

Title Here

Edited by
authors here

Contents

1 Computational Mechanism Design	<i>page 4</i>
---	---------------

1

COMPUTATIONAL MECHANISM DESIGN

Abstract

Computational mechanism design brings together the concern in microeconomics with decision making in the context of distributed private information and self-interest and the concern in computer science with computational and communication complexity. In constructing mechanisms, with application to the design of electronic markets and to protocols for automated negotiation, many new issues arise in resolving tensions between incentive, computation and communication constraints.

1.1 Introduction

Mechanism design (MD) is a sub-field of microeconomics and game theory which considers how to implement good system-wide solutions to problems that involve multiple self-interested agents, each with private information about preferences and capabilities. In recent years mechanism design has found many applications within computer science and operations research; e.g., in electronic market design [FGHK02, HKP04], distributed planning [HG00, BGT03], and in solving many distributed combinatorial optimization problems [dVV03, CSS06]. MD is becoming increasingly relevant in distributed systems for commerce, computation, and information.

Mechanisms are protocols for decision making in multi-agent systems with self-interested agents that have private information which, collectively, determines the appropriate decision. It is often useful to conceptualize mechanism design as “inverse game theory.” One is starting with a model of the beliefs and private information of agents, and a normative model that

asserts that agents will play a Nash equilibrium of game, and designing the “rules of the game” such that the equilibrium behavior of agents will generate outcomes with desirable properties.

In computational MD (CMD) much of the focus has been on the design of mechanisms that are *truthful*. Truthfulness can be thought of as a statement about non-manipulability (at least in the absence of collusion): no agent can do better for itself than by truthfully revealing private information in interacting with the mechanism. Truthfulness extends to indirect mechanisms, where mechanisms are designed such that agents follow “straightforward” or “intended” strategies in equilibrium. Common computational goals in CMD include making sure that the computation required by the protocol in determining the outcome can be efficiently implemented, and also minimizing the amount of information revelation from agents to the mechanism.

The need to simultaneously consider both computational and incentive issues arises in a number of practical settings. On eBay, where search engines, automated proxy agents, reputation mechanisms and ascending price auctions combine to form an electronic marketplace. At search engines such as Google, where automated proxy agents and machine learning techniques to predict click-through rates combine to determine the adverts that are co-located with search results. In expressive procurement auctions (e.g. CombineNet, Emptoris), where suppliers can use volume discounts and express capacity constraints and buyers can include business rules to influence winner determination (see Bichler et al. [CSS06, chapter23]).

Airline authorities such as the FAA in the U.S. have considered using combinatorial auctions for the allocation of takeoff and landing slots. As with other mechanisms for real world problems, determining the outcome of a combinatorial auction can be a hard computational problem, and expressive and concise languages are important, in this case to allow airlines to express values for different schedules. Markets have been deployed on sensor networks and computational test-beds to arbitrate resource allocation amongst competing users with conflicting needs [CBA⁺05]. Auctions are proposed as methods to coordinate multiagent planning where a joint plan must be formed in order to best complete shared tasks in a distributed environment [HG00, BGT03].

Table 1.1 provides a high-level comparison between the classic focus in computer science and the classic focus in economics, in terms of the model that is adopted for agents and the main concerns addressed (following Feigenbaum [FS].) CMD brings together these concerns and resolves tensions between incentive, computation and communication constraints.

In recent years, CMD has significantly broadened the scope of mecha-

Traditional Computer Science	Microeconomics
Agents are cooperative, sometimes adversarial	Agents are self-interested
Main concerns are computation and communication costs	Main concern is incentives

Table 1.1. *Drawing an Analogy between Computer Science and Economics*

nism design. For instance, new attention has been given to the problem of *distributed computation*, in which the agents (as computational devices) are used to perform some of the computation in making a decision and determining payments. *Preference elicitation* has emerged as a significant challenge in applying mechanisms to resource allocation in complex environments. Mechanisms have been proposed for *dynamic environments*, in which a mechanism must make a sequence of decisions and the agent population changes with time.

The goal of this chapter is to provide a broad, relatively self-contained, introduction to mechanism design, and follow this with an introduction to some of the problems studied in computational mechanism design. Section 1.2 introduces the basic model of mechanism design, including the most important game-theoretic solution concepts and some of the central possibility and impossibility results. Section 1.3 focuses on the computational complexity of centralized mechanisms, introduces the agenda of *algorithmic mechanism design*, and develops general characterizations for *truthful* mechanisms. Section 1.4 considers the problem of preference elicitation, and presents ascending price auctions and methods from learning theory in the context of combinatorial auctions. Section 1.5 introduces the challenges of *distributed implementation*, in which part of the computation in determining the outcome of a mechanism is performed by the agents. Section 1.6 extends mechanism design to dynamic environments, drawing connections to work on *online* algorithms and Markov Decision Processes. We conclude in Section 1.7.

1.1.1 Related Work

See Fudenberg and Tirole [FT91] and Osborne and Rubinstein [OR94] for useful introductions to game theory. See McAfee and McMillan [MM96], Klemperer [Kle00], and Krishna [Kri02] for introductions to auction theory.

Milgrom [Mil04] provides a more advanced treatment. Jackson [Jac03] provides an accessible survey of mechanism design, Mas-Colell et al. [MCWG95] a text book treatment, and Dasgupta et al. [DHM79] a comprehensive, technical survey.

The computational mechanism design topics covered in this chapter are necessarily restricted in scope and biased in selection. Readers are encouraged to consult the books on *Combinatorial auctions* [CSS06], and *Algorithmic Game Theory* [NRTV07], as well as the Proceedings of the *ACM Conference on Electronic Commerce* and the *International Conference on Autonomous Agents and Multiagent Systems* for a more complete view of work in the area.¹ Related work also appears in the main theoretical computer science conferences.

For a sampling of papers in the artificial intelligence community, consider the important early work of Ephrati and Rosenschein [ER91] and Rosenschein and Zlotkin [RZ94]. More recent papers include those by Sandholm [San96], Monderer and Tennenholtz [MT99], Porter et al. [PRST02], and Conitzer and Sandholm [CS02b, CS02a]. Numerous papers consider more specialized topics, for instance combinatorial and sequential auctions [Nis00, YSM04, LS04b, PU00, HKP04, e.g.]. For work in the theoretical computer science community, consider the papers of Nisan and Ronen [NR00, NR01], Lehmann et al. [LOS02] and Feigenbaum et al. [FKSS01].

1.2 Preliminaries

The decision to be made by a mechanism is formalized as the choice of some alternative from a set of alternatives $A = \{a, b, \dots\}$ and agents $N = \{1, 2, \dots\}$ with $|N| = n$. Agent i has private information (its *type*) $\theta_i \in \Theta_i$, and a value $v_i(a; \theta_i) \in \mathbb{R}$ for alternative $a \in A$. Often times we will just write $v_i(a)$. Agents are assumed to have *quasilinear utility*,

$$u_i(a, p) = v_i(a; \theta_i) - p, \quad (1.1)$$

for alternative a at price p .² We will restrict attention to private value models so that an agent's value for an alternative depends only on its own type.³

¹The graduate class, "Topics at the Interface between Computer Science and Economics" at Harvard University <http://www.eecs.harvard.edu/~parkes/cs286r> is also a good place to find papers and other information.

²Quasilinearity implies no budget constraints and risk neutrality, with agents indifferent between a payment equal to the expected value of a lottery and the lottery itself. See Borgs et al. [BCI⁺05] for a recent treatment of budget-constraints and other non-linearities in the utility function.

³Krishna [Kri02] provides an accessible treatment of mechanism design in interdependent value domains. Interdependent value models can be important, for instance when allocating resources

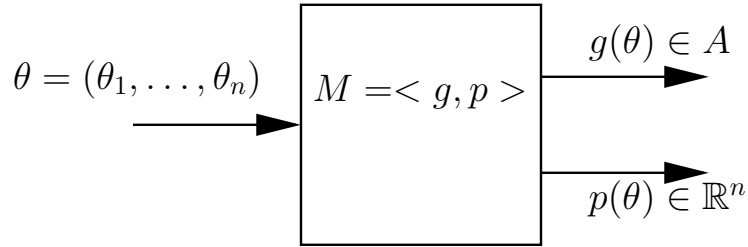


Fig. 1.1. The components of a direct-revelation mechanism.

The primary goal usually adopted in mechanism design can be expressed in terms of a *social choice function*, which defines an alternative to be selected for each possible private type vector $\theta = (\theta_1, \dots, \theta_n)$. For instance, one might be interested in implementing the value-maximizing alternative, i.e. $f(\theta) = \arg \max_{a \in A} \sum_i v_i(a; \theta_i)$. Revenue-maximization is another common goal adopted in mechanism design, especially in auction settings. In an allocation problem the alternatives represent allocations, and the transfers represent payments to the auctioneer.

Example 1.1 In an auction for a single-item, the alternatives define the possible allocations, i.e. which agent gets the item. Assuming that agent 1 has *value* $v_1(a; \theta_1) = 10$ for the alternative a in which it wins, then its utility for the alternative in which it is allocated the item at price p is $u_1(a, p) = v_1(a; \theta_1) - p = 10 - p$, and the agent has positive utility as long as $p < 10$.

1.2.1 Direct Revelation Mechanisms

We start by introducing the simple but important class of *direct revelation* mechanisms. A direct revelation mechanism (DRM), $M = \langle g, p \rangle$, is defined in terms of an outcome rule $g : \Theta \rightarrow A$ and a payment rule $p : \Theta \rightarrow \mathbb{R}^n$, where $\Theta = (\Theta_1 \times \dots \times \Theta_n)$ denotes the *joint* type space.⁴ A mechanism takes reports θ from agents and selects alternative $g(\theta)$ (the “outcome”) and payment $p_i(\theta)$ for each agent. (See Figure 1.1). Because agents are self-interested the reports need not be truthful. A mechanism M defines a situation of strategic interdependence, and thus a non-cooperative game.

in a competitive marketplace or when a better estimate of the value of an alternative can be constructed with information about the types of other agents.

⁴The outcome rule and payment rule can also be randomized. Randomization is useful in achieving competitive approximation guarantees in combinatorial auctions [DNS06] and auctions for digital goods [GH03]. We adopt deterministic rules here for ease of presentation.

The fundamental concept of agent choice in game theory is expressed as a *strategy*. A strategy defines the action an agent will select in all possible states of the world. For example, if the mechanism is an ascending price auction then a strategy defines the bid an agent will submit in response to all possible prices.

The basic model of agent rationality in game theory is that of an expected-utility maximizer. An agent will select a strategy that maximizes its expected utility, given its preferences over alternatives, beliefs about the strategies and types of other agents, and the structure of the game. The central solution concept is that of a *Nash equilibrium*, which states that in equilibrium every agent will select a utility-maximizing strategy given the strategy of every other agent. Thus, game theory allows a mechanism designer to reason about the alternative that will be implemented by a mechanism in equilibrium.

Mechanism design defines games of incomplete information because agents are typically modeled as having uncertainty about the types of other agents.⁵ When necessary, agents can be modeled as having beliefs about the types of other agents. In a DRM, strategies take a particularly simple form. A *strategy* $s_i : \Theta_i \rightarrow \Theta_i$, defines a reported type, $s_i(\theta_i)$, for every possible type of an agent. All agents simultaneously and privately make a claim about their type to the mechanism.

The most important solution concepts for DRMs are those of *dominant strategy equilibrium* and *Bayesian-Nash equilibrium*. These both characterize particular kinds of Nash equilibrium of the incomplete information game induced by a mechanism.

In defining these concepts we adopt the following standard short-hand notation. Let $s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ and $\theta_{-i} = (\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_n)$ denote the strategy profile and type profile without agent i . Given this, let $u_i(s_i, s_{-i}; \theta_i, \theta_{-i}) = v_i(g(s_i(\theta_i), s_{-i}(\theta_{-i})); \theta_i) - p_i(s_i(\theta_i), s_{-i}(\theta_{-i}))$, denote the utility to agent i in the game defined by mechanism $M = \langle g, p \rangle$ when agents adopt strategies (s_i, s_{-i}) and have types (θ_i, θ_{-i}) .

Definition 1.2 Strategy profile s^* is a *dominant strategy equilibrium* (DSE) in mechanism $M = \langle g, p \rangle$ when:

$$u_i(s_i^*, s_{-i}; \theta_i, \theta_{-i}) \geq u_i(s'_i, s_{-i}; \theta_i, \theta_{-i}), \quad \forall i, \forall \theta_i, \forall s'_i \neq s_i^*, \forall \theta_{-i}, \forall s_{-i} \quad (1.2)$$

In words, strategy s_i^* is a dominant strategy for agent i if the agent max-

⁵In contrast, the study of *Nash implementation theory* (as opposed to *mechanism design theory*) considers agents that are informed about each others' types and a center (or designer) that is uninformed. See Jackson [Jac01] for a useful survey of the literature on implementation theory.

imizes its utility with this strategy for all types θ_i , whatever the strategies (and types) of other agents.

Example 1.3 In a sealed-bid second-price (Vickrey auction), a single item is sold to the highest bidder for the second-highest price. Let type θ_i denote the value of agent i for the item. Then, bidding strategy:

$$s_i(\theta_i) = \theta_i \quad (1.3)$$

is a dominant strategy for agent i . To see this, consider some arbitrary θ_{-i} and s'_{-i} and fix θ_i . Agent i 's utility for report $s_i(\theta_i) = \hat{\theta}_i$ is:

$$u_i(s_i, s'_{-i}; \theta_i, \theta_{-i}) = \begin{cases} \theta_i - \max_{j \neq i} s'_j(\theta_j) & , \text{ if } \hat{\theta}_i > \max_{j \neq i} s'_j(\theta_j) \\ 0 & \text{ otherwise} \end{cases}$$

Agent i maximizes its utility by reporting $\hat{\theta}_i > \max_{j \neq i} s'_j(\theta_j)$ (so that it wins) if and only if $\theta_i > \max_{j \neq i} s'_j(\theta_j)$, which it achieves with strategy $s_i(\theta_i) = \hat{\theta}_i = \theta_i$. This is a dominant bidding strategy because it holds for any s'_i , any θ_{-i} and any θ_i .

Recognize that the fundamental reason for truthfulness in the Vickrey auction is that every agent faces a price that is independent of its report (i.e. $\max_{j \neq i} s'_j(\theta_j)$) and wins the item at that price if its reported value is greater than the price. We will revisit this idea of agent-independent prices in the general MD setting in Section 1.3.3.

Sometimes a mechanism with more desirable properties can be constructed by relaxing the solution concept to a Bayes-Nash equilibrium. Assume that types are distributed according to probability distribution function $Pr(\theta)$.

Definition 1.4 Strategy profile s^* is a *Bayes-Nash equilibrium* (BNE) in mechanism $M = \langle g, p \rangle$ when:

$$\mathbb{E}_{\theta_{-i}} [u_i(s_i^*, s_{-i}^*; \theta_i, \theta_{-i})] \geq \mathbb{E}_{\theta_{-i}} [u_i(s'_i, s_{-i}^*; \theta_i, \theta_{-i})], \quad \forall i, \forall \theta_i, \forall s'_i \neq s_i^*, \quad (1.4)$$

where the expectation is taken with respect to the conditional distribution θ_{-i} sampled from conditional distribution, $Pr(\theta_{-i}|\theta_i)$, given type θ_i .

In a BNE, every agent is assumed to share a common prior about the distribution of agent types. Moreover, both this prior belief and the rationality of agents must be common knowledge. In equilibrium, each agent plays a strategy that maximizes its expected utility given the distribution over types of other agents.

Example 1.5 In a sealed-bid first-price auction, a single item is sold to the highest bidder for its bid price. Let type θ_i denote the value of agent i for the item. Suppose that values are i.i.d. θ_i sampled uniformly on $[0,1]$. Then, bidding strategy

$$s_i(\theta_i) = \left(\frac{n-1}{n}\right)\theta_i$$

is the unique symmetric BNE.⁶

Definition 1.6 Mechanism $M = \langle g, p \rangle$ implements social choice function $f : \Theta \rightarrow A$ in DSE (or BNE) if $f(\theta) = g(s^*(\theta))$ for some DSE (or BNE) strategy profile s^* , for all types θ .

Mechanisms that implement a social choice function in a DSE are more robust than those which implement a social choice function only in a BNE because no assumptions are made about the knowledge that agents have about the types or rationality of other agents. Mechanisms with DSE are also preferred computationally because they free agents of the need to reason about the strategies of other agents.

An important subclass of DRMs are the *incentive-compatible* DRMs. In an incentive-compatible (IC) mechanism it is an equilibrium for every agent to report its true type.

Definition 1.7 Mechanism $M = \langle g, p \rangle$ is *dominant-strategy incentive-compatible* (DSIC) for type space Θ if strategy $s^*(\theta_i) = \theta_i$ is a dominant strategy equilibrium.

DSIC mechanisms are often called **strategy-proof** or **truthful** because they are non-manipulable in the sense that an agent can do no better than simply reporting its true type.

Definition 1.8 Mechanism $M = \langle g, p \rangle$ is *Bayes-Nash incentive-compatible* (BNIC) for type space Θ if strategy $s^*(\theta_i) = \theta_i$ is a Bayes-Nash equilibrium.

The social choice function implemented by an IC mechanism is defined by the outcome rule g ; an IC mechanism $M = \langle g, p \rangle$ is said to be a *dominant-strategy (or Bayes-Nash) implementation of social choice function g* .

By the *revelation principle* [Gib73] it is without loss of generality to focus on the space of incentive-compatible mechanisms. The revelation principle states that *any mechanism M' can be transformed into an equivalent IC*

⁶See the Appendix in Klemperer [Kle00] for a proof of this result via the revenue equivalence theorem and Krishna [Kri02] for a proof from first principles.

and DRM mechanism M that implements the same social choice function. See Mas-Colell et al. [MCWG95] for a text book treatment. The revelation principle holds for both dominant-strategy and Bayes-Nash equilibrium.⁷

Note that the revelation principle does not say that truth-revelation is “easy” to achieve, but simply that if some mechanism solves a problem in equilibrium then it can also be solved in a truth-revealing equilibrium of another mechanism. It is achieving a desired outcome in an equilibrium that is difficult, not making that equilibrium a truth-revealing equilibrium.

The revelation principle is useful in focusing goals and delineating what is and is not possible in MD (and thus also delineating impossibility results of relevance in computational MD). For instance, if there is no BNIC implementation of a social choice function with a set of desired properties then *no* mechanism (however complex) can succeed in implementing a social choice function with these properties.

Note, though, that the revelation principle puts computational considerations to one side. The revelation principle should not be construed as stating that direct mechanisms are the only mechanisms of practical relevance. On the contrary, indirect mechanisms can often enable more efficient preference elicitation and also distribute computation to agents.⁸ Moreover, the possibility results of MD ignore computational constraints and although possible from the perspective of incentive constraints, the computational complexity of a problem may also preclude implementation. We will consider the computational requirements placed on the center in Section 1.3 and the computational requirements placed on agents, for instance through the cost of preference elicitation, in Section 1.4.

1.2.2 Indirect Mechanisms

Indirect mechanisms define games with a more complicated information structure than direct mechanisms. Indirect mechanisms, which include ascending-price auctions and distributed mechanisms, are of great interest in many computational domains.

Indirect mechanisms differ from DRMs in two ways. First, the message

⁷The intuition behind the revelation principle is a reduction argument, and goes as follows. Mechanism M will simulate the entire system (the equilibrium bidding strategies s^* of agents and the outcome rule) of mechanism M' , given reports $\hat{\theta}$ from agents. Thus, if strategy s^* is an equilibrium in mechanism M' then an agent should report its true type $\hat{\theta}_i = \theta_i$ in mechanism M so that the mechanism simulates its correct equilibrium strategy. For a BNE, this simulation argument requires that the mechanism designer has access to the distribution over agent types.

⁸Conitzer and Sandholm [CS04] also provide an interesting construction that shows that computational complexity can be used to overcome some impossibility results, by shifting the complexity to agents.

space in an indirect mechanism does not correspond with reports about an agent's type. For instance, in an ascending-price auction the message space may instead allow an agent to report a bundle of items that maximizes its utility at the current prices, which provides indirect and partial information about an agent's type. Second, agents can typically send multiple messages while participating in a mechanism and can condition the messages that they send on information provided by the mechanism.

A *strategy* $s_i(\theta_i) \in \Sigma_i$ in an indirect mechanism defines the message(s) that an agent will send to the mechanism for all types θ_i and all possible *information states*. An indirect mechanism, $M = \langle \Sigma, g, p \rangle$, defines a space of feasible joint strategies $\Sigma = \Sigma_1 \times \dots \times \Sigma_n$ and an outcome rule, $g : \Sigma \rightarrow A$, and payment rule $p : \Sigma \rightarrow \mathbb{R}^n$. An information state delineates a possible state of the indirect mechanism, and a fully specified strategy should define a message to send for all possible information states.

Example 1.9 Consider a single-item ascending-price auction, in which type θ_i denotes the value of agent i for the item. In each round an agent can *bid* or *stop*. Once an agent stops it cannot bid in a later round. The price increases by some bid increment $\epsilon > 0$ while two or more agents bid. The winner is the last agent still bidding, and pays the final price. The information state p^t in round t defines the current price $p^t \geq 0$. A strategy defines the bid, $s_i(p, \theta_i)$, that an agent will place in every state p , and for every type θ_i . The straightforward bidding strategy,

$$s_i^*(p, \theta_i) = \begin{cases} \textit{bid} & , \text{ if } p \leq \theta_i \\ \textit{stop} & , \text{ otherwise} \end{cases}$$

is a DSE. All strategies are characterized by a threshold value, $\hat{\theta}_i$, such that the agent will *stop* for prices above this value. (This is because the auction constrains an agent to *stop* for all subsequent rounds to its first *stop*.) Fix threshold values $\hat{\theta}_{-i}$. The auction is now strategically equivalent for agent i to a second-price auction in which the highest bid from another agent is $\epsilon \lceil \frac{\max_{j \neq i} \hat{\theta}_j}{\epsilon} \rceil$, and s_i^* is agent i 's dominant strategy. Note that it is without loss of generality to fix $\hat{\theta}_{-i}$ because the threshold values selected by other agents are conditionally independent of agent i 's strategy, when conditioned on the case that agent i wins.

A more typical solution concept adopted in the analysis of indirect mech-

anisms is that of an *ex post* Nash equilibrium.⁹ This is a concept of intermediate strength, in between that of DSE and BNE.

Definition 1.10 Strategy profile s^* is an *ex post Nash equilibrium* (ex post NE) in mechanism $M = \langle \Sigma, g, p \rangle$ when:

$$u_i(s_i^*, s_{-i}^*; \theta_i, \theta_{-i}) \geq u_i(s_i', s_{-i}^*; \theta_i, \theta_{-i}), \quad \forall i, \forall \theta_i, \forall s_i' \neq s_i^*, \forall \theta_{-i} \quad (1.5)$$

In words, strategy s^* is an ex post NE if no agent can improve its utility by deviating, whatever the type of other agents, as long as the other agents are rational and play the equilibrium strategy. It is instructive to compare the definition of ex post Nash equilibrium with the definition of BNE (Eq. 1.4); an ex post NE is also a BNE, it is a BNE for any distribution on types. In DRMs, an ex post NE is equivalent to a DSE,¹⁰ but this need not be the case in indirect mechanisms. This is illustrated in the following example.

