: can we design mechanisms that induce games that have computable equilibrium, or equilibrium that are readily computed by simple learning agents?

In addition to identifying classes of mechanisms that induce game-theoretic situations with good computational properties, we can also consider whether there is a role for *automated mechanism design* in which the rules of mechanisms are automatically adjusted *online* to provide robustness against unmodeled properties of a real system, such as those due to the limited-rationality of participants.

## Acknowledgments

The author has benefited from discussions with Laszlo Gulyas, Eric Friedman, Chaki Ng, Margo Seltzer, and David Wolpert. This work was supported in part by grant NCC 2-5505 from Nasa Ames.

### 1.6 REFERENCES

- [1] R Albert and A-L Barabasi. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, 2002.
- [2] W Brian Arthur. Inductive reasoning and bounded rationality. *AEA Papers and Proceedings*, 84(2):406–411, 1994.
- [3] R Aumann and R Myerson. Endogeneous formation of links between players and coalitions. In Al Roth, editor, *The Shapley Value*, pages 175–191. Cambridge University Press, 1988.
- [4] Craig Boutilier. A pomdp formulation of preference elicitation problems. In *Proc. 18th National Conference on Artificial Intelligence (AAAI-02)*, July 2002. to appear.
- [5] J M Carlson and J Doyle. Highly Optimized Tolerance: A mechanism for Power Laws in designed systems. *Physics Review E*, 60(2):1412–1427, 1999.
- [6] K Chatterjee and W Samuelson. Bargaining under incomplete information. *Operations Research*, 31:835–851, 1983.
- [7] E H Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.
- [8] Olivier Compte and Philippe Jehiel. Auctions and information acquisition: Sealed-bid or Dynamic Formats? Technical report, CERAS and UCL, 2002.
- [9] Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions. In *Proc. 3rd ACM Conf. on Electronic Commerce (EC-01)*. ACM Press, New York, 2001.
- [10] B Dutta and S Mutuswami. Stable networks. *Journal of Economic Theory*, 76:322–344, 1997.
- [11] Bhaskar Dutta and Matthew O Jackson. On the formation of networks and groups. In Bhaskar Dutta and Matthew O Jackson, editors, *Models of the Strategic Formation of Networks and Groups*. Heidelberg: Springer Verlag, 2002.

- [12] Alex Fabrikant, Elias Koutsoupias, and Christos H. Papadimitriou. Heuristically optimized trade-offs: A new paradigm for Power Laws in the Internet. In *Proc. STOC'02*, 2002.
- [13] Joan Feigenbaum and Scott Shenker. Distributed Algorithmic Mechanism Design: Recent Results and Future Directions. In *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 1–13, 2002.
- [14] Dean P Foster and Rakesh Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 1997.
- [15] Dean P Foster and Rakesh Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, pages 7–36, 1999.
- [16] Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.
- [17] E Friedman and S Shenker. Learning and implementation in the internet. Preprint. Available at: <http://www.aciri.org/shenker/decent.ps>, 1997.
- [18] Eric Friedman and David C. Parkes. Pricing WiFi at Starbucks– Issues in online mechanism design. In *Fourth ACM Conf. on Electronic Commerce (EC'03)*, 2003. Shorter version. Extender version available at <http://www.eecs.harvard.edu/parkes/pubs/online.pdf>.
- [19] Drew Fudenberg and David M Kreps. Learning mixed equilibria. *Games and Economic Behavior*, 5:320–367, 1993.
- [20] Drew Fudenberg and David Levine. *Theory of Learning in Games*. MIT Press, 1997. forthcoming.
- [21] Alan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41:587–602, 1973.
- [22] Jerry Green and Jean-Jacques Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45:427–438, 1977.
- [23] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.
- [24] R Holzman, N Kfir-Dahav, D Monderer, and M Tennenholtz. Bundling equilibrium in combinatorial auctions. Technical report, Technion and Stanford, 2001.
- [25] B. Hudson and T. Sandholm. Effectiveness of preference elicitation in combinatorial auctions. In *Agent Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, volume 2531 of *Lecture Notes in Artificial Intelligence*. 2002.