Example 1.11 Consider a single-item ascending-price auction with jump bids. Again, type θ_i denotes agent i 's value for the item. Bids are associated with a bid price. In each round, t , the auctioneer announces an ask price, p^t , which is $\epsilon > 0$ above the highest bid received so far from an agent. Any agent can bid in any round, as long as the bid is at some price at or above p^t . The provisional winner is the agent with the current highest bid (breaking ties at random). The auction terminates when no agent bids at the current price, and the item is then sold to the provisional winner at its final bid price. The information state (p^t, x^t) defines the current ask price p^t and $x^t \in \{1, \dots, n\}$ to indicate the provisional winner. A straightforward bidding strategy is:

$$s_i^*(p, x, \theta_i) = \begin{cases} p & , \text{ if } p \leq \theta_i \text{ and } x \neq i \\ \text{no bid} & , \text{ otherwise} \end{cases}$$

This is an ex post NE but not a DSE. To see that it is an ex post NE, fix straightforward strategies s_{-i}^* by agents other than i . Each agent's strategy is completely characterized by a threshold value, that of its own value for the item. The analysis then follows essentially as in the previous example. On the other hand, straightforward bidding is not a DSE. To see this suppose there are two agents and agent 1's value is 20, agent 2's value is 15, the bid

⁹Ex post IC was discussed as "uniform incentive compatibility" by Holmström and Myerson [HM83]. See also Jehiel and Moldovanu [JM01] and Bergemann and Morris [BM07].

¹⁰This equivalence is true in private value but not interdependent value models where an agent's best-response depends on whether other agents will report their true types and thus allow correct conditioning of value on the types of other agents. See Krishna [Kri02].

increment $\epsilon = 1$, and agent 2 follows the following bidding strategy:

$$s_2(p, x, \theta_2) = \begin{cases} p & , \text{ if } p \leq \theta_2 \text{ and } x \neq 2 \text{ and } p \neq 10 \\ 10000 & , \text{ if } p = 10 \\ \text{no bid} & , \text{ otherwise} \end{cases}$$

This is not a rational bidding strategy for agent 2, but nevertheless a feasible strategy given the rules of the auction. If agent 1 bids straightforwardly the price will reach 10, triggering agent 2's "crazy" bid of 10000 and agent 1 will lose. If agent 1 bids a jump bid of 16 from the start then it will win for 16. Thus, this response to the specific "crazy" strategy is better than the straightforward strategy.

This example illustrates the role of *ex post* NE in the design of indirect mechanisms: the presence of multiple information states allows one agent to condition its messages ("reports") on the messages of another agent, thus leading to richer strategic interactions.

The following limitation should be kept in mind in designing indirect mechanisms with *ex post* NE:

Theorem 1.12 *Any social choice function $f(\theta)$ that is implementable in an *ex post* NE of an indirect mechanism is implementable in the dominant strategy equilibrium of a truthful DRM.*

This is an immediate consequence of the arguments that underlie the revelation principle.¹¹ Thus, only dominant-strategy implementable social choice functions can be implemented in the *ex post* Nash equilibrium of an indirect mechanism.

Before continuing, we make some brief comments about Bayes-Nash equilibrium in the context of indirect mechanisms. Great care is required in this analysis because one must allow for the possibility that agents can usefully update their beliefs about the types of other agents as they interact with the mechanism and observe information states. The most convenient solution concept is a refinement called *perfect Bayesian-Nash equilibrium* in which agents use Bayes rule to update their belief states along the equilibrium path, and are required to follow an equilibrium strategy from all information states. In practice, this gets difficult because one must also define belief updates off the equilibrium path when probability zero events occur. See

¹¹In simulating the *ex post* NE of the indirect mechanism M' when constructing the corresponding IC and DRM mechanism M one "locks down" the strategies of all agents and thus ensures that all (simulated) agents will follow the *ex post* NE strategy of the indirect mechanism, albeit for some (perhaps untruthful) reported type.

Osborne and Rubinstein [OR94] and Fudenberg and Tirole [FT91] for a detailed discussion.

1.2.3 Possibility Results

Given the framework of MD one can ask *what properties of social choice functions can be implemented in the equilibrium of a mechanism, given the constraints implied by agents' private information and self-interest*. MD provides a number of interesting possibility and impossibility results. These are important to understand in the context of computational MD, which will layer on top additional constraints, i.e. those related to computational and communication considerations.¹²

Here are some possible desiderata for the social choice functions and payments implemented in the equilibrium of a mechanism $M = \langle g, p \rangle$:

- (**EFF**) The outcome is (ex post) efficient, i.e. $g(s^*(\theta)) \in \arg \max_{a \in A} \sum_i v_i(a; \theta_i)$, for all θ and equilibrium s^* .
- (**WBB**) The payments are (ex post) weak budget balanced, i.e. $\sum_i p_i(s^*(\theta)) \geq 0$ for all θ and equilibrium s^* .
- (**IR**) The outcome and payments satisfy (ex post) individual rationality, i.e. $v_i(g(s^*(\theta)); \theta_i) - p_i(s^*(\theta)) \geq 0$ for all i , all θ , and equilibrium s^* .

Efficiency (EFF) is termed *allocative efficiency* when applied in a resource allocation domain. Budget balance (WBB) can be relaxed to *ex ante* WBB when it is only required to hold in expectation given a distribution on agent types, i.e. $\mathbb{E}_\theta [\sum_i p_i(s^*(\theta))] \geq 0$, and can be strengthened to *strong* budget balance (BB) when payments must exactly balance to zero, i.e. $\sum_i p_i(s^*(\theta)) = 0$.

Individual-rationality (IR) corresponds to a *participation* constraint because it asserts that all agents receive non-negative utility from participating in the mechanism. IR can be relaxed to *interim* IR, meaning non-negative expected utility for any type θ_i given the distribution across other types, i.e. $\mathbb{E}_{\theta_{-i}} [v_i(g(s_i^*(\theta_i), s_{-i}^*(\theta_{-i})); \theta_i) - p_i(s_i^*(\theta_i), s_{-i}^*(\theta_{-i}))] \geq 0$, and can be further relaxed to *ex ante* IR, where the expectation is also taken with respect to an agent's own type, i.e. $\mathbb{E}_{\theta_i} \mathbb{E}_{\theta_{-i}} [v_i(g(s_i^*(\theta_i), s_{-i}^*(\theta_{-i})); \theta_i) - p_i(s_i^*(\theta_i), s_{-i}^*(\theta_{-i}))] \geq 0$. The appropriate variation of IR depends on the commitment power of the mechanism.

¹²Computational complexity can sometimes be used to reverse negative results by designing protocols in which desirable strategies are computable (or given) but undesirable strategies are hard to compute. See Conitzer and Sandholm [CS04] and Sanghvi and Parkes [SP04a] for a discussion.

Name	Utility	Solution concept	Possible	Valuation environment
Groves	QL	DSE	EFF & (IR <i>or</i> WBB)	general
VCG	QL	DSE	EFF, IR & WBB	no positive externalities
d'AGVA	QL	BNE	EFF, BB & <i>ex ante</i> IR	general
Myerson	QL	BNE (DSE)	OPT, WBB & IR	single-item (symmetric priors)

Table 1.2. *Mechanism Design Possibility Results.*

However, as a rule of thumb, *ex ante* IR is hard to justify and *ex post* IR is the usual standard that is adopted.¹³

An additional desiderata for a mechanism in some environments is to maximize the expected utility of a particular agent, most commonly formulated as that of maximizing the expected revenue of the seller in an auction:

(OPT) Mechanism M is revenue optimal if the expected revenue, $E_{\theta}[\sum_i p_i(s^*(\theta))]$, is maximal across all possible mechanisms, where s^* is an equilibrium strategy.

Table 1.2 summarizes four of the most well known possibility results.¹⁴ The possibility results are delineated by the form of the agent utility function (i.e. quasilinear (QL) in all of these cases), the equilibrium solution concept, and the “valuation environment,” which describes the assumptions made about the valuation functions of agents. General value environments allow arbitrary (private) values on alternatives. *No positive externalities* requires that each agent’s presence in the economy has a *negative* effect on the value of other agents in the efficient solution. We return to this requirement below. *Single item* environments are those for which agents have a private value for a single item to be allocated.

The celebrated Groves [Gro73] family of mechanisms, which are truthful DRMs (i.e. DSIC), are defined with outcome rule:

$$g(\theta) = \arg \max_{a \in A} \sum_i v_i(a; \theta_i) \quad (1.6)$$

¹³If a mechanism can make an agent commit before it even learns its own type then *ex ante* IR can be reasonable. This is possible, for instance, if a population of agents (such as the U.S. Congress) chooses a mechanism for decision making some time ahead of when their individual preferences are realized. If a mechanism can make an agent commit after it learns its own type but before learning the outcome of the mechanism then *interim* IR is reasonable.

¹⁴Dasgupta et al. [DHM79], Jackson [Jac03] and Mas-Colell et al. [MCWG95] provide additional examples.

and payment rule:

$$p_i(\theta) = h_i(\theta_{-i}) - \sum_{j \neq i} v_j(g(\theta); \theta_j) \quad (1.7)$$

where $h_i : \Theta_{-i} \rightarrow \mathbb{R}$ is an arbitrary function on the reports of all agents except i . This freedom in selecting h_i leads to the description of a “family” of mechanisms. Different choices make different tradeoffs between the desiderata of WBB and IR.

A Groves mechanism selects an alternative that maximizes the reported values of all agents and then makes a payment to each equal to the total reported value to the other agents for the decision, with an agent also paying the mechanism some amount that is independent of its own report.

To see that Groves mechanisms are strategy-proof, consider an agent with utility $u_i(a, p) = v_i(a; \theta_i) - p$ for alternative a at price $p \in \mathbb{R}$. Now, fixing the reports from other agents, $\hat{\theta}_{-i}$, the utility to agent i given report, $\hat{\theta}_i$, is:

$$\pi_i(\hat{\theta}_i) = v_i(g(\hat{\theta}_i, \hat{\theta}_{-i}); \theta_i) + \sum_{j \neq i} v_j(g(\hat{\theta}_i, \hat{\theta}_{-i}); \hat{\theta}_j) - h_i(\hat{\theta}_{-i}) \quad (1.8)$$

Ignore the final term, which is independent of the agent’s report.¹⁵ Then, the only effect that agent i ’s report has on its utility in Eq. (1.8) is via the choice by the center of alternative $a^* = g(\hat{\theta}_i, \hat{\theta}_{-i})$. Agent i maximizes Eq. (1.8) by reporting $\hat{\theta}_i = \theta_i$ so that in choosing a^* to solve $\arg \max_a \sum_i v_i(a; \hat{\theta}_i)$ the center will maximize Eq. (1.8), i.e. choose an alternative to maximize agent i ’s true value and the total reported values of the other agents.

Thus, the payment term in a Groves mechanism is defined to align the incentives of every agent with that of maximizing the total value to all agents. This simple idea provides truthfulness. Groves mechanisms are EFF for all environments by definition of outcome rule g and from their DSIC property.

The first term in the Groves payment rule can be used to achieve IR while also maximizing the total payments made to the center. This corresponds to the *Vickrey-Clarke-Groves* [Cla71, Gro73, Vic61] mechanism.

Definition 1.13 The *Vickrey-Clarke-Groves* (VCG) mechanism is a Groves mechanism with:

$$h_i(\theta_{-i}) = \sum_{j \neq i} v_j(g(\theta_{-i}); \theta_j), \quad (1.9)$$

¹⁵Note that it does not even depend on the agent’s report via the strategies of other agents since they cannot condition on agent i ’s report in a DRM.

where $g(\theta_{-i}) = \arg \max_{a \in A} \sum_{j \neq i} v_j(a; \theta_j)$, i.e. an efficient alternative without agent i .

The total payment by agent i in the VCG mechanism is the *marginal negative effect that agent i has on the total value to the other agents by its presence*.

Example 1.14 The special case of VCG mechanism for the allocation of a single item is the familiar second-price sealed-bid auction, or Vickrey [Vic61] auction. In this case, with bids $\hat{\theta}_1$ and $\hat{\theta}_2$ to denote the first- and second-highest bids, the item is sold to the agent with the highest bid (agent 1), for a price computed as $p_1(\hat{\theta}) = \sum_{j \neq 1} v_j(g(\hat{\theta}_{-1}); \hat{\theta}_j) - \sum_{j \neq 1} v_j(g(\hat{\theta}); \hat{\theta}_j) = \hat{\theta}_2 - 0 = \hat{\theta}_2$, i.e. the second-highest bid. Notice that we have IR here because $\hat{\theta}_2 \leq \hat{\theta}_1 = \theta_1$ in equilibrium and WBB because $\hat{\theta}_2 \geq 0$ and all other payments are zero.

Let $V(N) = \max_{a \in A} \sum_{i \in N} v_i(a; \theta_i)$, i.e. the total value from the efficient choice. Simple algebraic manipulation establishes that the VCG mechanism satisfies IR whenever the environment satisfies the following *non-negative marginal product* condition:

$$V(N) \geq V(N \setminus i), \quad (1.10)$$

for all types θ and all agents i . Introducing an agent should never reduce the total value available from the maximal alternative. This is reasonable, holding in market environments such as exchanges because all trades remain feasible when introducing additional agents.¹⁶

The VCG mechanism also satisfies WBB when each agent's payment is non-negative, for which we need the following *no positive externalities* condition:

$$\sum_{j \neq i} v_j(g(\hat{\theta}_{-i}); \hat{\theta}_j) \geq \sum_{j \neq i} v_j(g(\hat{\theta}); \hat{\theta}_j) \quad (1.11)$$

Removing agent i should allow the remaining agents to achieve at least as much value from the maximal alternative as they achieve when agent i is present. This holds, for instance, in an auction setting with a seller with no intrinsic value for the goods, but not in an exchange because agent i could be a seller and facilitate new trades and thus have a positive externality on the other agents. The VCG mechanism also satisfies WBB in a public

¹⁶This also holds in public choice problems, when introducing a new agent cannot change the range of public projects that can be implemented and no agent has negative value for any public project (in relation to no choice being made). It may not hold in environments with physical congestion, for instance when introducing an additional robot can block the paths of all robots.

project choice problem because the set of choices available is static however many agents are in the system.

The VCG mechanism is especially useful when EFF is a primary goal but revenue optimality a secondary goal.

Theorem 1.15 [KP00] *The VCG mechanism maximizes the expected revenue (and thus comes the closest to satisfying BB) amongst all EFF, IR and BNIC mechanisms.*

Other than having practical importance, for instance to the designer of an efficient marketplace that nevertheless wishes to drive as much revenue as possible to sellers, this result is also useful in establishing some of the central *impossibility* results in MD. See Krishna and Perry [KP00] for an extended discussion.

In many environments the fact that the VCG mechanism runs at a surplus to the center may be undesirable. Consider, for instance, a group of friends using a Vickrey auction to decide who should use a shared car for the evening. They would rather not “burn” the proceeds of the auction (or, for the sake of argument, give the proceeds to charity), but cannot blindly return the collected payment to the participants without compromising truthfulness. In addressing this loss of utility by the participants, one approach is to sacrifice some efficiency in return for strong BB [Fal04]. Another approach is to leverage structure in agent valuations and redistribute payments back to agents in a way that maintains truthfulness and full efficiency (but necessarily without achieving strict BB) [Cav06, GC07].

Continuing, we make some brief remarks about the d’AGVA [Arr79, dG79] mechanism, often referred to as the *expected externality* mechanism (see also Mas-Colell et al. [MCWG95]) because of its connection with the VCG mechanism.

The d’AGVA mechanism is interesting because it achieves EFF and strong ex post (strong) BB by relaxing ex post IR to ex ante IR and DSIC to BNIC. The outcome rule is the same as for the Groves mechanism but d’AGVA is not a Groves mechanism and the payment rule is instead defined as:

$$p_i(\theta) = \left(\frac{1}{n-1} \cdot \sum_{j \neq i} SW_{-j}(\theta_j) \right) - SW_{-i}(\theta_i), \quad (1.12)$$

where

$$SW_{-i}(\theta_i) = \mathbb{E}_{\theta_{-i}} \left[\sum_{j \neq i} v_j(g(\theta_i, \theta_{-i}); \theta_j) \right] \quad (1.13)$$

Note that $SW_{-i}(\theta_i)$ *only* depends on the report of agent i and is independent of the reports of all agents $j \neq i$. This term is the expected total value, in equilibrium, for agents $j \neq i$ when agent i announces type $\hat{\theta}_i$. For this reason, the d'AGVA mechanism is BNIC instead of DSIC, with the incentive-compatibility requiring that the other agents bid truthfully and according to the distribution that defines the payment. In Groves, this term would be the *actual* value to the rest of the agents. Similarly, d'AGVA is *ex ante* IR (in environments such as exchanges, that satisfy positive marginal product), which is a critical weakness in many domains.¹⁷

The possibility results available for revenue optimality are less general but extremely interesting. While the VCG mechanism is revenue optimal across all EFF and IR mechanisms, it is typically possible to achieve better (expected) revenue by sacrificing some efficiency. In seminal work, Myerson [Mye81] constructs a revenue optimal single-item auction. See Klemperer [Kle00] for a more recent treatment. For agents with i.i.d. types (i.e. symmetric priors), the optimal auction is equivalent to a Vickrey auction with a reserve price. The item is sold to the highest bidder if the bid price is greater than the reserve price, and sold at the reserve price or second-highest bid, whichever is greater. By setting a reserve price the auction will sometimes sell the item for more than the Vickrey price, however the auction will sometimes forfeit a sale and is generally inefficient.

More generally, the optimal auction allocates the item with the highest “virtual valuation.” The virtual valuation is determined as an adjustment from the reported value of an agent and depends on the prior distribution for that agent’s value [Mye81]. In this case the optimal auction is Bayes-Nash IC, such an auction an auction can achieve more revenue than a DSIC mechanism. Revenue-optimal auctions are not known for general valuation environments, such as *combinatorial auctions* in which bidders want to buy bundles of items and have complements (“I only want A if I also get B ”) and substitutes (“I only want A or B ”) values. See [JtVM07, Led07, Ü06, IK06, MV04] for recent progress in restricted settings, and Likhodedov and Sandholm [LS04b, LS05, Voh07, CHK07] for a computational approach.

¹⁷The first term in Eq. (1.12), as in Groves mechanisms, is agent independent. Here, it represents the average (over agents j except i) of the estimated total “without j ” value given report θ_j and the distribution on types. Taken together, this gives strong *ex post* (strong) BB, with each agent except i making a payment back to the mechanism equal to a $1/(n-1)$ share of the payment made by the mechanism to agent i .

1.2.4 *Impossibility Results*

The impossibility results in MD appeal to the revelation principle and work with incentive-compatibility conditions to establish combinations of desiderata that cannot be achieved in any mechanism. These negative results arise entirely as a result of the private information in multi-agent systems; this coupled with agent self-interest has a cost in terms of properties that can be achieved.

Table 1.3 describes some important impossibility results. Results are delineated by agent utility (e.g. unrestricted utilities, or quasilinear (QL)), the equilibrium solution concept, and by any structure allowed in the value environment. The “Impossible” column lists the combinations of desirable mechanism properties that cannot be achieved in each case. Impossibility for restricted preferences and structured environments implies impossibility for more general settings; similarly, impossibility for weak solution concepts such as BNE imply impossibility for stronger solution concepts such as DSE.

The Gibbard [Gib73] and Satterthwaite [Sat75] impossibility theorem (GibSat) states that for unrestricted preferences and at least 3 alternatives only *dictatorial* social choice functions can be implemented in DSE. See Mas-Colell et al. [MCWG95] for a proof. A mechanism is dictatorial if there is some agent i , defined independently of agent strategies s (although perhaps at random), that always receives one of its most-preferred alternatives (from the alternatives in the range of the outcome rule given the strategies of the other agents.) Although Pareto efficient, a dictatorial social choice function is undesirable for other reasons since it does a poor job of aggregating preference information.

The Gibbard-Satterthwaite impossibility result at first appears very negative. However the assumption of unrestricted preferences is a strong one. Most real environments will impose some structure. For instance, in allocation problems it is common to assume free disposal (weakly-increasing value for allocations of more goods) and no-externalities (indifference to the distribution of goods across other agents). In a voting setting, positive results are available again when values are “single-peaked” (e.g. candidates fall on a spectrum from the political left to the political right) [Jac03].

The Hurwicz [Hur75] and Myerson-Satterthwaite [MS83] impossibility results are significant, then, because they hold even with quasi-linear utility. Hurwicz (see Groves and Ledyard [GL77b] for a discussion) precludes EFF and (strong) BB mechanisms in DSE. The result holds in a simple exchange environment, for instance in which a single unit of a resource is to be allocated amongst a group of agents. Myerson-Satterthwaite further strengthens

Name	Utility	Solution concept	Impossible	Valuation environment
GibSat	unrestricted	DSE	Non-Dictatorial	≥ 3 alternatives
Hurwicz	QL	DSE	EFF & BB	simple-exchange
MyerSat	QL	BNE	EFF, WBB & <i>interim</i> IR	simple-exchange
Roberts	QL	DSE	non linear affine	no structure
Holmström	QL	DSE	EFF & non-Groves	smoothly-connected
LMN	QL	DSE	non linear affine & IIA	order-based

Table 1.3. *Mechanism Design Impossibility Results.*

the result by establishing that EFF and (weak) BB (and thus also strong BB) is impossible even with BNIC, if one also requires interim (and thus also ex post) IR.¹⁸ An immediate consequence of these results is that we can only hope to achieve at most *two* of EFF, IR and WBB in many market settings.¹⁹

Kevin Roberts [Rob79] showed, for unrestricted valuations but quasilinear utilities, that the *only* social choice functions that can be implemented in a dominant strategy equilibrium are the *affine maximizers*,

$$f(\theta) = \arg \max_{a \in A} \sum_i \alpha_i v_i(a; \theta_i) + \gamma(a), \quad (1.14)$$

for $\alpha_i \in \mathbb{R}_{\geq 0}$ and $\gamma(a) \in \mathbb{R}$. The affine maximizers contain the EFF social choice function, which is an affine maximizer for which $\alpha_i = 1$ for all $i \in N$ and $\gamma(a) = 0$ for all $a \in A$.

Groves mechanisms easily generalize to implement linear-affine maximizers, and so it would seem from Roberts' result that Groves mechanisms are the only DSIC mechanisms available with quasilinear preferences. Not so! Roberts' result crucially relies on the assumption that valuations are unrestricted. In fact, many domains impose considerable structure on agent valuations, for example with free disposal and no-externalities. To paraphrase Lavi-Mu'alem-Nisan [LMN03] (LMN), "the assumption of unrestricted valuations is not without restriction." LMN extend Roberts to hold in value environments that can be described as *order-based* domains. This provides additional structure and includes, for example, the domain of combinato-

¹⁸The centrality of the Groves mechanisms can be used to establish the Myerson-Satterthwaite impossibility result; see Krishna and Perry [KP00].

¹⁹See Babaioff and Walsh [BW05] for a recent discussion of mechanisms for two-sided markets.

rial auctions.²⁰ LMN also requires *independence of irrelevant alternatives* (IIA)²¹ and two other technical conditions.

In fact, Holmström [Hol79] (see also Green and Laffont [GL77a]) shows that Groves mechanisms are the only available DSIC mechanisms when *EFF is required* and when the valuation environment is *smoothly connected*.²² We defer to Holmström for the technical definition of a smoothly connected domain. It suffices for our purposes to note that the primary example of a smoothly connected domain is a convex domain. Valuation domain Θ_i is convex if for any $\theta_i, \theta'_i \in \Theta_i$, and corresponding valuations $v_i(a; \theta_i)$ and $v_i(a; \theta'_i)$, then type $\theta''_i \in \Theta_i$ such that

$$v_i(a; \theta''_i) = \lambda v_i(a; \theta_i) + (1 - \lambda)v_i(a; \theta'_i), \quad (1.15)$$

for all $\lambda \in [0, 1]$. This should not be confused with a statement that an agent's valuation function is convex. This is a property on the *domain* of valuations, not on the valuations themselves.

Order-based domains are convex, and thus this characterization result encompasses CAs and multi-unit auctions, as well as other domains outlined in Saks and Yu [SY05]. Single-minded combinatorial auctions (CAs)²³ provide a domain that is smoothly-connected but not convex [CP05], and thus Holmström's result is also relevant here.

Remark: It is instructive to compare the possibility and impossibility results and understand the tradeoffs that are made. For instance, one can see that BNIC is sometimes more useful than DSIC by contrasting the d'AGVA positive result with the Green-Laffont impossibility result.

1.2.5 Exact Characterizations

In the light of the many possibility and impossibility results, it is also interesting to note the kinds of *exact* characterizations that are available for implementation in a dominant-strategy equilibrium. We will briefly summarize some of what is known in this regard. All of what follows is for quasilinear preferences.

²⁰An order-based valuation domain is one in which ordinal constraints on agent valuations can be used to characterize the domain of types. For example, free disposal says that any alternative a in which agent i gets more goods than some alternative b has more value to agent i .

²¹A social choice function satisfies IIA if for any $\theta, \theta' \in \Theta$, if $f(\theta) = a$ and $f(\theta') = b \neq a$ there exists an agent i such that $v_i(a; \theta_i) - v_i(b; \theta_i) \neq v_i(a; \theta'_i) - v_i(b; \theta'_i)$.

²²Holmström's result is incomparable with that of Roberts. Holmström is imposing the requirement of EFF while working with a more general set of valuations.

²³In a single-minded CA each agent has a value for some particular bundle of items, with both this value and the bundle in which it is "single-mindedly" interested, private to the agent.

Unrestricted Valuation Domains From Roberts [Rob79], the only social choice functions that can be implemented in DSE are affine maximizers. Moreover, we know that any affine maximizer can be implemented by a simple modification to the Groves mechanism. Therefore, with this (restrictive) assumption of an unrestricted valuation domain the affine maximizers provide an exact characterization of the social choice functions that can be truthfully implemented.

Convex Valuation Domains Saks and Yu [SY05] establish that the property of *weak monotonicity* (W-MON) (see also Bikhchandani et al. [BCL⁺06]) is an exact characterization of the truthful social choice functions when the valuation domain is convex.

Definition 1.16 A social choice function f satisfies W-MON if whenever $f(\theta_i, \theta_{-i}) = a$, and $f(\theta'_i, \theta_{-i}) = b$ then $v_i(b; \theta'_i) - v_i(a; \theta'_i) \geq v_i(b; \theta_i) - v_i(a; \theta_i)$.

If the alternative changes from a to b for a change in agent i 's type then alternative b should not be less preferred relative to a . It is straightforward to show that W-MON is *always* necessary for a social choice function to be truthfully implementable, irrespective of the valuation domain.²⁴ The contribution of Saks and Yu [SY05] is to show that W-MON is sufficient for truthfulness in a convex domain. As noted above, this domain subsumes the order-based preference domain [LMN03] and includes many practical economic environments, including CAs, multi-unit auctions, and auctions with marginal-decreasing values.

Single-Minded Valuation Domains A valuation domain is *single-minded* if agents have single-minded valuations. If an agent has a single-minded valuation, then its type, $\theta_i = (L_i, w_i) \in (A, \mathbb{R}_{\geq 0})$, defines a valuation function:

$$v_i(a; \theta_i) = \begin{cases} w_i & , \text{ if } a \succeq_{A,i} L_i \\ 0 & , \text{ otherwise} \end{cases} \quad (1.16)$$

This valuation is defined with respect to a partial-order $\succeq_{A,i}$ on alternatives A , where $a \succeq_{A,i} b$ if alternative a is at least as preferred to agent i as alternative b . For example, in the setting of a single-minded CA, then the alternatives $\{a : a \succeq_{A,i} L_i\}$ correspond to those allocations in which agent i gets some bundle of items that she demands (and perhaps additional items

²⁴To see this, suppose $g(\theta_i, \theta_{-i}) = a$ and $g(\theta'_i, \theta_{-i}) = b$ and g is truthful, and has corresponding agent-independent and admissible price function $\pi_i(a, \theta_{-i})$. By admissibility, we have $v_i(a; \theta_i) - \pi_i(a, \theta_{-i}) \geq v_i(b; \theta_i) - \pi_i(b, \theta_{-i})$ and $v_i(b; \theta'_i) - \pi_i(b, \theta_{-i}) \geq v_i(a; \theta'_i) - \pi_i(a, \theta_{-i})$. Combining, this gives $v_i(b; \theta'_i) - v_i(a; \theta'_i) \geq v_i(b; \theta_i) - v_i(a; \theta_i)$, which is W-MON.

as well, and irrespective of the allocation to other agents.) Define a partial-order, $\succeq_{\Theta, i}$ on types, with

$$(\theta'_i \succeq_{\Theta, i} \theta_i) \Leftrightarrow (w'_i \geq w_i) \wedge (L_i \succeq_{A, i} L'_i) \quad (1.17)$$

Adopt $f_i(\theta_i, \theta_{-i}) \in \{0, 1\}$ as shorthand for whether or not agent i is “satisfied” (i.e. has value) for the allocation chosen by social choice function f . Given this, we can define

Definition 1.17 A social choice function f is monotonic in a single-minded domain if whenever $f_i(\theta_i, \theta_{-i}) = 1$ and $\theta'_i \succeq_{\Theta, i} \theta_i$, then $f_i(\theta'_i, \theta_{-i}) = 1$.

This simplified form of monotonicity is necessary and sufficient for a truthful social choice function (see Lehmann et al. [LOS02] for an early treatment). The corresponding payment, that makes a mechanism with a monotonic outcome rule truthful, is the *critical value* payment; where a satisfied agent makes payment $p_i(\theta) = \min_{w'_i}$ s.t. $f_i(\theta'_i, \theta_{-i}) = 1$, where $\theta'_i = (L_i, w'_i)$.

One can also cast this result into a *one-dimensional* domain in which each agent’s private information is a single number that defines its value for some (known) set of alternatives.²⁵ In such domains, the notion of monotonicity simplifies to *value-monotonicity* [Mye81, AT01]. Writing $f(w)$ for $w \in \mathbb{R}^n$ to emphasize that the type profile is now a vector of numbers, a social choice function f is value-monotonic if whenever $f_i(w_i, w_{-i}) = 1$ and $w'_i \geq w_i$ then $f_i(w'_i, w_{-i}) = 1$ (where $f_i(w) = 1$ if and only if agent i is satisfied by the outcome). This is necessary and sufficient for truthfulness.

Arbitrary Valuation Domains A generalized form of monotonicity, known as *cycle monotonicity*, is necessary and sufficient for truthfulness in discrete, but otherwise *arbitrary*, valuation domains. This early result due to Rochet [Roc87] is fully general, and holds whatever the structure on the domain (i.e. in all of the aforementioned settings).²⁶ Rather than state the condition formally we refer the interested reader to Gui et al. [GMV04] and Lavi and Swamy [LS07] for a useful exposition. The generality of cycle monotonicity comes at some cost: the concept can be quite unwieldy to work with, and only recently has cycle monotonicity been used for practical mechanism design [MV04, Voh07, LS07].

²⁵An example of such a setting is the so-called “known single-minded CA” [MN02], where the bundle in which an agent is interested in is known and only its value is private.

²⁶By Roberts [Rob79], we know that cycle monotonicity is equivalent to affine-maximization for unrestricted domains. On the other hand, as we shall see in the next section, when the valuation domain imposes structure on agent valuations (e.g. in single-minded CAs), then cycle monotonicity does not imply affine-maximization.

1.3 Computation by the Center

Computational MD seeks to apply MD to a variety of real-world domains and, in constructing actual mechanisms, needing to integrate the concerns of computation and communication efficiency from computer science with the incentives concern of economics. In this section we restrict our attention to *direct-revelation* mechanisms, and consider the computational complexity of the problem of computing the outcome of a mechanism. Nisan and Ronen [NR01] introduced this agenda, requiring useful economic properties as well as *polynomial-time* computability. Computational considerations are especially interesting because they interact with incentive considerations. For example, naively adopting an approximation algorithm within the VCG mechanism leads to the unraveling of incentives.

1.3.1 Combinatorial Auctions

A canonical problem in CMD is that of implementing an efficient allocation in combinatorial auctions. CAs are of practical interest with many applications (see Chapters 20–23 in Cramton et al. [CSS06]), and of theoretical appeal because the efficient allocation problem is NP-hard and also *inapproximable* (under reasonable assumptions).

Before continuing we give a very brief introduction to complexity theory, as it applies to optimization problems such as the winner determination problem (WDP) in a CA. This exposition is based on that in Lehmann et al. [LMS06]. An algorithm solves a problem in polynomial time if the run time (measured in basic arithmetic operations) is bounded above by some polynomial function of the size of the input. Complexity theory is applied to *decision problems*, in which some instance is presented and the algorithm must decide whether the instance has a particular property. For example, “is there a feasible allocation with value at least \$100” would be a decision problem in the context of the WDP.

P is the class of decision problems that have a polynomial time algorithm. It is common to consider problems in P to be tractable while problems outside of P to be intractable. NP is the class of decision problems for which there exists a polynomial time algorithm that can *check* the validity of a property, when given as input the instance and a certificate for the validity. Clearly $\text{NP} \supseteq \text{P}$. A decision problem is NP-hard if a polynomial time algorithm would imply a polynomial time algorithm for all problems in NP; the special case of a decision problem that is NP-hard and also in NP is said to be NP-complete. Finally, *an optimization problem whose decision version is NP-complete* is NP-hard. It is generally believed, but

never proven, that $P \neq NP$. The implication is that it is thought unlikely that an NP-hard optimization problem can be solved in polynomial time.

Before stating the complexity result for the WDP in CAs we need to be precise about the representation of the input. In a CA there is a set of goods G , $|G| = m$, and each alternative $a \in A$ corresponds to an allocation of goods, denoted $S = (S_1, \dots, S_n)$, so that agent i receives bundles of goods $S_i \subseteq G$. An allocation is *feasible* if

$$S_i \cap S_j = \emptyset, \quad \forall i, j \in N \quad (1.18)$$

Let $\mathbb{F}(G)$ denote the set of feasible allocations. Since an agent's value for alternative a depends only on its own allocation we write $v_i(S; \theta_i)$ to denote agent i 's value for bundle S . Valuations are normalized, with

$$v_i(\emptyset; \theta_i) = 0, \quad (1.19)$$

for the empty bundle \emptyset and satisfy *free disposal*, with

$$v_i(S; \theta_i) \leq v_i(T; \theta_i), \quad (1.20)$$

for all $S \subseteq T$. The efficient allocation maximizes the total value to all agents, i.e.

$$V(N) = \max_{S \in \mathbb{F}(G)} \sum_i v_i(S_i; \theta_i) \quad (1.21)$$

The representation of the input to the WDP corresponds to the choice of *bidding language*. Many bidding languages have been proposed; see Nisan [Nis06] for a recent survey.

We consider the *exclusive-or* (XOR) bidding language. An agent's valuation $v_i(S; \theta_i)$ is represented in the XOR language as a set of (*bundle, value*) pairs,

$$\theta_i = \{(L_1, w_1), \dots, (L_k, w_k)\} \subseteq 2^G \times \mathbb{R}_{\geq 0}. \quad (1.22)$$

The XOR semantics of the bidding language define valuation function,

$$v_i(S; \theta_i) = \max[w : (L, w) \in \theta_i, L \subseteq S], \quad (1.23)$$

so that an agent's value for a bundle S is the maximal value of all bundles contained in L that are explicitly stated in the XOR bid.

Example 1.18 For instance, an XOR valuation $\{(AB, 10), (ABC, 15), (D, 20)\}$ indicates that an agent has value 10 for AB , 15 for any bundle containing ABC that does not contain D , and 20 for any bundle containing D . Note that an agent has value 20 and not 35 for $ABCD$.

Given an XOR representation the problem of solving the WDP is NP-hard. This can be established from its equivalence to the NP-hard weighted set packing problem [RPH98]. This is true even if we restrict instances to single-minded valuations (described in detail below), and even if every bid has a value equal to 1 and every bidder only bids on bundles of size at most 2 [LOS02, LMS06]. The winner determination problem is also *inapproximable*, meaning that no polynomial algorithm can have a competitive ratio²⁷ better than $\min(l^{1-\epsilon}, m^{1/2-\epsilon})$ unless $\text{NP}=\text{ZPP}$,²⁸ where l denotes the total number of bundles in the XOR value representation across all agents, and m is the number of items [H99, San02, LOS02].

On the other hand, the picture is not all negative. The following three kinds of computational results exist for the WDP in CAs:

- There exist polynomial-time algorithms for the WDP for restricted problems; e.g., for sub-classes of agent valuations such as *substitutes* valuations, or when all bidders demand bundles of items that have a single-ordering property, where there is a circular order can be imposed so that all bundles contain a contiguous sequence of items [M06, dVV03, RPH98]. There are also better worst-case approximation guarantees for problems with valuations that do not exhibit complements [DNS05, DS06].
- Heuristic algorithms, for instance local search algorithms or LP-rounding approaches, can provide good empirical performance on some problem distributions [HB00, ZN01].
- Algorithms have been developed that can solve large problems (with tens of thousands of bids and items) to *optimality* in economically feasible times. These exploit structure in bid representations, and leverage new advances in branch-and-cut technology for solving integer programs [NW99, dVSV07, ATY00, San06, SSSL05].

1.3.2 Case Study: Single Minded CAs

Despite the progress that has been made on the WDP in CAs, the problem remains NP-hard and combinatorial auctions provide a nice setting in which to understand some of the tensions that exist between DSIC and compu-

²⁷The competitive ratio of an optimization algorithm B over a set of inputs X is defined as $\min_{x \in X} V_B(x)/V^*(x)$ where $V_B(x)$ denotes the value of the solution computed by B on input x and $V^*(x)$ denotes the value of the optimal solution.

²⁸ZPP is a sub-class of NP that consists of those decision problems for which there exists an algorithm that can check the validity of a property, when given as input the instance and a certificate for validity, in *expected* polynomial time. The question of whether $\text{NP}=\text{ZPP}$ is also an open problem, although it is generally believed that $\text{NP} \neq \text{ZPP}$.

tational tractability in CMD. To illustrate these tensions we consider the special case of CAs with *single-minded* valuations.

In defining these valuations we follow the outline for single-minded valuations introduced in Section 1.2.5. In a single-minded CA, an agent's type $\theta_i = (L_i, w_i)$ defines a single interesting bundle, L_i , such that:

$$v_i(S; \theta_i) = \begin{cases} w_i & , \text{ if } S \supseteq L_i \\ 0 & , \text{ otherwise.} \end{cases} \quad (1.24)$$

As noted above, the WDP with single-minded bids remains NP-hard [LOS02]. It is interesting, then, to consider a simple *greedy* approximation algorithm. The algorithm sorts the bids in decreasing order of $w_i/|L_i|$ and then performs a single pass in rank order, allocating a bid when it is feasible given the bids already accepted. This algorithm is “greedy” because it accepts bids in order of their per-item value, without consideration of how these bids might fit with other bids.

Example 1.19 Given types $\theta_1 = \{(A, 10)\}$, $\theta_2 = \{(AB, 19)\}$, $\theta_3 = \{(B, 8)\}$. The greedy algorithm orders the bids (1,2,3) and implements allocation (A, \emptyset, B) , with bid 2 not allocated because item A is allocated to bid 1, which has a higher rank.

In what follows we consider a slight generalization to this algorithm, in which the bids are ordered by $w_i/|L_i|^b$ for some $b > 0$. Let $g_b(\theta)$ denote the allocation computed by a greedy algorithm, where $b > 0$ is the parameter that defines the ranking function (with $b = 1$ above).

It is instructive to consider the effect of defining a VCG-based mechanism in which this outcome rule g_b is used to determine the outcome and also the payments:

$$p_i(\theta) = \sum_{j \neq i} v_j(g_b(\theta_{-i}); \theta_j) - \sum_{j \neq i} v_j(g_b(\theta); \theta_j) \quad (1.25)$$

The payments defined so that agent i pays the negative externality that it imposes on agents $\neq i$ by its presence, given that decisions are made according to the greedy algorithm.

Example 1.20 Consider again types $\theta_1 = \{(A, 10)\}$, $\theta_2 = \{(AB, 19)\}$, $\theta_3 = \{(B, 8)\}$, and greedy algorithm with $b = 1$. We have $g_b(\theta) = (A, \emptyset, B)$. Now remove agent 1 to determine $g_b(\theta_{-1}) = (\emptyset, AB, \emptyset)$, and remove agent 3 to determine $g_b(\theta_{-3}) = (A, \emptyset, \emptyset)$. Agent 1's payment is $19 - 8 = 11$ and agent 3's payment is $10 - 10 = 0$. The mechanism is not IR for agent 1. Moreover, it is not truthful. Suppose agent 2 bids 30 instead of 19. Then the outcome

is $(\emptyset, AB, \emptyset)$ and its payment is $18 - 0 = 18$ since $g)b(\theta_{-2})$ selects outcome (A, \emptyset, B) .

We see the tension between truthful and tractable mechanisms. One cannot just naively substitute an approximation algorithm into the VCG mechanism. More generally, an approximation algorithm must be *maximal in range* for truthfulness to be retained [NR00], meaning that it must be optimal for some *a priori* restricted range of alternatives.

In this case there is a simple solution to the apparent conundrum. Rather than use a VCG-based method to define the payment we can collect as payment from a winning agent the smallest bid that the agent could have made and still won [LOS02]. This is the critical-value payment introduced in Section 1.2.5.

Let $g_{b,i}(\theta) = 1$ when agent i is successful given type reports θ and $g_{b,i}(\theta) = 0$ otherwise. The payment is defined as:

Definition 1.21 (critical value payment) Given agent i 's report is (L_i, w_i) , reports θ_{-i} from agents $\neq i$, and greedy allocation rule g_b , the critical value payment collected from a winning agent i is

$$p_i(L_i, \theta_{-i}) = \min\{w'_i \geq 0 : \theta'_i = \{(w'_i, S)\}, g_{b,i}(\theta'_i, \theta_{-i}) = 1\} \quad (1.26)$$

Example 1.22 In the earlier example, with $\theta_1 = \{(A, 10)\}$, $\theta_2 = \{(AB, 19)\}$, $\theta_3 = \{(B, 8)\}$, and parameter $b = 1$, we implement outcome (A, \emptyset, B) and collect payments $p_1(\theta) = 19/2 = 9.5$ (since bid 1 must rank above bid 2 to win) and $p_3(\theta) = 0$ (since bid 3 would be allocated for any bid value.)

It is easy to check for this example that the new payment rule removes any incentive for any of the agents to deviate and misreport their private type. In particular, agent 2 can no longer do better by over-reporting his value because if he wins then he pay 20 (since this is the smallest bid value at which his bid will be ranked above that of agent 1).

The LOS mechanism is truthful because the greedy allocation rules are monotonic, with $g_{b,i}(\theta_i, \theta_{-i}) = 1 \Rightarrow g_{b,i}(\theta'_i, \theta_{-i}) = 1$, $\forall \theta'_i \succ \theta_i$, where $\theta'_i = (L'_i, w'_i) \succ \theta_i = (L_i, w_i)$ if and only if $L'_i \subseteq L_i$ and $w'_i > w_i$.

Theorem 1.23 [LOS02] *The LOS mechanism is truthful for all monotonic greedy outcome rules. When parameter $b = 1/2$ then the mechanism achieves competitiveness $m^{1/2}$ with respect to efficiency, which is the best achievable across all polynomial-time mechanisms for this problem (unless $P=NP$), and when $m \leq n^2$.*

Proof Fix θ_{-i} and consider agent i with type (L_i, w_i) . Case (a): agent i wins. First, fix some report \hat{L}_i . Agent i does not want to misreport w_i because its payment is independent of w_i . Second, fix report $\hat{w}_i = w_i$ and consider a misreport $\hat{L}_i \neq L_i$. A report $\hat{L}_i \not\supseteq L_i$ is not useful because the agent is single-minded. But then, a report $\hat{L}_i \supset L_i$ only leads to a *higher* payment. This is because of monotonicity. If $p_i(\hat{L}_i, \theta_{-i}) < p_i(L_i, \theta_{-i})$ for some $\hat{L}_i \supset L_i$ then allocation rule g_b fails monotonicity because agent i would win with report $(\hat{L}_i, p_i(\hat{L}_i, \theta_{-i}) + \epsilon)$ but lose with report $(L_i, p_i(\hat{L}_i, \theta_{-i}) + \epsilon)$, for $p_i(\hat{L}_i, \theta_{-i}) < p_i(\hat{L}_i, \theta_{-i}) + \epsilon < p_i(L_i, \theta_{-i})$. Case (b): agent i loses. The only interesting misreports are (\hat{L}_i, \hat{w}_i) with $\hat{L}_i \supseteq L_i$ and it is again WLOG to focus on $\hat{L}_i = L_i$ because of monotonicity, since the critical-value payment increases with larger bundles. Fixing $\hat{L}_i = L_i$, agent i cannot win with a bid $\hat{w}_i < w_i$ by monotonicity, and will pay more than its true value if it wins because the critical value $p_i(L_i, \theta_{-i}) > w_i$.

The proof that the competitive ratio is $m^{1/2}$ when $b = 1/2$, and that this is tight, is omitted in the interest of space. \square

Remark. The LOS mechanism stands in contrast to the characterization result of Roberts [Rob79], which holds that in unrestricted valuation domains the only social choice functions that can be truthfully implemented are affine maximizers. The greedy allocation rules g_b for $b > 0$ are not linear-affine maximizers.

1.3.3 Price-Based Characterization

Price-based characterizations provide another way to understand why a mechanism is truthful, and provide a nice complement to the approaches based on monotonicity in that the characterization is simple to state and fully general. The following result is something of a “folk theorem” (and is recently stated, e.g. in [Seg03, BGN03, LMN03, Yok03]):

Theorem 1.24 *A direct revelation mechanism, $M = \langle g, p \rangle$, is truthful (or strategy-proof) if and only if there is a corresponding (agent-independent) **price function** $\pi_i : A \times \Theta_{-i} \rightarrow \mathbb{R}$, with the property that:*

- (i) *(agent-independent prices) the mechanism collects payment, $p_i(\theta) = \pi_i(g(\theta), \theta_{-i})$, i.e. the price defined by some agent-independent price on the alternative selected by the outcome rule.*
- (ii) *(admissible outcome rule) the alternative selected by the mechanism maximizes the utility to every agent given the price function, i.e. $g(\theta) \in \arg \max_{a \in A} [v_i(a; \theta_i) - \pi_i(a, \theta_{-i})]$, for all i and all θ .*

Proof (\Leftarrow) Agent i cannot change prices π_i and maximizes its utility $v_i(a; \theta_i) - p_i(\theta)$ by reporting its true type θ_i by admissibility. (\Rightarrow) Given truthful $M = \langle g, p \rangle$, construct $\pi_i(a, \theta_{-i}) = p_i(\theta'_i, \theta_{-i})$ where $g(\theta'_i, \theta_{-i}) = a$ for some θ'_i and $\pi_i(a, \theta_{-i}) = \infty$ otherwise. For agent-independence, suppose towards a contradiction that $g(\theta) = g(\theta'_i, \theta_{-i}) = a$, but $p_i(\theta) \neq p_i(\theta'_i, \theta_{-i})$, for some $\theta'_i \neq \theta_i$. Without loss of generality, suppose $p_i(\theta) > p_i(\theta'_i, \theta_{-i})$. Then agent i should declare θ'_i . This is a contradiction with truthfulness. For admissibility, suppose towards a contradiction that $g(\theta) = a$ and $v_i(a; \theta_i) - \pi_i(a, \theta_{-i}) < v_i(b; \theta_i) - \pi_i(b, \theta_{-i})$ for some θ and some $b \neq a$. Then agent i should declare θ'_i for which $g(\theta'_i, \theta_{-i}) = b$, which is a contradiction with truthfulness. \square

The sufficiency of an agent-independent price function and an admissible allocation rule for truthfulness is easy to see. The intuition is that of the familiar act of shopping in a supermarket: the mechanism fixes a price on all alternatives $a \in A$ based on the reports of other agents, and then promises to choose the alternative that maximizes an agent's utility based on the reported valuation function $v_i(a, \theta'_i)$ of the agent. The agent should report its true valuation so that the mechanism selects an alternative that maximizes its utility.