- [26] M O Jackson and A Wolinsky. A strategic model of social and economic networks. *Journal of Economic Theory*, 71:44–74, 1996.
- [27] Matthew O. Jackson. Mechanism theory. In *The Encyclopedia of Life Support Systems*. EOLSS Publishers, 2000.
- [28] Matthew O Jackson. The stability and efficiency of economic and social networks. In Murat Sertel, editor, *Advances of Economic Design*. Springer-Verlag, 2001. forthcoming.
- [29] S. Kakade, M. Kearns, J. Langford, and L. Ortiz. Correlated equilibria in graphical games. In *Proc. 4th ACM Conf. on Electronic Commerce*, 2003.
- [30] Ehud Kalai and John O Ledyard. Repeated implementation. *Journal of Economic Theory*, 83(2):308–317, 1998.
- [31] Ehud Kalai and Ehud Lehrer. Rational learning leads to Nash equilibrium. *Econometrica*, 61(5):1019–1045, 1993.
- [32] Michael Kearns, Michael L Littman, and Satinder Singh. Graphical models for game theory. In *Proc. of Conf. on Uncertainty in Artificial Intelligence*, pages 253–260, 2001.
- [33] Vijay Krishna. *Auction Theory*. Academic Press, 2002.
- [34] Vijay Krishna and Motty Perry. Efficient mechanism design. Technical report, Pennsylvania State University, 1998. Available at: <http://econ.la.psu.edu/~vkrishna/vcg18.ps>.
- [35] Daniel Lehmann, Liadan O’Callaghan, and Yoav Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. In *Proc. 1st ACM Conf. on Electronic Commerce (EC-99)*, pages 96–102, 1999.
- [36] M Littman, M Kearns, and S Singh. An efficient exact algorithm for singly connected graphical games. In *Neural Information Processing Systems*, 2002.
- [37] Andreu Mas-Colell, Michael D Whinston, and Jerry R Green. *Microeconomic Theory*. Oxford University Press, 1995.
- [38] Paul Milgrom. *Putting Auction Theory to Work*. MIT Press, 2002.
- [39] M.Littman and P.Stone. A polynomial-time nash equilibrium algorithm for repeated games. In *Proc. 4th ACM Conf. on Electronic Commerce*, 2003.
- [40] Dov Monderer and L S Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.
- [41] Robert B Myerson. Optimal auction design. *Mathematics of Operation Research*, 6:58–73, 1981.

- [42] Robert B Myerson and Mark A Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 28:265–281, 1983.
- [43] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.
- [44] Noam Nisan and I Segal. The communication complexity of efficient allocation problems. Technical report, Hebrew University and Stanford University, 2002.
- [45] David C Parkes. Optimal auction design for agents with hard valuation problems. In *Proc. IJCAI-99 Workshop on Agent Mediated Electronic Commerce*, pages 206–219, July 1999. Stockholm.
- [46] David C Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, May 2001. <http://www.cis.upenn.edu/~dparkes/diss.html>.
- [47] David C. Parkes. Price-based information certificates for minimal-revelation combinatorial auctions. In *Agent Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, volume 2531 of *Lecture Notes in Artificial Intelligence*. 2002.
- [48] David C Parkes, Jayant R Kalagnanam, and Marta Eso. Vickrey-based surplus distribution in combinatorial exchanges. In *Proc. 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001.
- [49] David C Parkes and Lyle H Ungar. Iterative combinatorial auctions: Theory and practice. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, pages 74–81, July 2000.
- [50] R Preston McAfee and John McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, June 1987.
- [51] S Shenker. Making greed work in networks: A game-theoretic analysis of switch service disciplines. In *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 47–57, 1994.
- [52] Jeff Shneidman and David C. parkes. Rationality and self-interest in peer to peer networks. In *2nd Int. Workshop on Peer-to-Peer Systems (IPTPS'03)*, 2003.
- [53] Satinder Singh, Michael Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proc. UAI'00*, 2000.
- [54] Kagan Tumer and David H Wolpert. Collective Intelligence and Braess' Paradox. In *aaai00*, pages 104–109, 2000.

- [55] D Vickrey and D Koller. Multi-agent algorithms for solving graphical games. In *Proc. Nat. Conf. on Artificial Intelligence (AAAI'02)*, 2002.
- [56] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.
- [57] Alison Watts. A dynamic model of network formation. *Games and Economic Behavior*, 34:331–341, 2001.
- [58] David H Wolpert and Kagan Tumer. Optimal payoff functions for members of collectives. In *Advances in Complex Systems*. 2001.