Example 1.25 The VCG mechanism can be interpreted as a price-based mechanism. For instance, in the case of CAs, the agent-independent price function that is admissible with respect to the efficient outcome rule g is:

$$\pi_i(S, \theta_{-i}) = V(N \setminus i, G) - V(N \setminus i, G \setminus S_i). \quad (1.27)$$

Where $V(N \setminus i, G) = \max_{S \in \mathbb{F}(G)} \sum_{j \neq i} v_j(S_j; \theta_j)$ and $V(N \setminus i, G \setminus S_i) = \max_{S \in \mathbb{F}(G \setminus S_i)} \sum_{j \neq i} v_j(S_j; \theta_j)$. We leave it as an exercise to verify that this price function corresponds to the payment made by an agent in the VCG mechanism (i.e., with $p_i(\theta) = \pi_i(S_i^*, \theta_{-i})$ where S^* is the efficient allocation), and that it maximizes the utility of an agent (i.e., with allocation S^* maximizing each agent's utility given this price function).²⁹

This price-based approach has been quite useful in developing tractable and truthful mechanisms in other domains.

Consider the following examples:

Multi-item CAs. Bartal et al. [BGN03] consider the multi-item CA problem in which there are multiple copies of each item and each bid is for a

²⁹Yokoo [Yok03] refers to this framework, in the context of CAs, as the “price-oriented rationing free” (PORF) approach.

small number of items. A random ordering is imposed on agents and each agent faces prices on items that are defined in terms of reports from preceding agents. In the case of agents that demand at most one unit of an item and a supply with k duplicates of each item, they design a polynomial-time and truthful mechanism with a worst-case $km^{1/(k-2)}$ -approximation for efficiency.

Digital goods. Price-based methods have led to the development of *prior-free* revenue-competitive auctions for digital goods (i.e. with unlimited supply). The results stand in contrast to those in optimal auction design that leverage prior information on type distributions. A particularly influential idea is that of using some subset of agents (randomly sampled) to define prices (or alternatively a revenue target) for another subset of agents. As long as no agent in the sample set faces the prices constructed from its own set then the prices are agent-independent. Fiat et al. [FGHK02] and Goldberg et al. [GHK⁺06] have pioneered this approach; see also Segal [Seg03] and Baliga and Vohra [BV03].

Truthful CAs. Dobzinski et al. [DNS06] develop a truthful polynomial-time mechanism for the general CA problem with an $m^{1/2}$ -approximation for efficiency. The idea of sampling a subset of agents to define prices is adopted for this purpose, and agent valuations are queried via “demand queries” (see Section 1.4). The randomized mechanism constructs a partition of agents into three sets. The first set of agents is used to estimate the (agent-independent) prices that are first used to parameterize the reserve price in a second-price auction on the master bundle G of goods to the second set of agents, and then within a fixed-price auction with random ordering to the third set of agents.

Budget constraints. Price-based methods have also been adopted to achieve competitive efficiency and revenue results in the presence of budget constraints [BCI⁺05, Abr06], where the truthfulness of the VCG mechanism breaks down because utility functions are no longer quasilinear.

1.3.4 Working with Cycle Monotonicity

Cycle monotonicity is necessary and sufficient for the truthfulness of a social choice function. Gui et al. [GMV04] (and recently Vohra [Voh07]) demonstrate that this can be used to reduce the problem of truthful MD to one of

optimization on a network. This opens up the possibility of applying combinatorial optimization algorithms in the purpose of automated mechanism design (see Conitzer and Sandholm [CS03]). Lavi and Swamy [LS07] provide a general construction, in a scheduling domain, for adapting an approximation algorithm to make it satisfy cycle monotonicity and also to compute the required payments. Although there are as yet just a few results, it seems that working with this general notion of cycle monotonicity (and its graph theoretic interpretation) may hold promise for CMD.

1.4 Efficient Preference Elicitation

The direct revelation mechanisms that we have studied so far, such as the VCG mechanism and the LOS mechanism, are completely centralized. Agents report all of their private information to the center. The center computes an outcome and reports the outcome (and payments) back to the agents.

This computational architecture is unappealing in domains for which it is costly for agents to report their complete type to a center. Consider, for instance, the use of a VCG mechanism to auction the right to operate bus lanes in London [CP06]. An agent would be required to report its value for every possible combination of possible bus lanes, with each combination potentially entailing a new business plan. This is unreasonable, not only for reasons of communication complexity but also because of the *cost of determining valuations*. Similar observations can be made in many business settings: although fitting within the private values model (so that information about the values of other firms may be irrelevant in determining a firm's own value), it can be a costly process to determine value for different outcomes; e.g., requiring business meetings, information gathering, or solving complex optimization problems [San93, CJ07, Par05].

In this section, we consider *indirect mechanisms*, such as ascending price auctions, and mechanisms that interact with agents through multiple rounds of preference elicitation. The problem of CAs, paradigmatic in CMD, will continue to attract much of our attention. Indirect mechanisms, such as ascending price CAs, have two main computational advantages over direct mechanisms:

- (i) They can allow agents to avoid unnecessary valuation effort and are often able to implement a social choice function without agents reporting (or even needing to know) their exact value for all possible alternatives. For instance, the winning agent in a single-item ascend-

ing auction does not need to know its exact value for the item, only that its value is greater than the ask price in the last round of the auction. Similarly, the losers do not need to know their exact value, but only that their value is less than the price in the round in which they drop out.

- (ii) They can distribute some computation to agents. We see this most clearly with ascending auctions that can be interpreted as primal-dual or subgradient algorithms: in responding to prices in each round of the auction, agents are performing part of the computation that is required to check for complementary-slackness between primal and dual solutions and thus for the correct implementation of the social choice function [Par01].

We first identify the central role of *competitive equilibrium* (CE) prices in characterizing the minimal information that must be elicited from agents in CAs for the center to determine the efficient allocation, and also VCG payments. This leads to two main paradigms for the design of useful, indirect CAs: (a) ascending price auctions that support “straightforward bidding” in an ex post NE and terminate with CE prices; (b) an approach in which learning theory is adopted to provide polynomial query complexity, with *demand* and *value* queries used to elicit agent preferences.

1.4.1 Case Study: Role of Competitive Equilibrium in CAs

Consider indirect mechanisms for CAs, and let $\text{EFF}(\theta)$ denote the set of efficient allocations for type profile θ . We are interested in indirect mechanisms, $M = \langle \Sigma, g, p \rangle$, that implement $f(\theta) \in \text{EFF}(\theta)$, for all θ , in an ex post Nash equilibrium. That is, we require $g(s^*(\theta)) \in \text{EFF}(\theta)$ for all $\theta \in \Theta$, where s^* is an ex post Nash equilibrium strategy profile. Call such an indirect mechanism an *efficient mechanism*.

In the equilibrium of an efficient mechanism, the messages sent by agents in strategy $s^*(\theta)$, for any θ , must provide enough information about types to define an efficient allocation. Consider message space W . Following Nisan and Segal [NS06], let $\mu : \Sigma \times \theta \rightarrow W$ define the messages $\mu(s^*, \theta) \in W$ sent by agents to the mechanism for strategy profile s^* and type profile θ . Let $\mu^{-1}(s^*(\theta)) \subseteq \Theta$ denote the set of types that are consistent with messages $\mu(s^*, \theta)$, defined as:

$$\mu^{-1}(s^*(\theta)) = \{\theta' : \mu(s^*, \theta') = \mu(s^*, \theta)\} \quad (1.28)$$

An efficient mechanism must have the property that, for all $\theta \in \Theta$, there

is some feasible allocation $T^* \in \mathbb{F}(G)$, for which:

$$\forall \hat{\theta} \in \mu^{-1}(s^*(\theta)) \implies T^* \in \text{EFF}(\hat{\theta}) \quad (1.29)$$

In words, there must always be at least one feasible allocation that is efficient for all types that are possible given the messages reported by agents in equilibrium.

In fact, given that our interest is in implementing the efficient allocation in an ex post Nash equilibrium, and also introducing the goal of maximizing revenue subject to IR constraints, we know from Theorem 1.12 that we must terminate with VCG payments. Thus, an efficient and revenue-maximal and IR mechanism must have the property that, for all $\theta \in \Theta$, there is some feasible allocation $T^* \in \mathbb{F}(G)$ and payments $p \in \mathbb{R}^n$, for which:

$$\forall \hat{\theta} \in \mu^{-1}(s^*(\theta)) \implies (T^*, p) \in \text{VCG}(\hat{\theta}), \quad (1.30)$$

where $\text{VCG}(\theta) \subseteq \mathbb{F}(G) \times \mathbb{R}^n$ denotes the set of efficient allocations and corresponding VCG payments for types θ .

The following simple example illustrates that it is possible to design indirect mechanisms that terminate with the VCG outcome without learning complete information about agent types.

Example 1.26 Consider a “staged Vickrey auction” (reminiscent of the eBay proxy auction). In each round an agent can refine a lower and upper bound on its value, which is maintained by the center. The auction maintains an ask price equal to the current second-highest lower bound and the agent with the highest lower bound (breaking ties at random) as the provisional winner. In each round, an agent must increase its lower bound above the ask price or decrease its upper bound to be no greater than the ask price. Suppose all values are integers. The auction terminates when only one agent is still bidding, that agent pays the final ask price. An ex post Nash equilibrium is for each agent to increase its lower bound by 1 while its value is greater than the current ask price but not the provisional winner, and stop bidding by lowering its upper bound otherwise. Consider an instance with 3 bidders, and types that define their values for the item. When $\theta_1 = 10, \theta_2 = 6, \theta_3 = 4$ the mechanism might terminate with bounds $[7, 12], [6, 6]$ and $[4, 4]$ for agents 1, 2 and 3 respectively. This is the information set $\mu^{-1}(s^*(\theta))$ in this example. Agent 1 wins for 6, which is the outcome of the Vickrey auction. The mechanism never learns the true value of agent 1.

A natural goal that arises in indirect mechanism design is to characterize

the minimal amount of information that must be elicited by *any* mechanism to determine the efficient allocation. This question can be asked both in the context of cooperative agents and self-interested agents. Competitive equilibrium (CE) price theory will provide a nice response to this question and lead, in turn, to natural algorithms for indirect mechanisms.

Let $q_i(S) \in \mathbb{R}_{\geq 0}$ define the price to agent i on bundle of goods $S \subseteq G$. Each agent can face individualized (non-anonymous) prices, so that $q_i(S) \neq q_j(S)$ for $i \neq j$, and non-linear prices, so that $q_i(S) \neq q_i(T) + q_i(T')$ where (T, T') partition the goods in S . We require that prices are *normalized*, so that $q_i(\emptyset) = 0$ and *monotone*, so that $q_i(T) \geq q_i(S)$ for $T \supseteq S$. The language adopted to define prices is a separate issue. Note that it is not necessary to explicitly enumerate the price on all bundles.³⁰

Let $E(N) = \{\theta_1, \dots, \theta_n; G\}$ denote an economy, comprised of a set of agents $N = \{1, \dots, n\}$ and a single seller with G goods.

Definition 1.27 (Demand set) Given prices $q = (q_1, \dots, q_n)$, define agent i 's demand set as:

$$D_i(q; \theta_i) = \{S : v_i(S; \theta_i) - q_i(S) \geq \max_{T \subseteq G} [v_i(T; \theta_i) - q_i(T)]\}, \quad (1.31)$$

where $D_i(q; \theta_i) = \{\emptyset\}$ if $v_i(S; \theta_i) < q_i(S)$ for all $\emptyset \neq S \subseteq G$.

Definition 1.28 Prices $q = (q_1, \dots, q_n)$ are *competitive equilibrium* (CE) prices for economy $E(N)$ if there is an efficient allocation, S^* , for which:

- (i) Bundle $S_i^* \in D_i(q; \theta_i)$ for all $i \in N$
- (ii) Allocation $S^* \in \arg \max_{S \in \mathbb{F}(G)} \sum_{i \in N} q_i(S_i)$

Condition (1) states that the bundle is in the demand set of each agent at the prices. Condition (2) states that the allocation is in the supply set of the seller at the prices, i.e. it maximizes the seller's revenue at the prices. Thus, supply equals demand.

Given that prices can be both non-anonymous and non-linear it is easy to see that CE prices always exist. For instance, the (trivial) prices $q_i(S) = v_i(S; \theta_i)$ for every agent i are CE prices [BO02].

Definition 1.29 Prices $q = (q_1, \dots, q_n)$ are *universal competitive equilibrium* (UCE) prices if they are CE prices and if prices $q_{-i} = (q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_n)$ are CE prices for marginal economy $E(N \setminus i)$, for every agent i .

³⁰For example, an XOR language is adopted for prices in the *i*Bundle auction [PU00] and the representation of prices is no larger than the number of bundles on which agents have placed bids.

agent	\emptyset	$\{A\}$	$\{B\}$	$\{A, B\}$
1	0	3	0	3
2	0	0	6	6
3	0	0	2	4

Table 1.4. *Example valuations to illustrate UCE prices.*

UCE prices always exist. For instance, the (trivial) prices $q_i(S) = v_i(S; \theta_i)$ for every agent i are UCE prices.

Example 1.30 In the single item example, with $\theta_1 = 10, \theta_2 = 6, \theta_3 = 4$, a price $q \in [6, 10]$ is an (anonymous) CE price because agent 1 will demand the item, agents 2 and 3 will have \emptyset in their demand sets, and the seller will want to sell the item. However, only prices $q \in [4, 6]$ are CE for the marginal economy $E(\{2, 3\})$. Thus, the only (anonymous) UCE price in the example is $q \in [6, 10] \cap [4, 6] = \{6\}$.

Example 1.31 Consider an example in which there are two goods A, B and three agents with valuations defined as in Table 1.4, A UCE price vector is the following: $q_1(\emptyset) = q_2(\emptyset) = q_3(\emptyset) = 0$, $q_1(\{A\}) = 2, q_1(\{B\}) = 0, q_1(\{A, B\}) = 2$, $q_2(\{A\}) = 0, q_2(\{B\}) = 4, q_2(\{A, B\}) = 4$, $q_3(\{A\}) = 0, q_3(\{B\}) = 2$ and $q_3(\{A, B\}) = 4$. CE price conditions can be verified for the efficient allocation (A, B, \emptyset) in $E(\{1, 2, 3\})$, allocation (\emptyset, B, A) at prices $q_{-1} = (q_2, q_3)$ in $E(\{2, 3\})$, allocation (A, \emptyset, B) at prices $q_{-2} = (q_1, q_3)$ in $E(\{1, 3\})$ and allocation (A, B, \emptyset) in $E(\{1, 2\})$ at prices $q_{-3} = (q_1, q_2)$.

Theorem 1.32 [LCP05] *Any mechanism that implements the efficient allocation, and satisfies IR while maximizing revenue to the seller (i.e. an indirect VCG mechanism) also elicits enough information to determine universal CE prices.*

Thus, although it is not necessary to elicit exact and complete type information from agents to determine the outcome of the VCG mechanism in an indirect mechanism, one must elicit enough information to determine a set of UCE prices.³¹³²

³¹Nisan and Segal [NS06] (and Parkes [Par02] for a more restricted setting) first showed that CE prices are necessary for EFF. See Lahaie et al. [LCP05] and Segal [CSS06, chapter 11] for a more detailed treatment of this result.

³²This is not to say that price-based indirect mechanisms, which query agents by asking them to respond to prices with best-response bundles, are optimal for elicitation. For example, Nisan and Segal [NS05] demonstrate that there are CA instances that require an exponential number

A universal price equilibrium is also sufficient to compute VCG payments. Given UCE prices q and an efficient allocation S^* , the VCG payments can be computed as:

$$p_i(\theta) = q_i(S_i^*) - [\Pi^s(q) - \Pi^s(q_{-i})], \quad (1.32)$$

where $\Pi^s(q) = \max_{S \in \mathbb{F}(G)} \sum_i q_i(S_i)$ (i.e., the maximal possible revenue to the seller given goods G to sell, prices q , and matching demand) and $\Pi^s(q_{-i}) = \max_{S \in \mathbb{F}(G)} \sum_{j \neq i} q_j(S_j)$ (i.e. the maximal possible revenue to the seller given goods G to sell, prices $q_{-i} = (q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_n)$, and matching demand).

Example 1.33 In the single item example with $\theta_1 = 10$, $\theta_2 = 6$ and $\theta_3 = 4$ and UCE price $q_1 = q_2 = q_3 = 6$, the VCG payment to agent 1 is $6 - [\Pi^s(q) - \Pi^s(q_{-1})] = 6 - (6 - 6) = 6$ since $q = (6, 6, 6)$ and $q_{-1} = (6, 6)$ and the seller's maximal revenue is 6 for both sets of prices.

Example 1.34 In the two good, three seller example in Table 1.4, with UCE prices as defined above we compute the VCG payments as $p_1(\theta) = q_1(\{A\}) - [\Pi^s(q) - \Pi^s(q_{-1})] = 2 - (6 - 4) = 0$, $p_2(\theta) = q_2(\{B\}) - [\Pi^s(q) - \Pi^s(q_{-2})] = 4 - (6 - 4) = 2$ and $p_3(\theta)(\{\emptyset\}) - [\Pi^s(q) - \Pi^s(q_{-3})] = 0 - (0 - 0) = 0$.

The impact of this result is to identify price-based indirect mechanisms as a particularly interesting class of indirect mechanisms to implement the efficient allocation in CAs. There is by now a large literature on ascending price CAs and a smaller literature on the use of price-based queries in more general elicitation mechanisms. We will discuss each in turn. For a more complete survey on work in elicitation for CAs see Sandholm and Boutilier [CSS06, chapter10].

1.4.2 Ascending Price CAs

An ascending price CA maintains (non-anonymous, non-linear) prices p^t in each round and a provisional allocation S^t . A typical ascending price auction maintains prices $q^t = (q_1^t, \dots, q_n^t)$ in each round, and proceeds as follows:

- (i) Collect (perhaps untruthful) demand sets $D_i(q^t) \subseteq 2^G$ from each agent.
- (ii) Solve the winner determination problem to maximize revenue given bids.

of demand queries to determine the efficient allocation while there is a fast efficient elicitation protocol.

- (iii) Check termination conditions. Increase prices if the termination conditions are not met.

Some important design questions in formulating an ascending price auction include: the method used to increase prices, the method used to determine termination, and the *activity rules* that are imposed to restrict the feasible strategy space, e.g. by requiring active participation of bidders across rounds. See Parkes [CSS06, chapter2] for a recent survey.

It has proved very useful to adopt an optimization-based approach to the design of rules for increasing prices and checking termination. With this view, an ascending price auction—when coupled with equilibrium bidding strategies—corresponds to a *primal-dual* algorithm to solve the efficient allocation problem [dVSV07, PU00].³³

Straightforward bidding, in which an agent truthfully reports its demand set in each round of the auction given current prices, can be made an ex post Nash equilibrium by terminating with VCG payments.

In some problems there is a correspondence between *minimal* CE prices (the prices that generate the smallest revenue to the seller across all possible prices) and the VCG payments; e.g., when agents have substitutes but not complements valuations [AM02] the minimal CE prices, \underline{q}_i are such that $\underline{q}_i(S_i^*) = p_i(\theta)$ for all type profiles, θ , where $p_i(\theta)$ is the VCG payment by agent i . In such an environment, one can design an efficient, ascending-price CA by defining “minimal price increases” so that the auction terminates at this minimal CE price vector. The ascending-price (single item) auction in Section 1.2.2 provides a simple example of such an auction.

Alternatively, one can modify the dynamics of the auction to ensure termination with UCE prices, from which VCG payments can then be determined via the adjustment defined in Eq. (1.32). Thus the idea is slightly different: the price dynamics can overshoot the minimal CE price vector but the auction can nevertheless terminate with the VCG payments by computing discounts from the final prices.

To illustrate this approach we describe the class of *uQCE-invariant* auctions [MP07], which maintain *universal quasi-CE prices* (uQCE) in every round. Informally, prices q^t are uQCE when demand is at least supply in the main economy and also in each of the marginal economies. uQCE-invariant auctions work as follows:

³³First one formulates a linear program for the allocation problem. An auction is then interpreted as a primal-dual or subgradient algorithm. The auction maintains a feasible primal and dual solution in each round: the allocation, and the current prices. Prices are increased until termination conditions are satisfied, which establish complementary slackness conditions and demonstrate that the primal and dual solutions are optimal. On termination, the primal solution defines an efficient allocation and the dual solution defines CE prices.

- (i) In round t :
 - (a) collect demand sets at prices q^t
 - (b) if q^t are UCE, then stop
 - (c) else, select some set of buyers $U^t \subseteq B^+(q^t)$, from the set of buyers still bidding, that will see price increases
 - (d) $q_i^{t+1}(S) := q_i^t(S) + \epsilon$ for all $i \in U^t$, and all $S \in D_i(q^t)$
- (ii) On termination in round T ,
 - (a) implement the final allocation
 - (b) collect payments $q_i^T(S_i) - [\Pi^s(q) - \Pi^s(q_{-i})]$ from each agent.

Proposition 1.35 [MP07] *uQCE-invariant auctions terminate with UCE prices, and thus the VCG outcome, when agents follow straightforward strategies.*

Together with appropriate activity rules this makes straightforward bidding an ex post NE of any uQCE-invariant auction. Activity rules must be defined to restrict the feasible strategy space to that which supports straightforward bidding (for some, perhaps untruthful type $\hat{\theta}_i \neq \theta_i$) but no other strategies.

A canonical example is given by the subgradient-based adjustment dynamics of *iBundle, Extend and Adjust* [MP07] (*iBEA*), which builds on the *iBundle* [PU00] auction. In *iBEA*, the auction chooses in each round a “pivot” marginal economy (or main economy) that is not yet in CE. The WDP for this economy is solved, and the adjusted bidders defined as the losing bidders that are still bidding. The auction terminates when there are no pivot economies, with the adjusted price giving the VCG outcome.

Example 1.36 [MP07] As an example we illustrate the progress of *iBEA* on the CA problem in Table 1.4. In Table 1.5, each row provides the prices on each bundle to each buyer, and the seller revenue in the main economy and in each marginal economy. The bid of each buyer is indicated with parentheses. Comments in each round indicate which allocation is selected to solve the WDP. The main economy $E(N)$ is adopted as the initial pivot economy, and retained as the pivot economy until round 7 at which point the prices are CE for $E(N)$. They are also CE for $E(\{1, 3\})$ and $E(\{1, 2\})$ in this round. So, pivot economy $E(\{2, 3\})$ is adopted for the final two rounds, at which point *iBEA* terminates with a UCE price vector.

One might wonder whether non-anonymous and non-linear prices are necessary to implement an efficient allocation. Blumrosen and Nisan [BN05]

Values \rightarrow	Buyer 1		Buyer 2		Buyer 3			Revenue in all economies		
	{1}	{2}	{1,2}	{1}	{2}	{1,2}	{1}		{2}	{1,2}
	3	0	3	0	6	6	0	2	4	
1	(0)	0	(0)	0	(0)	(0)	0	0	(0)	{0,0,0,0}
	Pivot: $E(N)$. WD selects $\{\{1\}, \{2\}, \emptyset\}$.									
2	(0)	0	(0)	0	(0)	(0)	0	0	(1)	{1,1,1,0}
	Pivot: $E(N)$. WD selects $\{\emptyset, \emptyset, \{1, 2\}\}$.									
3	(1)	0	(1)	0	(1)	(1)	0	0	(1)	{2,1,1,2}
	Pivot: $E(N)$. WD selects $\{\{1\}, \{2\}, \emptyset\}$.									
4	(1)	0	(1)	0	(1)	(1)	0	(0)	(2)	{2,2,2,2}
	Pivot: $E(N)$. WD selects $\{\{1\}, \{2\}, \emptyset\}$.									
5	(1)	0	(1)	0	(1)	(1)	0	(1)	(3)	{3,3,3,2}
	Pivot: $E(N)$. WD selects $\{\emptyset, \emptyset, \{1, 2\}\}$.									
6	(2)	0	(2)	0	(2)	(2)	(0)	(1)	(3)	{4,3,3,4}
	Pivot: $E(N)$. WD selects $\{\{1\}, \{2\}, \emptyset\}$.									
7	(2)	0	(2)	0	(2)	(2)	(0)	(2)	(4)	{4,4,4,4}
	CEs of economies $E(N)$, $E(N_{-2})$, and $E(N_{-3})$ are reached. Note: Buyer 3 also demands \emptyset from this round onwards. $\{\{1\}, \{2\}, \emptyset\}$ is an efficient allocation of $E(N)$. Pivot: $E(N_{-1})$. WD selects $\{\emptyset, \emptyset, \{1, 2\}\}$. Buyer $\{2\}$ is unsatisfied.									
8	(2)	0	(2)	0	(3)	(3)	(0)	(2)	(4)	{5,4,4,5}
	Pivot: $E(N_{-1})$. WD selects $\{\emptyset, \emptyset, \{1, 2\}\}$. Buyer $\{2\}$ is unsatisfied.									
9	(2)	0	(2)	0	(4)	(4)	(0)	(2)	(4)	{6,4,4,6}
	An UCE price vector is reached. Final allocation: $\{\{1\}, \{2\}, \emptyset\}$; Final payment: $(0, 2, 0)$.									

Table 1.5. Progress of iBEA for an example

show that they are necessary for general CA instances, although there are classes of problems (such as additive valuations) for which an auction such as iBEA needs an exponential price space while auctions with linear prices ($q(S) = \sum_{j \in S} q_j$) need only m prices. In addition, Mishra and Parkes [MP07] show that non-anonymous and non-linear prices are required for “natural” ascending auctions even for substitutes valuations. See also Gul and Stacchetti [GS00].

1.4.3 Price-Based Elicitation via Learning Theory

Elicitation for CAs can also be addressed by drawing an analogy with methods in computational learning theory (CLT) [ZBS03, BJSZ04].

We will consider a general method to convert learning algorithms into *elicitation* algorithms [LP04]. Whereas learning can be posed as the problem of determining the *exact* valuation function of an agent, the problem of elicitation is different. The goal in elicitation is to learn just enough about the valuations of agents to be able to determine the efficient allocation, and (in order to provide incentive properties) the VCG payments.

The use of learning algorithms provides another benefit in application to elicitation, and even if exact valuations are eventually learned. This is because learning algorithms work with simple “oracle” models of the questions that an agent can answer and do not assume that an agent already knows how to represent its valuation in a given representation class, such as a bidding language. The relevant measure of this is the “query complexity,” which is often stated in terms of the *minimal* representation of a valuation function, given a language.

We first summarize the model of *exact query learning*. In exact query learning from membership and equivalence queries, the goal is to identify an unknown target function $h : X \rightarrow Y$, from some class, via queries to an oracle. A representation class C is adopted to encode the functions. For instance, the function could be a monotone Boolean function and the representation class the monotone DNF formulae.

The two kinds of queries in the learning context are:

- *Membership query.* The learner presents some x and the oracle replies with $h(x)$.
- *Equivalence query.* The learner presents its current estimate, the *manifest hypothesis* \tilde{h} and the oracle either replies ‘YES’ if $\tilde{h} = h$, or returns a counterexample x such that $\tilde{h}(x) \neq h(x)$.

Definition 1.37 (efficient learnability) Representation class C can be polynomial-query exactly learnable from membership and equivalence queries if there is an algorithm that can determine a representation $\tilde{h} \in C$, for which $\tilde{h}(x) = h(x)$ for all $x \in X$, in a number of queries that is polynomial in $m = \dim(X)$ and $\text{size}(h)$, which is the minimal size of h in representation class C .

Thus, a class can be efficiently learned if, for all possible functions that can be represented in the class, there is an algorithm that asks a small (i.e.

Learning	Elicitation
Function class C e.g. <i>monotone Boolean functions</i>	Valuation classes, V_1, \dots, V_n e.g. <i>free disposal</i>
Representation class C e.g. <i>monotone DNF formulae</i>	Bidding languages L_1, \dots, L_n e.g. <i>XOR bids</i>
Target function $h : X \rightarrow Y$ boolean X , $\dim(X) = m$ boolean or real Y	Valuations $v_i : 2^G \rightarrow \mathbb{R}_{\geq 0}$ m goods
Membership query query: x , resp.: $h(x) \in Y$	Value query query: $S \in 2^G$, resp.: $v_i(S) \in \mathbb{R}_{\geq 0}$
Equivalence query query: hypothesis \tilde{h} resp.: ‘YES,’ if $\tilde{h}(x) = h(x)$, $\forall x \in X$ else, some x' s.t. $\tilde{h}(x') \neq h(x')$	Demand query query: $S \in 2^G$, prices $q_i : 2^G \rightarrow \mathbb{R}_{\geq 0}$ resp.: ‘YES,’ if $S_i \in \arg \max_{S \in 2^G} [v_i(S) - q_i(S)]$ else, some S' s.t. $v_i(S') - q_i(S') > v_i(S_i) - q_i(S_i)$

Table 1.6. *Computational Learning Theory vs. Elicitation*

polynomial) number of queries *in the size of the minimal possible representation of the function*. If the function is simple to represent then it should be simple to learn. Learning theory is interested in understanding the query complexity of representation classes and developing efficient algorithms.

We now compare query learning with that of elicitation. See Table 1.6. To enable as direct a comparison as possible we suppress the semantics of *type* and deal directly with *valuation classes*, which correspond to type classes, and valuation functions which correspond to valuations defined for some particular type.

In *efficient elicitation from value and demand queries*, the goal is to identify an efficient allocation for some class of valuations $v_i : 2^G \rightarrow \mathbb{R}_{\geq 0}$, via queries to agents. In place of a representation class is a bidding language, L , which describes an instance of the class.

The two kinds of queries in the elicitation context are:

- *Value query*. The elicitation algorithm presents some bundle S to an agent and the agent is asked to reply with its value $v_i(S)$.
- *Demand query*. The elicitation algorithm presents a bundle S_i and prices q_i to an agent and the agent is asked to reply ‘YES,’ if the bundle is in its demand set at the prices, or return a counterexample S' such that $v_i(S') - q_i(S') > v_i(S_i) - q_i(S_i)$ otherwise.

Definition 1.38 (efficient elicitation) The valuation classes V_1, \dots, V_n

are said to be *polynomial-query elicited* from value and demand queries if there is an algorithm that can determine an efficient allocation $S^* \in \arg \max_{S \in \mathbb{F}(G)} \sum_i v_i(S_i)$ for any $(v_1, \dots, v_n) \in V_1 \times \dots \times V_n$ in a number of queries that are polynomial in the number of goods m , the number of agents n , and $\max_i \text{size}(v_i)$, which is the maximum size across all agents of the *minimum representation* of each agent’s valuation function.

This definition differs from that for query learning in that the goal is to determine the efficient allocation, which need not require learning the exact valuation of every agent.

Membership queries are completely equivalent to value queries and equivalence queries can be simulated as demand queries. Consider a manifest hypothesis $\tilde{v}_i \in V_i$ for agent i ’s value, and bundle S_i that is in the demand set for an agent with valuation \tilde{v}_i at prices q_i . Then, if agent i responds with a preferred bundle S' when presented with demand query $\text{demand}(S_i, q_i)$ either $v_i(S_i) \neq \tilde{v}_i(S_i)$ or $v_i(S') \neq \tilde{v}(S')$. Value queries can then be issued to determine which of the two bundles S_i and S' is the counterexample.

Based on this observation, Lahaie and Parkes [LP04] show how to convert a learning algorithm to an elicitation algorithm.

Theorem 1.39 [LP04] *The efficient allocation can be determined in $\text{poly}(n, m, \max_i \text{size}(v_i))$ value and demand queries for valuation classes V_1, \dots, V_n , if they can each be polynomial-query exactly learned from membership and equivalence queries.*

The idea is to simulate a separate learning algorithm for each agent. We conceptualize this as occurring with an “proxy” for the agent. When the learning algorithm requires a membership query, this is immediately issued by the proxy as a value query to the agent. When the learning algorithm requires an equivalence query, the proxy waits to issue a demand query to the agent until CE prices have been updated. Only when *every* proxy (and its corresponding learning algorithm) is at the point where it requires the answer to an equivalence query are new CE prices computed. A provisional allocation and CE prices are computed based on the current manifest valuations, as known to each proxy. This allocation and price tuple is then issued to agents as *demand* queries. If all agents reply ‘Yes,’ we have found an efficient allocation and the algorithm terminates. Otherwise, it is guaranteed that one or more proxies will find a counterexample to their current manifest, and the relevant agent’s learning algorithm can proceed.

For equilibrium considerations we can again appeal to the VCG mechanism and to UCE prices. A straightforward strategy in the context of

query-based elicitation is one in which an agent responds truthfully to every query. For demand queries this means responding ‘YES’ if and only if the proposed bundle is in the true demand set, and providing a preferred bundle otherwise. Lahaie et al. [LCP05] extend demand queries to *universal demand queries*,³⁴ and generalize the elicitation algorithm to terminate with UCE prices and adjust to VCG payments. This brings the straightforward strategy into an ex post Nash equilibrium.

These query-learning based methods can be applied to the following representation classes for CAs:

- (i) *Polynomial bidding language*. For instance $v_i(S) = a_0x_1 + a_1(x_1x_3) - a_2(x_1x_5)$, where $x_j \in \{0, 1\}$ indicates if good j is allocated to the agent. These are concise for valuations that are almost substitutes and fully expressive. A learning algorithm due to Schapire and Sellie [SS93] leads to elicitation with $O(nmt)$ demand queries and $O(nmt^3)$ value queries, where t is the maximal size of the (minimally represented) valuation of each agent for the particular instance under consideration.
- (ii) *XOR bidding language*. The XOR language is concise for valuations that are almost complements and fully expressive. A generalization of a learning algorithm due to Angluin [Ang87] for monotone DNF formulae leads to elicitation in $O(nt)$ demand queries and $O(nmt)$ value queries where t is the maximal size of the (minimally represented) valuation of each agent for the particular instance under consideration.
- (iii) *Atomic languages*. The atomic languages generalize the XOR language, for example defining a language that is concise for additive valuations where XOR is not. Lahaie et al. [LCP05] provide a learning algorithm for this setting, and support elicitation in $O(nmt)$ demand queries and $O(nt)$ value queries, where t is the maximal size of the (minimally represented) valuation of each agent for the particular instance under consideration.

It is interesting that an exponential number of demand queries are required to learn an XOR representation if queries are restricted to linear prices [BJSZ04]; thus, we again see the power of non-linear price queries.

³⁴A universal demand query presents an agent with prices p and n bundles, representing the bundle allocated to agent i in the main economy and in each of the marginal economies in which it is present. An agent is asked to respond ‘YES,’ if every bundle is in its true demand set and otherwise provide a preferred bundle.

1.5 Distributing the Computation to Agents

In a direct mechanism, the agents report their types and then wait for the outcome and payments to be determined by the center. In an indirect mechanism, such as an ascending price auction, the agents are also performing useful computation by computing their demand sets in response to the current prices. *Distributed algorithmic mechanism design* (DAMD), as introduced by Feigenbaum and colleagues [FPSS02, FS02], and extended to *Distributed Implementation* (DI) by Parkes and Shneidman [PS04b, SP04b] to better encompass equilibrium considerations, pushes this distribution one step further.³⁵ The ultimate goal is to be able to completely remove the center and have the agents determine the outcome of a mechanism amongst themselves, by distributing the required computation.

For instance, we might ask agents to communicate partial information about their types with each other, and structure the problem solving amongst themselves, before coming to a shared consensus about the appropriate decision and payments. In addition to the benefits of robustness and scalability that can accrue from distributed computation, this distribution may be necessary, for instance in an Internet-scale application in which the dynamics and scale preclude one center having a full view of the entire state of the system.

Because the same self-interested agents that care about the outcome of the mechanism are now also involved in the computation, this distribution necessitates expanding the strategic considerations from those of incentive-compatibility (with its focus on truthful information revelation) to also include *computation-* and *communication* compatibility so that agents will choose to *faithfully* follow the rules of the mechanism. What is possible will also be constrained by physical considerations, for instance by the network communication topology.

The primary objectives in DI are to couple good communication and computational properties with faithfulness, so that the computation and message passing actions required of nodes forms a game-theoretic equilibrium. The earlier examples of ascending price auctions already provide a simple version of this: the agents choose, in equilibrium, to determine their correct demand sets in response to the prices generated by the auction in each round. DI equates a distributed algorithm with an agent strategy and the

³⁵*Distributed games* [MT99] are also relevant, which provide a formalism to study the effect of communication structures on the problem of implementing social choice functions in multi-agent systems. Relevant work in distributed artificial intelligence (DAI), considers algorithms for distributed constrained optimization (DCOP) in message-passing computational architectures [ML04, MSTY05, PF05], but without considering incentive issues.

algorithm must, itself, form an equilibrium. See Feigenbaum et al. [FSS07] for a comprehensive survey.

1.5.1 DI: Preliminaries

In full generality, DI need not have a dedicated center. The traditional center’s duties of deciding an outcome and payments are still done, just not by a dedicated, trusted mechanism-designer introduced node. However, it is most convenient in the current treatment to retain the notion of a center, although it will play a much smaller role than in standard MD.³⁶

An instance of DI, $d_M = (\Sigma, g, p; s^m)$, is defined in terms of a *strategy space* $\Sigma = (\Sigma_1 \times \dots \times \Sigma_n)$, an *outcome rule* $g : \Sigma \rightarrow A$, a *payment rule* $p : \Sigma \rightarrow \mathbb{R}^n$ and an *intended strategy* (or *intended algorithm*) $s^m = (s_1^m, \dots, s_n^m)$ where $s_i^m : \Theta_i \rightarrow \Sigma_i$.

Formally, this definition augments that of an *indirect mechanism* (see Section 1.2.2) with the intended strategy, s^m . This strategy represents the algorithm “intended” by the designer, i.e. that which the designer wishes the agents to follow. But, hidden under the covers are some deeper differences:

- (i) In DI, some agents may not be able to send messages directly to the center and agents may be able to send messages directly to other agents. This is captured within the strategy space Σ .
- (ii) In DI, an agent’s strategy $s^m(\theta_i) \in \Sigma_i$ can define messages that it will send to one or more agents in addition to the center, and messages that it will forward on behalf of other agents.
- (iii) In DI, an agent’s strategy $s^m(\theta_i) \in \Sigma_i$ can define computation that it will perform in response to messages received.
- (iv) In DI, the center (if one exists) need not receive enough information about the types of agents to be able to compute for itself the outcome and payments.

DI is best conceptualized as a distributed and asynchronous message-passing algorithm on a communication graph with agents located on nodes of the graph. The center is just viewed as another (trusted) node. There can be limited connectivity between agents, and between agents and the center. Rules g and p define the outcome and payments selected based on messages

³⁶The subset of messages that are communicated between agents and the center ultimately determine the outcome. A fully distributed implementation would terminate with *shared state* that is sufficient to define the outcome and also be self-enforcing, in that agents would not deviate from the outcome, once computed. We are not aware of *completely* distributed implementations at this time; for instance (at least) a trusted bank, able to receive messages from any agent and collect payments, seems to be required in current work.

that are received by the center. The computational agents asked to follow the algorithm are self-interested and will deviate from the algorithm if this is in their best-interest.

Agents have internal, computational actions, and external actions in which they send (private) messages to other agents with which they are connected. The feasible strategy space Σ_i constrains the external actions, in that the center, and other agents when following the intended algorithm, will only respond to “legal” messages. The expanded strategy space can allow an agent to misrepresent the reported type of another agent when used as a relaying node, or perform computation that affects the outcome of the mechanism.

Given that activities beyond information-revelation are now allowed we extend the notion of IC from DRMs, and introduce the following solution concept, which is central to DI:

Definition 1.40 Distributed implementation d_M is *ex post faithful* if intended strategy s^m is an *ex post* Nash equilibrium for all types $\theta \in \Theta$.

Faithfulness can be defined for game-theoretic solution concepts other than *ex post* NE. However, *ex post* NE seems especially useful and is adopted in current work on DI.

For analysis, it is useful to logically partition the actions performed by the intended strategy, s_i^m , and thus the strategy itself, into three different components. For a strategy to be partitioned into different components means that in each information state, exactly one of the component strategies is defined, and consulted to determine the action taken by the agent. Let $s_{-i}^m(\theta_{-i}) = (s_1^m(\theta_1), \dots, s_{i-1}^m(\theta_{i-1}), s_{i+1}^m(\theta_{i+1}), \dots, s_n^m(\theta_n))$. The actions are divided into the following components:

(r_i^m) *information-revelation actions*. Let r_i^m denote the component of strategy s_i^m that is equivalent, in equilibrium, to reporting information about an agent’s type. Formally, for all strategies \hat{s}_i that differ from s_i^m only in component r_i^m , and for all $\theta_{-i} \in \Theta_{-i}$, there exists some $\hat{\theta}_i \in \Theta_i$, for which:

$$g(\hat{s}_i, s_{-i}^m(\theta_{-i})) = g(s_i^m(\hat{\theta}_i), s_{-i}^m(\theta_{-i})) \quad (1.33)$$

$$p(\hat{s}_i, s_{-i}^m(\theta_{-i})) = p(s_i^m(\hat{\theta}_i), s_{-i}^m(\theta_{-i})). \quad (1.34)$$

(p_i^m) *message passing actions*. Let p_i^m denote the component of strategy s_i^m that defines an action in which the agent sends a message received, unchanged, to one or more neighbors or the center.

(c_i^m) *computation actions*. Let c_i^m denote the component of strategy that

defines a computational action, which is an action that is not message passing and whose effect cannot be simulated by following the intended strategy for some alternate type $\hat{\theta}_i \neq \theta_i$.

By the definition of information-revelation actions, any strategy \hat{s}_i that *only* deviates from the intended strategy in information states in which strategy r_i^m is defined will have an effect that can also be achieved by following the intended strategy s_i^m but for some type $\hat{\theta}_i \in \Theta_i$, possibly untruthful. This is the sense in which this component of the strategy is the *information-revelation* component.³⁷ On the other hand, in an information state in which the computational component c_i^m of the intended strategy is defined, the agent can have the effect of misrepresenting the information reported by another agent, or changing the definition of the outcome or payment rules of the mechanism, by deviating to another action. This aspect of an agent's strategy, as well as that of message-passing, is new to DI and can present new opportunities for strategic behavior.

Given this decomposition into different strategy components, faithfulness can be usefully decomposed into three corresponding components:

- (IC) *Incentive-compatible.* A DI, d_M , is incentive-compatible (IC) if every agent will respond to all of its intended information-revelation actions r_i^m in an ex post NE, for all type profiles $\theta \in \Theta$.
- (CC) *Communication-compatible.* A DI, d_M , is communication-compatible (CC) if every agent will perform all of its intended message passing actions p_i^m in an ex post NE, for all type profiles $\theta \in \Theta$.
- (AC) *Algorithm-compatible.* A DI, d_M , is algorithm-compatible (AC) if every agent will perform all of its intended computational actions c_i^m in an ex post NE, for all type profiles $\theta \in \Theta$.

This makes clear that faithfulness extends the more traditional concern of truthful information revelation to other kinds of deviations. Note that it is without loss of generality to assume that the intended information-revelation actions are truth-revealing, and so the property of *IC* will be equated with a strategy in which agents respond truthfully to queries, from other agents or from the center, about their type. We have the following, which follows immediately from the way in which IC, CC and AC have been defined:

Proposition 1.41 [SP04b] *A distributed implementation is ex post faithful*

³⁷For example, the feasible strategy space in the uQCE-invariant auctions defined in Section 1.4.2 *only* includes information-revelation actions, since all deviations from straightforward bidding select the outcome of the VCG mechanism for some (possibly untruthful) valuation.

when the intended strategy is IC, CC and AC in the same ex post Nash equilibrium.

1.5.2 Extended Example: Second-price Auction

The following example will serve to illustrate the definitions introduced above as well as suggest some of the challenges in designing faithful protocols. Consider a *distributed second-price auction* $d_M = \langle \Sigma, g, p; s^m \rangle$ for a single item in which there is a communication graph structured as a tree rooted at the center. One agent is associated with each node of the tree. Let θ_i denote agent i 's value. Agents communicate with messages $w = (v_1, v_2, x)$ where v_1 and v_2 are the first- and second highest value the agent has seen (including its own), and x is the identity of the agent with the highest value. Use “upstream” and “downstream” to denote towards and away from the root, respectively. The intended strategy, s^m , has the following components:

- (i) *information-revelation*: Send message $(v_i, 0, i)$ to the upstream agent on receiving a *START* message.
- (ii) *computation*: Maintain internal state (v_1, v_2, x) , initialized to $v_1 = \theta_i$, $v_2 = 0$ and $x = i$. Upon receiving message (v'_1, v'_2, x') from a downstream agent, update $v_1 := \max(v_1, v'_1)$, $v_2 := \{v_1, v_2, v'_1, v'_2\}^{(2)}$ (i.e. the second-highest seen), and $x := x'$ if $v'_1 > v_1$. If the local state has changed, then send message (v_1, v_2, x) to the upstream neighbor.
- (iii) *message-passing*: Upon receiving a *START* message, send *START* to all downstream neighbors.

The outcome rule $g : \Sigma \rightarrow A$ defines the winner of the auction as the agent for whom the highest value is reported by the neighbors to the center. Assume that all messages take a finite time and the center knows the maximal number of agents and so knows when it can terminate the mechanism and make the final decision. The payment rule $p : \Sigma \rightarrow \mathbb{R}_{\geq 0}^n$ defines the payment by the winner as the second highest value received by the center (in either value entry of an incoming message), with all other agents making zero payment. The feasible strategy space limits agents to sending either the message *START* (as the first message sent to each downstream neighbor), or messages to upstream neighbors with syntax (v_1, v_2, x) where $v_1, v_2 \in \mathbb{R}_{\geq 0}$ and $x \in N$.

Example 1.42 For a concrete instance of this problem of DI see Figure 1.2. The mechanism is IC (for the simple reason that in equilibrium only the agents directly connected to the center will affect the outcome via their

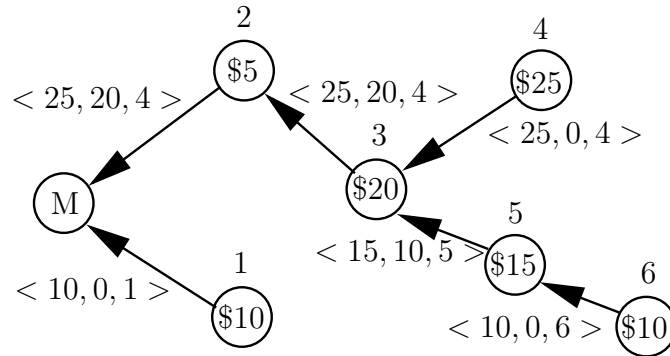


Fig. 1.2. An example of a second price distributed auction [not faithful].

report), but not AC or CC. AC fails. For instance, agent 3 in the figure should never propagate any value information that it receives from downstream agents (e.g. $(25, 0, 4)$ from agent 4). Rather it should deviate and claim that it has received no bids from downstream. CC fails. For instance, agent 3 should never propagate the *START* message to downstream agents because it would rather them not participate in the auction.

1.5.3 Analysis Techniques

With all these definitions in hand we are ready to develop some faithful DIs. It might be surprising that this is possible, given the apparent complexity of establishing that an intended algorithm is IC, CC, and AC. The analysis techniques introduced below will be subsequently illustrated in a sequence of case studies.

Strong Compatibility

It is of course not sufficient to assume AC and CC and then prove IC, and then assume AC and IC and prove CC, and finally assume CC and IC and prove AC. This is because agents can also jointly deviate from multiple components of their strategy at the same time.

Progress can be made by the following observation. We define strong versions of AC and CC, and then appeal to the dominant-strategy truthfulness of direct-revelation mechanisms to get faithfulness.

Definition 1.43 A distributed implementation is *strong CC* (resp. *strong AC*) if an agent will perform all of its intended message-passing actions p_i^m (resp. computational actions c_i^m) in the ex post NE of a restricted game in

which it is forced to deviate from its information-revelation actions r_i^m and computational actions c_i^m (resp. message-passing actions p_i^m) in an arbitrary way, for all types θ_i and all types θ_{-i} .

A DI d_M is said to be *equivalent* to DRM, $M' = \langle g', p' \rangle$, with g' and p' defined so that $g(s^m(\theta)) = g'(\theta)$ and $p(s^m(\theta)) = p'(\theta)$. Given this, we have the following:

Theorem 1.44 [SP04b] *A strong CC and strong AC distributed implementation that is equivalent to a DSIC mechanism M is faithful as long as the information-revelation actions are restricted to be consistent.*

This holds by an argument similar to that employed in the revelation principle. Once the new opportunities for manipulation provided to agents within a DI have been shown to be ineffective the faithfulness is inherited from the DSIC of the equivalent DRM.

Partition Principle

For illustrative purposes, consider the following “canonical distributed algorithm” to determine the outcome of a VCG mechanism. To keep things simple, it assumes that every agent can communicate directly with the center without sending messages through other agents.

- (step 1) Each agent reports its type $\hat{\theta}_i$ to the center.
- (step 2) The center involves some subset of the agents in computing a solution $g(\hat{\theta})$ and $g(\hat{\theta}_{-i})$ for each marginal economy $E(N \setminus i)$. This step can utilize a variety of distributed algorithms.
- (step 3) The center receives reported solutions $a' = g'(\hat{\theta})$, $a'_{-1} = g'(\hat{\theta}_{-1}), \dots, a'_{-n} = g'(\hat{\theta}_{-n})$ (where a' and g' are adopted to denote that they may not be correct).
- (step 4) These reports are used to define the outcome of the VCG mechanism:

- (i) Alternative a' is selected as the outcome.
- (ii) Payments $p_i(\theta) = \sum_{j \neq i} v_j(a'_{-i}; \hat{\theta}_j) - \sum_{j \neq i} v_j(a'; \hat{\theta}_j)$.

To see that agent i can usefully deviate from the intended strategy, notice that it can now affect the first component of its payment term, $p_i(\theta) = \sum_{j \neq i} v_j(a'_{-i}; \hat{\theta}_j)$, the payment which is usually independent of its strategy. Here, if agent i is involved in the computation of a'_{-i} then it should obstruct the computation of the best alternative in the marginal economy without i .

Theorem 1.45 [PS04b] *A canonical distributed VCG implementation with an intended strategy s^m in which computation is partitioned so that no deviation from agent i can affect the correct solution of $g(\theta_{-i})$ is faithful.*

A canonical distributed VCG implementation that satisfies this *partition principle* has the property that deviating from the intended computational actions has no more effect than deviating from truthful revelation. Just as it is in an agent’s best interest to report its true type, it is an agent’s best interest to assist in computing the best alternative given the reports of all agents. An agent is indifferent to performing the computation of the solutions to marginal economies for other agents, because this just affects the payment of other agents. Computation, when assigned carefully, becomes no more useful than revelation.

Example 1.46 Consider the following simple example. This is for a single-minded CA, as introduced in Section 1.3.2. Each agent is interested in a single bundle of goods $G = \{A, B, C\}$, with types: $\theta_1 = \{(ABC, 14)\}$, $\theta_2 = \{(B, 8)\}$, $\theta_3 = \{(AC, 12)\}$. A valid assignment of agents to perform computation (that will give faithfulness) is: agent 1 for the problem without agent 3, agents 2 and 3 for the problem without agent 1, and agents 2 and 3 for the entire problem.

This principle can also be used in the context of blackboard-based algorithms, where the blackboard is “moderated” by the center, and only agents other than i are allowed to post improvements to the solution $g(\theta_{-i})$ [PS04b].

Information-Revelation Principle

A second principle is the *information-revelation* principle. This is a corollary of Theorem 1.44, but worth stating explicitly:

Corollary 1.47 *A distributed implementation that is equivalent to a DSIC mechanism M is faithful if the only actions available in the feasible strategy space are information-revelation actions, and as long as the information-revelation must be consistent.*

This explains, for instance, why ascending price CAs that terminate with the VCG outcome have straightforward bidding in an ex post NE. Similarly, distributed optimization algorithms from operations research such as Dantzig-Wolfe, Bender’s, and methods such as column generation and Lagrangian relaxation [BHM77, Geo70] involve only information-revelation and provide faithful DIs for the VCG mechanism.

Redundancy Principle

A third principle is the *redundancy principle*. This supposes that the computation required in determining g and p in mechanism $M = \langle g, p \rangle$ can be divided into a sequence of finite computational step. The pieces can then be dispatched to agents in one of two ways:

- Each piece can be given to two agents, along with all necessary inputs to perform the computation. If both agents respond with the same result then that is adopted by the center and the next computational step is dispatched. Otherwise, the center steps in and repeats the computation itself and punishes one or both agents, whichever is found to have deviated (e.g. with a fine.)
- Each piece can be given to three or more agents, along with all necessary inputs to perform the computation. The mechanism then adopts the quorum outcome.

Collectively, we refer to this approach (which provides strong AC) as “chunk-and-dispatch”.

Theorem 1.48 [PS04b] *A distributed implementation d_M that is equivalent to a DSIC mechanism M is faithful if all computation is performed with redundancy via chunk-and-dispatch.*

Example 1.49 The redundancy principle can be used to structure distributed computation of tree search, in determining the outcome of winner determination in a CA. Pairs of agents can be used dynamically to search below a node in the search tree, e.g. performing a particular number of node expansions before passing the updated search frontier back to the center.

1.5.4 Case Study: Lowest Cost Routing

Feigenbaum et al. [FPSS02] (FPSS) study a distributed algorithm to compute VCG payments for lowest-cost inter-domain routing. For an example see Figure 1.3. Each node has an associated agent, and represents an autonomous system on the Internet. Agents have *transit costs* for forwarding messages to other nodes. In the example, node Z has transit cost 1000 for forwarding messages from A to C , or from B to C .

The social objective is to compute all pairwise lowest cost paths (LCPs). For instance, the LCP from X to Z is 2, and goes via D and C . The LCP from X to B is 0 and involves no transit nodes. This is a MD problem because the transit cost at each node is private to that node. This is,

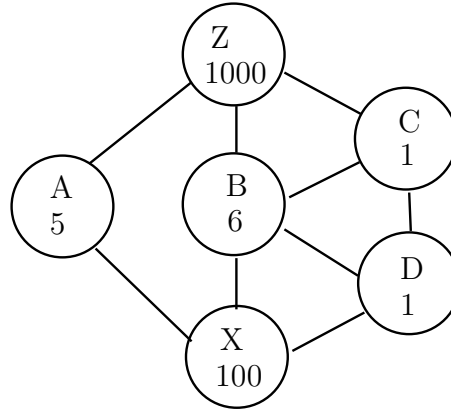


Fig. 1.3. An example of LCP inter-domain routing.

more broadly, a DAMD problem and one of DI, because there is no central computational node; rather, the algorithm for LCP must be distributed and nodes are self-interested.

FPSS propose a modification to the local state that each node maintains, and retain the basic message passing structure of BGP, extending the algorithm so that the algorithm computes VCG payments as well as LCPs. Realize that VCG payments will be made to transit nodes because they incur costs. VCG payments are made by nodes with node-originating traffic to compensate nodes that provide transit capacity. The graph is assumed to be biconnected so that VCG payments are well defined. Without this, the marginal positive effect on the cost of a path due to a node with a pivotal position in the network is unbounded, and its VCG payment would be unbounded.

In the example, the VCG payment made by node X for X -originating traffic with destination Z is to transit nodes D and C . Node D receives payment 5 and node C receives payment 5. In both cases, this represents the maximal cost that each node could have reported and still formed the LCP from X to Z .

However, the FPSS algorithm is not faithful. The most obvious way in which FPSS fails faithfulness is because the sender, X in our example, is ultimately responsible for computing *its own* payment. X would prefer to deviate from this computation. Even without this problem, nodes can benefit by overstating the transit cost of nodes on competing paths, to boost the payments they receive in the VCG scheme.

Shneidman and Parkes [SP04b] make FPSS faithful by splitting the pro-

to be divided into three phases (two construction and one execution, in which traffic flows and payments are made and collected), and using redundancy and the idea of “catch and punish” to make the implementation *strong AC* and *strong CC* within each phase, irrespective of behaviors in other phases. Faithfulness leverages the biconnectedness of the graph, so that no agent can unilaterally block information flow.

The most important idea is that of assigning *every neighbor of a node as a checker node*. For instance, nodes A, B and D act as checker nodes for node X (as well as principal nodes in their own right). The intended strategy is augmented so that the principal is required to relay all messages it receives to every checker node. CC is established for this aspect of the strategy because there is always at least one checker that knows about any message the principal receives (because it was the sender!), and thus a deviation is caught when the checkers compare local state at the end of a phase. All checker nodes replicate all computation of the principal node; again, this is shown to be AC through a catch-and-punish argument. A limited center is assumed to compare the states of checker nodes and impose penalties if a mismatch is discovered.

1.5.5 Case Study: Faithful Distributed Constrained Optimization

Petcu et al. [PFP06] have provided a faithful DI, called *M-DPOP*, for solving distributed constraint optimization problems (DCOP). In a DCOP, each agent is associated with a variable X_i with domain d_i and is responsible for setting the value of its variable. There are hard constraints that preclude certain values in combination for pairs of variables. Finally, there are *relations*, $R = \{r_1, \dots, r_n\}$ where r_i is the set of relations known to agent i and relation r_i^j is a function $d_1 \times \dots \times d_k \rightarrow \mathbb{R}$ that defines a value to agent i for each possible combination of values of the involved variables. The objective in DCOP is to find the optimal solution $x^* = \arg \max_{x \in \mathbb{F}} \sum_i \sum_{r \in r_i} r(x)$, where \mathbb{F} defines the set of feasible assignments given the hard constraints.

DCOP assumes a message passing architecture. Each agent is connected in the communication graph to all agents for which it has a relation with that agent’s value in its domain, or that shares a hard constraint with its variable. A number of optimal and complete (i.e. will solve all instances) DCOP algorithms are known, able to terminate with the value-maximizing assignment of values that satisfies all hard constraints [ML04, MSTY05, PF05]. However, none of these algorithms are faithful in the sense of DI.

M-DPOP extends the DPOP algorithm [PF05], making it faithful so that no agent can benefit from a unilateral deviation from the algorithm (as long

as the other agents follow the algorithm). In doing so, M-DPOP retains the useful computational properties of DPOP [PF05], including a linear number of messages, each of size that is exponential in a parameter that scales with the degree of interdependence between different parts of the problem.

M-DPOP is designed to terminate with the outcome of the VCG mechanism and is made faithful by the partition principle. Agent i is not able to prevent agents other than i from computing the correct solution to the DCOP problem without agent i . The other agents are simply instructed to ignore any messages that agent i sends during the process of solving the problem without i . In addition, agents other than i are responsible for finally computing the payment that agent i must make. This is achieved by ensuring that they have enough information locally to understand the marginal effect of agent i on their own value. M-DPOP is also able to re-use computation performed in solving the main problem when solving the problem without agent i , by carefully retaining a similar control flow in solving the problem without i and identifying which messages can be re-used.

1.5.6 Case Study: A Distributed Second Price Auction.

Consider again the earlier example of a distributed second price auction. Recall that the intended strategy was not faithful because an agent would choose *not* to propagate the values of agents further downstream from the center.

Monderer and Tennenholtz [MT99] (MT) study a special case of this problem, with message-passing and information-revelation actions but without computational actions. The basic function of agents is to propagate bids received from agents further downstream on, towards the center. The appropriate concerns, then, are those of IC and CC but the agents are not required to perform any computational actions as part of the intended strategy.

MT consider a *biconnected graph* so that a message sent by any agent i to be delivered to the center would still be received by the center even if one other agent deviates and chooses not to propagate the message. This is almost sufficient by itself because an agent can no longer benefit from dropping a message. However, this would not lead to the correct implementation because an agent that sees a bid from downstream with value greater than its own would stop forwarding messages from that round forwards (or at least be weakly indifferent.) Agents at the far edges of the network would be disadvantaged.

MT address this by masking the information so that an agent does not learn anything useful when forwarding the bid of another agent. For this,

MT assume that values are defined on a domain of size 2^k for some positive integer k , and with an agent values distributed uniformly on $\{1, \dots, 2^k\}$.

Let v_i denote a bit string representation of agent i 's value, i.e. $v_i \in \{0, 1\}^k$. Before sending its value, agent i selects a bit string $y_i \in \{0, 1\}^k$ uniformly at random. This acts to mask its value. The agent sends y_i on one of its outgoing edges and $v_i \oplus y_i$ (the bit-by-bit exclusive or of v_i and y_i) on the other. A forwarding agent cannot distinguish the mask y_i from the masked bid $v_i \oplus y_i$, and even if it could determine $v_i \oplus y_i$ would not learn anything about v_i because the posterior distribution on the masked bid is the same as the prior on value.

Taken together, this brings the intended strategy of correct forwarding of messages into an ex post Nash equilibrium. Each agent chooses to report its $v_i \oplus y_i$ and mask y_i truthfully, and the center finally combines this information, with $y_i \oplus (v_i \oplus y_i) = v_i$ and recovers the values and determines the Vickrey outcome.

1.6 Dynamic Environments: Online Mechanisms

Many multi-agent problem domains are inherently dynamic. Consider, for instance, multi-agent planning domains [BGT03], and resource allocation in grid computing, where jobs have state and require resources for some period of time [FK00]. Other compelling examples are the eBay marketplace, in which auctions open and close over time and the bidder population is dynamic, and the sponsored search auctions used by Google and Yahoo!, in which supply and demand is realized online. One can also think about a group of suppliers in long-term contracts with a car manufacturer; in this setting the private state of a supplier (and its value for different decisions) can change dynamically, e.g., perhaps its workers go on strike, part of its plant fails, or the price of electricity increases.

In each of these settings at least one of the following is true: agents are dynamically arriving or departing, or fixed but with each agent realizing new information about its local problem across time; or, there is uncertainty about the set of feasible decisions in the future. These dynamics present a new challenges when seeking to sustain good system-wide decisions in multi-agent systems with self-interested agents. For example, if the agent population is dynamically changing then simultaneous reports of type (as in the standard model of direct-revelation mechanisms) is not possible.

The general problem of designing mechanisms for dynamic environments considers a center that implements a sequence of decisions, and agents that

report (perhaps untruthfully) private type information and have values that pertain to a sequence of decisions.

The appropriate notion of direct revelation is that of a direct-revelation, online mechanism in which the strategy space allows an agent to make a claim about its private information in each period. For example, in the simple case that each agent “arrives” in some period, a direct-revelation online mechanism will allow an agent to make a claim about its valuation function for different decisions by sending a single message in some period.

There are two main frameworks in which to study the performance of online mechanisms. The first is *model-free* [LN00], and adopts a worst-case analysis and is useful when a designer does not have good probabilistic information about future agent types or about feasible decisions in future periods. The second is *model-based* [FP03, PS03], and adopts an average-case analysis. As a motivating example, consider a search engine selling search terms to advertisers. This is a data rich environment and it is reasonable to believe that the seller can build an accurate model to predict the distribution on types of buyers, including the process governing arrival and departures.

We will consider each of these paradigms in turn. The mechanisms studied in the first framework will provide DSIC but require a restricted environment (single-valued). The mechanisms studied in the second framework will provide BNIC but apply to a more general environment. See Parkes [Par07] for a comprehensive survey.

1.6.1 Example: Dynamic Auction with Expiring Items

Consider a dynamic auction model with discrete time periods $T = \{1, 2, \dots\}$ and a single indivisible item to allocate in each time period. The type of an agent $i \in \{1, \dots, N\}$ is denoted $\theta_i = (a_i, d_i, w_i) \in T \times T \times \mathbb{R}_{>0}$. Agent i has arrival time a_i , departure time d_i , value w_i for an allocation of a single unit of the item in some period $t \in [a_i, d_i]$, and wants at most one unit. An agent has zero value if allocated an item outside of this period. This information is all private to an agent. Moreover, suppose that an agent does not *know* about its type until period a_i and thus cannot report an arrival time, \hat{a}_i , that is any earlier than this period. A payment can be collected from an agent in any period $t \in [a_i, d_i]$, including a period after an item is allocated [HKP04, HKMP05].

The following example illustrates why online auctions introduce new strategic considerations into MD.

Example 1.50 Suppose there are three agents, with types $\theta_1 =$

$(1, 2, 100)$, $\theta_2 = (1, 2, 80)$ and $\theta_3 = (2, 2, 60)$ and a seller that has a single unit of an item to sell in each of periods 1 and 2. Consider an online variant of the Vickrey auction: an agent can report its type in any period, and its bid will be considered in each of a sequence of second-price auctions until it wins or until its departure period, whichever occurs first. In the example, if the agents are truthful, then agent 1 wins in period 1 for 80, stops bidding, and agent 2 wins in period 2 for 60. But agent 1 can do better. It can report type $\hat{\theta}_1 = (1, 2, 61)$, so that agent 2 wins in period 1 for 61, stops bidding, and then agent 1 wins for 60 in period 2. Or, agent 1 can report type $\hat{\theta}_1 = (2, 2, 80)$ and delay its bid until period 2, so that agent 2 wins for 0 in period 1, stops bidding, and then agent 1 wins for 60 in period 2.

In this context, a direct-revelation online mechanism, $M = \langle g, p \rangle$, has an outcome rule $g : \Theta \rightarrow K$ and payment rule $p : \Theta \rightarrow \mathbb{R}^n$. Each agent can send a message in a single period, to make a claim about its type. We adopt notation K for the space of alternatives, instead of A , to avoid confusion with the arrival time, a_i .

Outcome $k \in K$ defines a *sequence* of allocations k^1, \dots, k^t, \dots for $t \in T$ where (in this environment) $k^t \subseteq N$ defines the winner(s) in period t . The outcome rule and payment rule must be *online implementable*, defined as follows: Let $g(\theta, t) = k^{[1..t]}$, i.e. the sequence of decisions made between periods 1 and t . Then, we must have $g(\theta, t) = z$ and $g(\hat{\theta}, t) = \hat{z} \neq z$ implies that $\hat{\theta}_i = (\hat{a}_i, \hat{d}_i, \hat{w}_i) \neq \theta_i$ for some agent i with $\hat{a}_i \leq t$. Similarly, for the payment rule, we need $p_i(\theta) = \pi$ and $p_i(\hat{\theta}) = \hat{\pi} \neq \pi$ implies that $\hat{\theta}_i \neq \theta_i$ or $\hat{\theta}_j = (\hat{a}_j, \hat{d}_j, \hat{w}_j) \neq \theta_j$ for some $j \neq i$ with $\hat{a}_j \leq d_i$.³⁸ We also write $g_i(\theta) \in \{0, 1\}$ to denote whether or not agent i is allocated an item in some period $t \in [a_i, d_i]$ by outcome rule g .

A Characterization of Truthful Auctions An online mechanism is DSIC (or *truthful*) if an agent's dominant strategy is to report its true type, whatever the future reports of other agents. For this setting in which type information is simple, with an arrival, departure and value for an allocation decision, we can define an appropriate form of monotonicity:

Definition 1.51 [HKMP05] Outcome rule $g : \Theta \rightarrow \{0, 1\}^n$ is *monotonic* if for every agent i and every $\theta, \theta' \in \Theta$ with $[a'_i, d'_i] \subseteq [a_i, d_i]$, $w_i > w'_i$, and $\theta_{-i} = \theta'_{-i}$, we have $g_i(\theta) > g_i(\theta')$.

³⁸By this definition, if a payment rule is online implementable then it will also have a fixed payment to agent i for all periods t after the reported departure of the agent, and thus satisfies the requirement that payments are collected no later than an agent's departure.

In words, if an agent wins for some report (a_i, d_i, w_i) then it should continue to win for a more relaxed arrival-departure interval, and a higher value.

Given an outcome rule, we can define the *critical value* to agent i for arrival and departure (a_i, d) :

$$v_{(a_i, d_i)}^c(\theta_{-i}) = \begin{cases} \min w'_i & \text{s.t. } g_i((a_i, d_i, w'_i), \theta_{-i}) = 1 \\ \infty, & \text{if no such } w'_i \text{ exists} \end{cases} \quad (1.35)$$

When the outcome rule is monotonic, the critical value to agent i is independent of value w_i and (weakly) monotonically increasing in tighter arrival-departure intervals. In this domain, any truthful (and deterministic) online mechanism that satisfies IR must collect a payment equal to the critical value from each allocated agent. Monotonicity is also necessary; any truthful online mechanism that does not pay unallocated agents must be monotonic [HKMP05, Par07].

For simplicity we will now make an additional assumption, that of *no late-departure* misreports. Together with no late-arrival misreports, we have that agent reports $a_i \leq \hat{a}_i \leq \hat{d}_i \leq d_i$. For example, if this is an auction for theater tickets, then we could argue that it is not credible to claim to have value for a ticket for a last minute Broadway show after 5pm because the auctioneer knows that it takes at least 2 hours to get to the theater and the show starts at 7pm. For network resources, such as an auction for access to WiFi bandwidth in a coffee house, think about requiring a user to be present for the entire period of time reported to the mechanism.³⁹

We have the following positive result in this environment:

Theorem 1.52 [HKMP05] *Online mechanism $\langle g, p \rangle$ is truthful with agents that arrive, depart and have a value for a single unit of an item, when g is monotonic and p collects the critical-value payment from winners, and given no early-arrival and no late-departure misreports.*

The intuition for this result is that an agent cannot do any better by reporting a tighter arrival-departure interval because this will only make it less likely that it will be allocated, and increase its payment. Moreover, its payment (for a given arrival-departure) and contingent on winning is independent of its bid value and if the agent loses when bidding its true

³⁹The assumption of no late-departures can be dispensed with, while still retaining truthfulness, in environments in which it is possible to schedule a resource in some period before an agent's reported departure, but withhold access to the benefit from the use of the resource until the reported departure; e.g., in grid computing, jobs can run on the machine but the result then held until reported departure [Por04].

value then even though it might win by increasing its bid, its critical value payment will be greater than its value. For a proof, see Parkes [Par07].⁴⁰

Model-Free Analysis. In this setting we will adopt a model-free framework, and not assume any particular distributional information about the environment. The performance of a mechanism can instead be studied via a *worst-case analysis*, for a sequence of types that are generated by an “adversary” whose task it is to make the performance as bad as possible. Of particular relevance is the method of *competitive analysis*, typically adopted in the study of online algorithms. The following question is asked: *how effectively does the performance of the online mechanism “compete” with that of an offline mechanism that is given complete information about the future arrival of agent types?* Again, this is asked in the worst-case, for a suitably adversarially-defined input.

Competitive analysis is most easily justified when the designer does not have a good model of the environment. As a motivating example, consider selling a completely new product or service, for which it is not possible to conduct market research to get a good model of demand. Competitive analysis can also lead to mechanisms that enjoy good average-case performance in practice, provide insight into how to design robust mechanisms, and produce useful “lower-bound” analysis. A lower-bound for a problem makes a statement about the best possible performance that can be achieved by *any* mechanism. Online mechanisms are of special interest when their realized performance matches the lower bound.

For our adversarial model, we consider a powerful adversary that is able to pick arbitrary agent types, including the value, arrival and departure of agents. Let $z \in \mathcal{Z}$ denote the set of inputs available to the adversary and θ_z the corresponding type profile. An online mechanism is *c-competitive for efficiency* if:

$$\min_{z \in \mathcal{Z}} \mathbb{E} \left\{ \frac{\text{Val}(g(\theta_z))}{V^*(\theta_z)} \right\} \geq \frac{1}{c}, \quad (1.36)$$

for some constant $c \geq 1$. Here $\text{Val}(g(\theta_z))$ is the total value of the allocation determined by outcome rule g and $V^*(\theta_z)$ is the best possible solution, determined with perfect information about all of the types. Such a mechanism is guaranteed to achieve within fraction $\frac{1}{c}$ of the value of the optimal

⁴⁰The sufficiency result can also be generalized to a domain with arbitrary misreports of departure, by placing a timing constraint on the allocation [HKMP05]. It can also be extended to a more general valuation domain, in which agents have *single-valued* types, with the same value for any of some interesting set of decisions [Par07, PD07].

offline algorithm, whatever the input sequence. The expectation allows for a randomized outcome rule.

Back to the Expiring Items Setting Consider the following auction, proposed in Hajiaghayi et al. [HKMP05] (HKMP):

- (i) In each period, t , allocate the good to the highest unassigned bid.
- (ii) Agent i pays the critical value, which is the smallest amount that it could have bid and still won an item.

For impatient bidders with $d_i = a_i$ for all θ_i , this is precisely a sequence of Vickrey auctions and truthful in a dominant strategy equilibrium.

Theorem 1.53 [HKMP05] *The HKMP online auction is 2-competitive for efficiency and truthful in the expiring items environment and with no early-arrival and no late-departure misreports.*

Proof Suppose that random tie-breaking is invariant to reported arrival and departure. The auction is strongly truthful because the allocation function is monotone: if agent i wins in some period $t \in [a_i, d_i]$ then it continues to win either earlier or in the same period for $w'_i > w_i$, and for $a'_i < a_i$ or $d'_i > d_i$. For competitiveness, consider a set of types θ and establish that the greedy online allocation rule is 2-competitive by a “charging argument”. For any agent i that is allocated offline but not online, charge its value to the online agent that was allocated in period t in which agent i is allocated offline. Since agent i is not allocated online it is present in period t , and the greedy rule allocates to another agent in that period with at least as much value as agent i . For any agent i that is allocated offline and also online, charge its value to itself in the online solution. Each agent that is allocated in the online solution is charged at most twice, and in all cases for a value less than or equal to its own value. Therefore the optimal offline value $V^*(\theta)$ is at most twice the value of the greedy solution. \square

There is actually a 1.618-competitive online algorithm for this problem but it is not monotonic and cannot be implemented truthfully. In fact, there is a matching lower bound for the problem of achieving efficiency and truthfulness:

Theorem 1.54 *No truthful, IR and deterministic online mechanism can obtain a $(2 - \epsilon)$ -approximation for efficiency in the expiring items environment with no early-arrival and no late-departure misreports, for any constant $\epsilon > 0$.*

For the proof of this result see Hajiaghayi et al. [HKMP05]. More than just this gap, between what is possible with and without incentive constraints, Lavi and Nisan [LN05] also show that *no* constant-competitive mechanism is possible in this environment without the assumption of no-late-departures. Thus, we have another justification for this assumption.

Example 1.55 Consider the earlier example, with three agents and types $\theta_1 = (1, 2, 100)$, $\theta_2 = (1, 2, 80)$ and $\theta_3 = (2, 2, 60)$ and one item to sell in each period. Suppose all three agents bid truthfully. The greedy allocation rule sells to agent 1 in period 1 and then agent 2 in period 2. Agent 1’s payment is 60 because this is the critical value for arrival-departure $(1, 2)$ given the bids of other agents. (A bid of just above 60 would allow the agent to win, albeit in period 2 instead of period 1.) Agent 2’s payment is also 60.

Example 1.56 In order to understand the distinction between DSIC and ex post Nash IC, suppose that the reports received by the mechanism are public. Now, agent 3 can condition its strategy on the report of agent 1. For instance, if agent 3’s strategy is “bid $(2, 2, 1000)$ if a bid of $(1, 2, 100)$ is received, and bid $(2, 2, 60)$ otherwise” then agent 1 would pay 80 if truthful but 60 with a bid of $(2, 2, 65)$.

Related work considers an environment in which there is a fixed number of non-expiring items to allocate before some deadline [LN00, HKP04]. Awerbuch et al. [AAM03] provide a method to convert a competitive online algorithm into a truthful and competitive online auction for algorithms that satisfy technical conditions that provide increasing prices. Bredin and Parkes [BP05] study the environment of online *double* auctions in which there are both buyers and sellers. Juda and Parkes [JP06] study a model related to eBay, and extend the framework to allow for sequential auctions with non-identical goods and buyers with general valuations.

1.6.2 General Environments: *BNIC*

In a general, dynamic valuation environment we allow agents with general valuation functions defined on sequences of decisions. To keep things simple, we will assume here that the *arrival* period, a_i , still models the period in which an agent learns its own type.⁴¹

An agent’s type, θ_i , defines its value $v_i(k; \theta_i)$ for a sequence k^1, k^2, \dots, k^t

⁴¹Even more generally, we can also consider agents (possibly persistent) that receive information that pertains to their own type over time [BV06, CPS06]. We make a brief comment about this model in closing this section.

of decisions $k \in K$ in each of T discrete time periods. An agent has both an arrival a_i and a departure d_i , and $v_i(k; \theta_i) \neq v_i(k'; \theta_i)$ implies that $k^t \neq k'^t$ for some $t \in [a_i, d_i]$.

A Markov decision process (MDP) provides a useful formalism for defining online mechanisms in model-based environments with general agent preferences [PS03]. We define an MDP model $(H, K, Prob, R)$ with the following components:

- State, $h^t = (\theta^1, \dots, \theta^t; k^1, \dots, k^{t-1})$, in period $t \in T$, is defined in terms of the reports that the mechanism has received up to and including this period, and the sequence of decisions it has made. Let H denote the set of states, and $K(h)$ denote the set of feasible decisions in each state with $K = \cup_{h \in H} K(h)$.
- $Prob(h^{t+1}|h^t, k^t)$ defines the probability of transition to state h^{t+1} given decision k^t in state h^t . A well-defined model requires $\sum_{h'} Prob(h'|h^t, k^t) = 1$ for all h^t, k^t . This transition models encapsulates both the stochastic arrival model of agents and the effect (deterministic or otherwise) of a decision on the state.
- $R(h^t, k^t) = \sum_{i \in I(h^t)} R_i(h^t, k^t)$, is the reward received by the policy for taking action k^t in state h^t . Here, $I(h^t)$ denotes the set of agents present in state h^t , and agent i has value $R_i(h^t, k^t) = v_i(k^{[1..t]}; \theta_i) - v_i(k^{[1..t-1]}; \theta_i)$, given type θ_i .⁴²

The Markov property requires that feasible decisions, transitions and rewards depend on previous states and actions only through the current state. It is achieved here, for example, by defining $h^t \in H^t = (\theta^1, \dots, \theta^t; \omega^1, \dots, \omega^t; k^1, \dots, k^{t-1})$, so that the state captures the complete history of types, stochastic events, and decisions. In practice a short summarization of state h^t is often sufficient to retain the Markov property.

An *optimal policy* $\pi^* : H^t \rightarrow K^t$ maximizes the MDP value $V^\pi(h^t) = E_\pi [R(h^t, \pi(h^t)) + R(h^{t+1}, \pi(h^{t+1})) + \dots + R(h^T, \pi(h^T))]$ in all states h^t .⁴³ The expectation here is taken with respect to the probabilistic transition model. The *optimal MDP value function*, V^* , which corresponds to the MDP value for the optimal policy, can be computed via the following *value*

⁴²The reward to agent i in period t is defined this way so that the cumulative component of reward R_i because of the presence of agent i with type θ_i , is $\sum_{t=1}^\tau R_i(h^t, k^t) = v_i(k^{[1..\tau]}; \theta_i)$, for all periods τ ; thus, $R_i(h^t, k^t)$ defines the total value obtained by agent i up to and including period t .

⁴³For infinite time horizons, a standard approach is to define a discount factor and maximize the expected discounted value of a policy.

iteration algorithm [Put94]: for time periods $t = T-1, T-2, \dots, 1$:

$$\forall h \in H^t \quad V^*(h) = \max_{k \in K^t(h)} [R(h, k) + \sum_{h' \in H^{t+1}} \text{Prob}(h'|h, k) V^*(h')], \quad (1.37)$$

where $V^*(h \in H^t) = \max_{k \in K^t(h)} R(h, k)$. This algorithm works backwards in time from the horizon and has time complexity polynomial in the size of the MDP and the time horizon T .

Given the optimal MDP value function, the optimal policy is derived as follows: for $t < T$, we have:

$$\pi^*(h \in H^t) = \arg \max_{k \in K^t(h)} [R(h, k) + \sum_{h' \in H^{t+1}} \text{Prob}(h'|h, k) V^*(h')], \quad (1.38)$$

and $\pi^*(h \in H^t) = \arg \max_{k \in K^t(h)} R(h, k)$.

What makes this a problem of mechanism design is because the MDP state is defined in terms of agent type information, which is private to agents. Thus, incentives must be provided for agents to report their private types to the center.

A direct-revelation online mechanism, $M = \langle \pi, p \rangle$, in this general setting, restricts each agent to making a single claim about its type, and defines *decision policy* $\pi = \{\pi^t\}^{t \in T}$ and *payment policy*, $p = \{p^t\}^{t \in T}$, where decision $\pi^t(h^t) \in K(h^t)$ is made in state h^t and payment $p_i^t(h^t) \in \mathbb{R}$ is collected from each agent $i \in I(h^t)$.

We define a dynamic VCG mechanism for this problem. We assume that the decisions and reports in previous periods $t' < t$ are all public in period t , although similar analysis holds without this. The center knows the probabilistic transition model (and this model is also common knowledge to agents) but the realization of types is private to agents.

Definition 1.57 [PS03] A *dynamic VCG mechanism* works for the finite time horizon online MD environment works as follows:

- (i) Each agent, i , reports its type $\hat{\theta}_i$ in some period $\hat{a}_i \geq a_i$.
- (ii) Implement optimal policy π^* , which maximizes the total expected value, assuming the current state as defined by agent reports is the true state.
- (iii) On reported departure, $t = \hat{d}_i$, collect payment

$$p_i^t(h^t) = v_i(\pi^*(\theta^{[1..t]}); \theta_i) - [V^*(h^{\hat{a}_i}) - V^*(h_{-i}^{\hat{a}_i})], \quad (1.39)$$

where $\pi^*(\theta^{[1..t]})$ denotes the sequence of decisions made up to and

including period t based on types $\theta^{[1..t]}$, and h_{-i}^t defines the (counterfactual) MDP state constructed to be equal to h^t but removing agent i 's type from the state. The payment is zero otherwise.

Agent i 's payment is its (ex post) value discounted by the value $(V^*(h^{\hat{a}_i}) - V^*(h_{-i}^{\hat{a}_i}))$, which is the expected marginal value it contributes to the system as estimated upon its arrival and based on its report. With this, the expected utility to agent i when reporting truthfully is equal to the expected marginal value that it contributes to the multi-agent system through its presence.

For incentive-compatibility, we need the technical property of *stalling*, which requires that the expected value of policy π^* cannot be improved (in expectation) by delaying the report of an agent.⁴⁴ In addition, we assume an independence property; namely, the probabilistic process defining the arrival of agents other than i is independent of whether or not agent i has arrived.

Theorem 1.58 [PS03] *The dynamic VCG mechanism, coupled with a policy that satisfies stalling, is Bayes-Nash incentive compatible (BNIC) and implements the expected-value maximizing policy, in a domain with no early-arrival misreports but arbitrary misreports of departure.*

To understand why the dynamic VCG mechanism is BNIC, consider the following expression, which is the expected utility (defined with respect to its information in period a_i) to agent i for report $\hat{\theta}_i$, and given that agents other than i are truthful. Let $c \geq 0$ denote the number of periods by which agent i misreports its arrival time. The expected utility is:

$$\mathbb{E}_{\pi^*} \{v_i(\pi^*(h^{a_i}); \theta_i) | \hat{\theta}_i\} \quad + \mathbb{E}_{\pi^*} \left\{ \sum_{t=a_i+c}^T R_{-i}(h^t, \pi^*(h^t)) \right\} \quad - \mathbb{E}_{\pi^*} \{V^*(h_{-i}^{a_i+c})\}$$

(A)
(B)
(C)

Here, $\mathbb{E}_{\pi} \{v_i(\pi(h); \theta_i)\}$ denotes the expected value to agent i with type θ_i for policy π executed from state h forward in time, and $R_{-i}(h^t, k^t)$ is the total value to all agents except i for decision k^t in state h^t .

Term (A) denotes the expected value to agent i given its misreport. Term (B) is the expected value to all other agents forward from reported arrival, $a_i + c$, given report $\hat{\theta}_i$ and optimal policy π^* . It corresponds to the expected value of terms $\{-v_i(\pi^*(\theta^{[1..d_i]}; \theta_i) + V^*(h^{\hat{a}_i}))\}$ in the payment. Term (A) + (B) is the expected value to all other agents, plus the expected *true* value to agent i given its misreport. Term (C) is the total expected value to other

⁴⁴This stalling property is typically reasonable, for example any optimal policy that is able to delay for itself any decisions that pertain to the value of an agent will automatically satisfy stalling.

agents forward from period $a_i + c$, but with agent i removed, and corresponds to the final term in the payment.

First, fix the reported arrival period. Now term (C) is agent-independent, as in the offline VCG mechanism, and agent i maximizes the sum of (A) + (B) in a Bayes-Nash equilibrium by reporting its true valuation function $v_i(k; \theta_i)$ because the policy π^* is defined so that it will maximize these terms when the agent reports its true type.

But, agent i can also delay its reported arrival, which can affect the value of term (C) in addition to terms (A) + (B). This is a new manipulation, not possible to an agent in the offline VCG but possible here because an agent can change the set of agents other than itself with which its marginal negative effect is judged by delaying its arrival. However, BNIC is retained because in equilibrium *the expected decrease in (C) caused by delay c to agent i 's arrival is equal to the expected decrease in (B)*, and these two effects cancel out. Thus, an agent can report its true arrival time, and contingent on this we have established that an agent should report its true type.

To understand this, add term $\mathbb{E}_{\pi^*} \left\{ \sum_{t=a_i}^{a_i+c-1} R_{-i}(h^t, \pi^*(h^t)) \right\}$ to term (B) and subtract it again from term (C). The adjusted term (C') is now agent independent (by the independence property) and can be ignored for the purpose of establishing BNIC. Term (A) combined with adjusted term (B') is the expected value to all other agents forward from period a_i , plus the expected true value to agent i . Agent i 's best response is to report its true type (and immediately upon arrival) because the policy π^* is defined to maximize (A)+(B') when the other agents are truthful, i.e. in a Bayes-Nash equilibrium.

Remarks

The dynamic VCG mechanism is BNIC but not dominant strategy IC. This is a different solution concept than in an offline VCG mechanism. We only have BNIC because the correctness of the policy depends on the center having the correct model for the distribution on agent types. Without the correct model the policy is not optimal in expectation and an agent with beliefs different from that of the center should deviate to improve (its belief about) the expected utility it will receive. The center can only have correct beliefs in equilibrium.

In addition to expected-value maximizing, the dynamic VCG mechanism is ex post IR if the environment satisfies *agent-monotonicity*, which requires that introducing an agent to a state always has a positive expected effect on the total value of the system. This is generally true, except in domains

where there are effects such as congestion, for instance physical domains where the arrival of an additional robot may block the movements of other robots.

The dynamic VCG mechanism also satisfies *ex ante* WBB if the environment satisfies *no positive externalities*, which requires that the arrival of an agent does not have a positive expected effect on the total value of the *other* agents. This holds when agents are all consumers, but not when some agents contribute value to a team by their presence (e.g. sellers in a market, or robots bringing a new skill that enables a task that all robots care about to be completed more quickly).

Parkes et al. [PSY04] construct an ϵ -BNIC online mechanism by coupling the dynamic VCG mechanism with an approximate, sparse-sampling algorithm to compute the online decision in each period [KMN99]. The algorithm is a good fit with the requirements of the dynamic VCG mechanism because the payments just require an estimate of the system value with agent i and without agent i upon its arrival, and sparse-sampling can be used to get such an estimate. The approximate, dynamic VCG mechanism is ϵ -BNIC, in the sense that no agent can gain more than some amount $\epsilon > 0$ (that can be made arbitrarily small) by deviating from truthful reporting, as long as the other agents are truthful. This example illustrates making a tradeoff between an exact equilibrium solution concept and computational tractability.

The dynamic VCG mechanism has been further generalized by Bergemann and Välimäki [BV06], whose work along with that of Cavallo et al. [CPS06] applies to a model in which agent state changes across time. Because of this, incentive compatibility must now be assured in every period, so that an agent will continue to report its true state information. For an application, consider a multi-agent variation on the classical multi-armed bandits problem. Each agent owns an “arm” and receives a reward when its arm is activated, sampled from a stationary distribution. The reward signals are privately observed and allow an agent to update its model for the reward on its arm. In a setting with an infinite time horizon and discounting, one can use Gittins’ [GJ74] celebrated index policy to characterize an efficient online policy that makes the optimal tradeoff between exploitation and exploration. In the presence of self-interest, the generalized (dynamic) VCG mechanism provides incentives to support truthful reporting of reward signals by each agent, and thus implement the efficient learning policy.

1.7 Conclusions

Mechanism design provides a beautiful but theory for how to perform optimization in multi-agent systems with self-interest and distributed, privately known information. However, mechanism design must be extended in many dimensions to make it widely applicable to distributed intelligent systems. This is natural because mechanisms are being realized, and in realizing them new considerations— e.g. those of computational and communication efficiency —come to light. Progress on this agenda, in the emerging subfield of computational mechanism design, requires a strong background in the techniques of computer science, microeconomics, and operations research.

In these notes we have considered three main topics: *algorithmic* mechanism design and its focus on the computational complexity of centralized problem solving in MD; indirect mechanisms and the problem of efficient preference elicitation in CMD; *distributed* implementation, distributed AMD, and the problems entailed in getting agents involved in computing the outcome of mechanisms; and *online* MD in which mechanism design is applied to dynamic environments.

To date, CMD has relied heavily on dominant-strategy implementation, and to a lesser extent ex post Nash implementation. Yet, it seems likely that an important future direction is to develop new solution concepts. There is mounting evidence of the difficulties in implementing desirable outcomes with these strong solution concepts. Moreover, agents (both computational and human) are intrinsically bounded in their capacity for deliberation and modeling. One challenge, going forward is to find analytically tractable models of bounded-rationality; we need models that support the *design* of new mechanisms.⁴⁵

Much current work in CMD focuses on worst-case approximation results and polynomial-time algorithms. This attention to the worst-case is somewhat at odds with the traditions in artificial intelligence and operations research of heuristic, anytime algorithms such as tree search for solving optimization problems. Algorithms such as these are designed to work well on typical problem instances and fail gracefully. We should expect future work to find methods to integrate the search and sample-based methods of AI and OR into the methods of CMD, retaining appropriate incentive properties [PS04a, PD07, e.g.].

Research in CMD will also need to acknowledge, and then systematically attack, the obvious limitations in the equilibrium concepts and models

⁴⁵Some work has started in this direction [LS01, LS04a, LN05, HB04, HB06, HB07, BW05, Par04]. We also note that economics is increasingly welcoming of *behavioral* models of agent behavior [LG04, Mul02, Rab98, Wil04].

adopted in current work. Most glaring of these is the almost exclusive consideration of unilateral deviation. With limited exception [GH05, e.g.], no attention has been given to the possibilities of collusive behavior or other coordinated manipulations. Another example of a largely unmodeled manipulation is that of *false-name* bidding [Yok03, YSM04]. Other weaknesses of current models include the private values assumption [IP06, e.g.] and the quasilinear utility assumption [BCI⁺05, e.g.]. Finally, there are many computational domains in which it is impossible to support payments, but there has been comparatively less focus on implementation in settings with ordinal preferences [CSL07].

1.8 Acknowledgments

This research is supported in part by National Science Foundation grants IIS-0238147, IIS-0534620 and an Alfred P. Sloan Foundation award. Many of the ideas have come out of inspiring conversations with collaborators, my wonderful group of students, and colleagues. Thanks to the IMS of the National University of Singapore, and especially Yeneng Sun, for supporting a visit in June 2005 and facilitating the tutorial upon which this chapter is structured. Peter Hammond offered helpful comments on an earlier draft.

Bibliography

- [AAM03] Baruch Awerbuch, Yossi Azar, and Adam Meyerson. Reducing truth-telling online mechanisms to online optimization. In *Proc. ACM Symposium on Theory of Computing (STOC'03)*, pages 503–510, 2003.
- [Abr06] Zoe Abrams. Revenue maximization when bidders have budgets. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.
- [AM02] Lawrence M Ausubel and Paul Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1:1–42, 2002.
- [Ang87] Dana Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1987.
- [Arr79] Kenneth J Arrow. The property rights doctrine and demand revelation under incomplete information. In M Boskin, editor, *Economics and Human Welfare*. Academic Press, New York, 1979.
- [AT01] A Archer and E Tardos. Truthful mechanisms for one-parameter agents. In *Proc. 42nd IEEE Symp. on Foundations of Computer Science*, 2001.
- [ATY00] Arne Andersson, Mattias Tenhunen, and Fredrik Ygge. Integer programming for auctions with bids for combinations. In *Proc. 4th International Conference on Multi-Agent Systems (ICMAS-00)* [icm00], pages 39–46.
- [BCI⁺05] C Borgs, J Chayes, N Immorlica, M Mahdian, and A Saberi. Multi-unit auctions with budget-constrained bidders. In *Proc. 6th ACM Conf. on Electronic Commerce*, pages 44–51, 2005.
- [BCL⁺06] S Bikhchandani, S Chatterji, R Lavi, A Mu'alem, N Nisan, and A Sen. Weak monotonicity characterizes incentive deterministic dominant strategy implementation. *Econometrica*, 74:1109–1132, 2006.
- [BGN03] Yair Bartal, Rica Gonen, and Noam Nisan. Incentive compatible multi unit combinatorial auctions. In *TARK '03: Proceedings of the 9th conference on Theoretical aspects of rationality and knowledge*, pages 72–87. ACM Press, 2003.
- [BGT03] Curt Bererton, Geoff Gordon, and Sebastian Thrun. Auction mechanism design for multi-robot coordination. In *Proc. 17th Annual Conf. on Neural Information Processing Systems (NIPS'03)*, 2003.
- [BHM77] S Bradley, A Hax, and T Magnanti. *Applied Mathematical Programming*. Addison-Wesley, 1977.
- [BJSZ04] Avrim Blum, Jeffrey Jackson, Tuomas Sandholm, and Martin Zinkevich. Preference elicitation and query learning. *Journal of Machine Learning Research (JMLR)*, 5:649–667, 2004.
- [BM07] Dirk Bergemann and Stephen Morris. Ex post implementation. *Games and Economic Behavior*, 2007. To appear.
- [BN05] Liad Blumrosen and Noam Nisan. On the computational power of iterative auctions. In *Proc. 6th ACM Conf. on Electronic Commerce*, pages 29–43, 2005.
- [BO02] Sushil Bikhchandani and Joseph M Ostroy. The package assignment model. *Journal of Economic Theory*, 107(2):377–406, 2002.
- [BP05] Jonathan Bredin and David C. Parkes. Models for truthful online double auctions. In *Proc. 21st Conference on Uncertainty in Artificial Intelligence (UAI'2005)*, pages 50–59, 2005.
- [BV03] Sandeep Baliga and Rakesh Vohra. Market research and market design. *BE Journal of Theoretical Economics*, 3, 2003.
- [BV06] Dirk Bergemann and Juuso Välimäki. Efficient dynamic auctions. Technical Report Cowles Foundation Discussion Paper No. 1584, Yale University, 2006.
- [BW05] Moshe Babaioff and William E Walsh. Incentive-compatible, budget-

- balanced, yet highly efficient auctions for supply chain formation. *Decision Support Systems*, 39:123–149, 2005.
- [Cav06] Ruggiero Cavallo. Optimal decision-making with minimal waste: Strategyproof redistribution of VCG payments. In *Proc. of the 5th Int. Joint Conf. on Autonomous Agents and Multi Agent Systems (AAMAS'06)*, 2006.
- [CBA⁺05] Brent N. Chun, Philip Buonadonna, Alvin AuYoung, Chaki Ng, David C. Parkes, Jeffrey Shneidman, Alex C. Snoeren, and Amin Vahdat. Mirage: A microeconomic resource allocation system for sensornet testbeds. In *Proceedings of 2nd IEEE Workshop on Embedded Networked Sensors (EmNetsII)*, 2005.
- [CHK07] Shuchi Chawla, Jason Hartline, and Robert Kleinberg. Algorithmic pricing via virtual valuations. In *Proc. ACM Conference on Electronic Commerce*, 2007.
- [CJ07] Olivier Compte and Philippe Jehiel. Auctions and information acquisition: Sealed-bid or Dynamic Formats? *Rand Journal of Economics*, 2007.
- [Cla71] E H Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.
- [CP05] Florin Constantin and David C. Parkes. Preference-based characterizations of truthfulness and the limited expressiveness of order-based domains. In *Proc. Workshop on Preference Handling*, Edinburgh, Scotland, 2005.
- [CP06] Estelle Cantillon and Martin Pesendorfer. Auctioning bus Routes: The London experience. In Cramton et al. [CSS06], chapter 22.
- [CPS06] Ruggiero Cavallo, David C. Parkes, and Satinder Singh. Optimal coordinated learning among self-interested agents in the multi-armed bandit problem. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI'2006)*, Cambridge, MA, 2006.
- [CS02a] V Conitzer and T Sandholm. Complexity of mechanism design. In *Proc. 18th Conf. on Uncertainty in Artificial Intelligence (UAI'02)*, pages 103–110, 2002.
- [CS02b] V Conitzer and T Sandholm. Vote Elicitation: Complexity and Strategy-Proofness. In *Proc. 18th National Conference on Artificial Intelligence (AAAI-02)*, pages 392–397, 2002.
- [CS03] Vincent Conitzer and Tuomas Sandholm. Applications of automated mechanism design. In *Proc. UAI Bayesian Modeling Applications Workshop*, Aca-pulco, Mexico, 2003.
- [CS04] Vincent Conitzer and Tuomas Sandholm. Computational criticisms of the revelation principle. In *Proc. 5th ACM Conference on Electronic Commerce (EC'04)*, pages 262–263, 2004. Short paper.
- [CSL07] Vincent Conitzer, Tuomas Sandholm, and Jerome Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 2007.
- [CSS06] Peter Cramton, Yoav Shoham, and Richard Steinberg, editors. *Combinatorial Auctions*. MIT Press, January 2006.
- [dG79] C d'Aspremont and L Gerard-Varet. Incentives and incomplete information. *J. of Public Economics*, 11:25–45, 1979.
- [DHM79] Partha S Dasgupta, Peter Hammond, and Eric S. Maskin. The Implementation of Social Choice Rules: Some General Results on Incentive Compatibility. *Review of Economic Studies*, 46:185–216, 1979.
- [DNS05] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation algorithms for combinatorial auctions with complement free bidders. In *Proc. 37th ACM Symposium on Theory of Computing (STOC)*, 2005.
- [DNS06] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Truthful random-

- ized mechanisms for combinatorial auctions. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 644–652. ACM Press, 2006.
- [DS06] Shahar Dobzinski and Michael Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1064–1073, 2006.
- [dVSV07] Sven de Vries, James Schummer, and Rakesh V Vohra. On ascending Vickrey auctions for heterogeneous objects. *Journal of Economic Theory*, 132:95–118, 2007.
- [dVV03] Sven de Vries and Rakesh V Vohra. Combinatorial auctions: A survey. *Inform Journal on Computing*, 15(3):284–309, 2003.
- [ec000] *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, 2000.
- [ER91] Eithan Ephrati and Jeffrey S Rosenschein. The Clarke tax as a consensus mechanism among automated agents. In *Proc. 9th National Conference on Artificial Intelligence (AAAI-91)*, pages 173–178, July 1991.
- [Fal04] Boi Faltings. A budget-balanced, incentive-compatible scheme for social choice. In *Workshop on Agent-mediated E-commerce (AMEC)*, 2004.
- [FGHK02] Amos Fiat, Andrew Goldberg, Jason Hartline, and Anna Karlin. Competitive generalized auctions. In *Proc. 34th ACM Symposium on Theory of Computing (STOC'02)*, 2002.
- [FK00] I T Foster and C Kesselman. Computational grids. In *VECPAR*, pages 3–37, 2000.
- [FKSS01] Joan Feigenbaum, Arvind Krishnamurthy, Rahul Sami, and Scott Shenker. Approximation and collusion in multicast cost sharing. In *Proc. of the 3rd Conference on Electronic Commerce*, pages 253–255, 2001.
- [FP03] Eric Friedman and David C. Parkes. Pricing WiFi at Starbucks— Issues in online mechanism design. In *Fourth ACM Conf. on Electronic Commerce (EC'03)*, pages 240–241, 2003. Short Paper.
- [FPSS02] Joan Feigenbaum, Christos Papadimitriou, Rahul Sami, and Scott Shenker. A BGP-based mechanism for lowest-cost routing. In *Proceedings of the 2002 ACM Symposium on Principles of Distributed Computing*, pages 173–182, 2002.
- [FS] Joan Feigenbaum and Scott Shenker. Distributed algorithmic mechanism design: Recent results and future directions. Presentation at: <http://cs-www.cs.yale.edu/homes/jf/DIALM02.pdf>.
- [FS02] 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.
- [FSS07] Joan Feigenbaum, Michael Schapira, and Scott Shenker. Distributed algorithmic mechanism design. In Nisan et al. [NRTV07], chapter 14.
- [FT91] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991.
- [GC07] M Guo and V Conitzer. Worst-case optimal redistribution of VCG payments. In *Proc. 8th ACM Conference on Electronic Commerce*, 2007.
- [Geo70] Arthur M Geoffrion. Elements of large-scale mathematical programming. *Management Science*, 16(11):652–691, 1970.
- [GH03] Andrew Goldberg and Jason Hartline. Envy-free auctions for digital goods. In *Proc. ACM E'Commerce*, pages 29–35, 2003.
- [GH05] Andrew Goldberg and Jason Hartline. Collusion-resistant mechanisms for single parameter agents. In *Proc. ACM-SIAM Symposium on Discrete Algo-*

- rithms (SODA)*, pages 620–629, 2005.
- [GHK⁺06] Andrew Goldberg, Jason Hartline, Anna Karlin, Mike Saks, and Andrew Wright. Competitive auctions. *Games and Economic Behavior*, 55:242–269, 2006.
- [Gib73] Alan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41:587–602, 1973.
- [GJ74] J. C. Gittins and D. M. Jones. A dynamic allocation index for the sequential design of experiments. In J. Gani, K. Sakadi, and I. Vinczo, editors, *Progress in Statistics*, pages 241–266. North-Holland, 1974.
- [GL77a] Jerry Green and Jean-Jacques Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45:427–438, 1977.
- [GL77b] Theodore Groves and John O Ledyard. Optimal allocation of public goods: A solution to the “free rider” problem. *Econometrica*, 45:783–809, 1977.
- [GMV04] Hongwei Gui, Rudolf Muller, and Rakesh Vohra. Characterizing dominant strategy mechanisms with multi-dimensional types. Technical report, Northwestern University, 2004.
- [Gro73] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.
- [GS00] Faruk Gul and Ennio Stacchetti. The English auction with differentiated commodities. *Journal of Economic Theory*, pages 66–95, 2000.
- [H99] J Hästad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.
- [HB00] Holger H Hoos and Craig Boutilier. Solving combinatorial auctions with stochastic local search. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-00)*, pages 22–29, 2000.
- [HB04] Nathanael Hyafil and Craig Boutilier. Regret minimizing equilibria and mechanisms for games with strict type uncertainty. In *Proceedings of the Twentieth Annual Conference on Uncertainty in Artificial Intelligence (UAI-04)*, pages 268–277, 2004.
- [HB06] Nathanael Hyafil and Craig Boutilier. Regret-based incremental partial revelation mechanisms. In *Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI-06)*, pages 672–678, 2006.
- [HB07] Nathanael Hyafil and Craig Boutilier. Mechanism design with partial revelation. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, 2007.
- [HG00] Luke Hunsberger and Barbara J Grosz. A combinatorial auction for collaborative planning. In *Proc. 4th International Conference on Multi-Agent Systems (ICMAS-00)* [icm00], pages 151–158.
- [HKMP05] Mohammad T. Hajiaghayi, Robert Kleinberg, Mohammad Mahdian, and David C. Parkes. Online auctions with re-usable goods. In *Proc. ACM Conf. on Electronic Commerce*, pages 165–174, 2005.
- [HKP04] Mohammad T. Hajiaghayi, Robert Kleinberg, and David C. Parkes. Adaptive limited-supply online auctions. In *Proc. ACM Conf. on Electronic Commerce*, pages 71–80, 2004.
- [HM83] Bengt Holmstrom and Roger B. Myerson. Efficient and durable decision rules with incomplete information. *Econometrica*, 51:1799–1820, 1983.
- [Hol79] Bengt Holmström. Groves’ scheme on restricted domains. *Econometrica*, 47(5):1137–1144, 1979.
- [Hur75] Leonid Hurwicz. On the existence of allocation systems whose manipulative Nash equilibria are Pareto optimal. Unpublished paper presented at the 3rd

- World Congress of the Economic Society, Toronto, 1975.
- [icm00] *Proc. 4th International Conference on Multi-Agent Systems (ICMAS-00)*, 2000.
- [IK06] Garud Iyengar and Anuj Kumar. Characterizing optimal adword auctions. In *Proc. Second Workshop on Sponsored Search*, 2006.
- [IP06] Takayuki Ito and David C. Parkes. Instantiating the contingent bids model of truthful interdependent value auctions. In *Proc. 6th Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, 2006.
- [Jac01] Matthew O. Jackson. A crash course in Implementation theory. *Social Choice and Welfare*, 18(4):655–708, 2001.
- [Jac03] Matthew O. Jackson. Mechanism theory. In Ulrich Derigs, editor, *The Encyclopedia of Life Support Systems*. EOLSS Publishers, 2003.
- [JM01] P. Jehiel and B. Moldovanu. Efficient design with interdependent valuations. *Econometrica*, 69:1237–1259, 2001.
- [JP06] Adam Juda and David Parkes. The sequential auction problem on eBay: An empirical analysis and a solution. In *Proc. 7th ACM Conf. on Electronic Commerce (EC’06)*, pages 180–189, 2006.
- [JtVM07] Philippe Jehiel, Moritz Meyer ter Vehn, and Benny Moldovanu. Mixed bundling auctions. *Journal of Economic Theory*, 134:494–512, 2007.
- [Kle00] Paul Klemperer. *The Economic Theory of Auctions*, chapter 1. Edward Elgar, 2000. Auction Theory: A Guide to the Literature.
- [KMN99] Michael Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large Markov Decision Processes. In *Proc. 16th Int. Joint Conf. on Artificial Intelligence*, pages 1324–1331, 1999.
- [KP00] Vijay Krishna and Motty Perry. Efficient mechanism design. Technical report, Pennsylvania State University, 2000. Available at: <http://econ.la.psu.edu/~vkrishna/vcg18.ps>.
- [Kri02] Vijay Krishna. *Auction Theory*. Academic Press, 2002.
- [LCP05] Sébastien M Lahaie, Florin Constantin, and David C Parkes. More on the power of demand queries in combinatorial auctions: Learning atomic languages and handling incentives. In *Proc. 19th Int. Joint Conf. on Artificial Intell. (IJCAI’05)*, 2005.
- [Led07] John Ledyard. Optimal combinatoric auctions for single-minded bidders. In *Proc. ACM Conference on Electronic Commerce*, 2007.
- [LG04] David Laibson and Xavier Gabaix. Bounded rationality and directed cognition. Technical report, Harvard University, 2004.
- [LMN03] R Lavi, A Mu’alem, and N Nisan. Towards a characterization of truthful combinatorial auctions. In *Proc. 44th Annual Symposium on Foundations of Computer Science*, page 574, 2003.
- [LMS06] Daniel Lehmann, Rudolf Muller, and Tuomas Sandholm. The winner determination problem. In Cramton et al. [CSS06], chapter 12.
- [LN00] Ron Lavi and Noam Nisan. Competitive analysis of incentive compatible on-line auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00) [ec000]*, pages 233–241.
- [LN05] Ron Lavi and Noam Nisan. Online ascending auctions for gradually expiring goods. In *Proc. of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1146–1155, 2005.
- [LOS02] Daniel Lehmann, Liadan Ita O’Callaghan, and Yoav Shoham. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, September 2002.

- [LP04] Sébastien M Lahaie and David C Parkes. Applying learning algorithms to preference elicitation. In *Proc. ACM Conf. on Electronic Commerce*, pages 180–188, 2004.
- [LS01] Kate Larson and Tuomas Sandholm. Bargaining with limited computation: Deliberation equilibrium. *Artificial Intelligence*, 132(2):183–217, 2001.
- [LS04a] Kate Larson and Tuomas Sandholm. Experiments on deliberation equilibria in auctions. In *Proc. 3rd Int. Joint. Conf. on Autonomous Agents and Multi Agent Systems*, pages 394–401, 2004.
- [LS04b] A Likhodedov and T Sandholm. Methods for boosting revenue in combinatorial auctions. In *Proc. 19th National Conference on Artificial Intelligence (AAAI-04)*, pages 232–237, 2004.
- [LS05] A Likhodedov and T Sandholm. Approximating revenue-maximizing combinatorial auctions. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2005.
- [LS07] Ron Lavi and Chaitanya Swamy. Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity. In *Proc. ACM Conference on Electronic Commerce*, 2007.
- [Mö06] Rudolf Müller. Tractable cases of the winner determination problem. In Cramton et al. [CSS06], chapter 13.
- [MCWG95] Andreu Mas-Colell, Michael D Whinston, and Jerry R Green. *Microeconomic Theory*. Oxford University Press, 1995.
- [Mil04] Paul Milgrom. *Putting Auction Theory to Work*. Cambridge University Press, 2004.
- [ML04] Roger Mailler and Victor Lesser. Solving distributed constraint optimization problems using cooperative mediation. In *Proceedings of Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, pages 438–445, 2004.
- [MM96] R Preston McAfee and John McMillan. Analyzing the airwaves auction. *J Econ Perspect*, 10:159–175, 1996.
- [MN02] A. Mu’alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. In *Proc. 18th National Conference on Artificial Intelligence (AAAI-02)*, 2002. .
- [MP07] Debasis Mishra and David C. Parkes. Ascending price Vickrey auctions for general valuations. *Journal of Economic Theory*, 132:335–366, 2007.
- [MS83] Robert B Myerson and Mark A Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 28:265–281, 1983.
- [MSTY05] P. J. Modi, W. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence Journal*, 161:149–180, 2005.
- [MT99] Dov Monderer and Moshe Tennenholtz. Distributed games. *Games and Economic Behavior*, 28(1):55–72, 1999.
- [Mul02] Sendhil Mullainathan. A memory-based model of bounded rationality. *Quarterly Journal of Economics*, 117:735–774, 2002.
- [MV04] A Malakhov and R V Vohra. Single and multi-dimensional optimal auctions – A network approach. Technical report, Kellogg School of Management, Northwestern University, 2004.
- [Mye81] Robert B Myerson. Optimal auction design. *Mathematics of Operation Research*, 6:58–73, 1981.
- [Nis00] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)* [ec000], pages 1–12.

- [Nis06] Noam Nisan. Bidding languages for combinatorial auctions. In Cramton et al. [CSS06], chapter 9.
- [NR00] Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)* [ec000], pages 242–252.
- [NR01] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.
- [NRTV07] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [NS05] Noam Nisan and Ilya Segal. Exponential communication inefficiency of demand queries. In *Tenth Conf. on the Theoretical Aspects of Rationality and Knowledge*, 2005.
- [NS06] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129:192–224, 2006.
- [NW99] George Nemhauser and Laurence Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, 1999.
- [OR94] Martin J Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
- [Par01] David C Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, May 2001.
- [Par02] 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*, pages 103–122. Springer Verlag, 2002.
- [Par04] David C. Parkes. On learnable mechanism design. In Kagan Tumer and David Wolpert, editors, *Collectives and the Design of Complex Systems*, pages 107–131. Springer-Verlag, 2004.
- [Par05] David C. Parkes. Auction design with costly preference elicitation. *Annals of Mathematics and AI*, 44:269–302, 2005. Special Issue on the Foundations of Electronic Commerce.
- [Par07] David C Parkes. On-line mechanisms. In Nisan et al. [NRTV07], chapter 16.
- [PD07] David C Parkes and Quang Duong. An ironing-based approach to adaptive online mechanism design in single-valued domains. In *Proc. 22nd National Conference on Artificial Intelligence (AAAI'07)*, 2007.
- [PF05] Adrian Petcu and Boi Faltings. A scalable method for multiagent constraint optimization. In *Proc. 19th Int. Joint Conf. on Art. Intelligence (IJCAI'05)*, pages 266–271, 2005.
- [PFP06] Adrian Petcu, Boi Faltings, and David C. Parkes. MDPOP: Faithful distributed implementation of efficient social choice problems. In *Proc. 5th Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, 2006.
- [Por04] Ryan Porter. Mechanism design for online real-time scheduling. In *Proc. ACM Conf. on Electronic Commerce (EC'04)*, pages 61–70, 2004.
- [PRST02] Ryan Porter, Amir Ronen, Yoav Shoham, and Moshe Tennenholtz. Mechanism design with execution uncertainty. In *Proc. 18th Conf. on Uncertainty in Artificial Intelligence (UAI'02)*, 2002.
- [PS03] David C. Parkes and Satinder Singh. An MDP-based approach to Online Mechanism Design. In *Proc. 17th Annual Conf. on Neural Information Processing Systems (NIPS'03)*, 2003.

- [PS04a] David C. Parkes and Grant Schoenebeck. Growrange: Anytime VCG-Based Mechanisms. In *Proc. 19th National Conference on Artificial Intelligence (AAAI-04)*, pages 34–41, 2004.
- [PS04b] David C. Parkes and Jeffrey Shneidman. Distributed implementations of Vickrey-Clarke-Groves mechanisms. In *Proc. 3rd Int. Joint Conf. on Autonomous Agents and Multi Agent Systems*, pages 261–268, 2004.
- [PSY04] David C. Parkes, Satinder Singh, and Dimah Yanovsky. Approximately efficient online mechanism design. In *Proc. 18th Annual Conf. on Neural Information Processing Systems (NIPS'04)*, 2004.
- [PU00] 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, 2000.
- [Put94] M. L. Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, New York, 1994.
- [Rab98] Matthew Rabin. Psychology and economics. *Journal of Economic Literature*, 36:11–46, 1998.
- [Rob79] Kevin Roberts. The characterization of implementable rules. In Jean-Jacques Laffont, editor, *Aggregation and Revelation of Preferences*, pages 321–348. North-Holland, Amsterdam, 1979.
- [Roc87] J.-C. Rochet. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics*, 16:191–200, 1987.
- [RPH98] Michael H Rothkopf, Aleksandar Pekewc, and Ronald M Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
- [RZ94] Jeffrey S Rosenschein and Gilad Zlotkin. *Rules of Encounter*. MIT Press, 1994.
- [San93] Tuomas W Sandholm. An implementation of the Contract Net Protocol based on marginal-cost calculations. In *Proc. 11th National Conference on Artificial Intelligence (AAAI-93)*, pages 256–262, July 1993.
- [San96] Tuomas W Sandholm. Limitations of the Vickrey auction in computational multiagent systems. In *Proc. Second International Conference on Multiagent Systems (ICMAS-96)*, pages 299–306, 1996.
- [San02] Tuomas W Sandholm. eMediator: A next generation electronic commerce server. *Computational Intelligence*, 18:656–676, 2002.
- [San06] Tuomas Sandholm. Optimal winner determination algorithms. In Cramton et al. [CSS06], chapter 14.
- [Sat75] Mark A Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.
- [Seg03] Ilya Segal. Optimal pricing mechanisms with unknown demand. *American Economic Review*, 93:509–529, 2003.
- [SP04a] Saurabh Sanghvi and David C. Parkes. Hard-to-manipulate combinatorial auctions. Technical report, Harvard University, 2004.
- [SP04b] Jeffrey Shneidman and David C. Parkes. Specification faithfulness in networks with rational nodes. In *Proc. 23rd ACM Symp. on Principles of Distributed Computing (PODC'04)*, pages 88–97, St. John's, Canada, 2004.
- [SS93] R Schapire and L Sellie. Learning sparse multivariate polynomials over a field with queries and counterexamples. In *Proc. 6th Annual ACM Workshop on Computational Learning Theory*, pages 17–26, 1993.
- [SSGL05] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. CABOB: A fast optimal

- algorithm for winner determination in combinatorial auctions. *Management Science*, 51(3):374–390, 2005.
- [SY05] Michael Saks and Lan Yu. Weak monotonicity suffices for truthfulness on convex domains. In *Proc. 6th ACM Conf. on Electronic Commerce*, pages 286–293, 2005.
- [Ü06] Levent Ülkü. Optimal combinatorial mechanism design. Technical report, Rutgers University, 2006.
- [Vic61] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.
- [Voh07] Rakesh V. Vohra. Paths, cycles and mechanism design. Technical report, Kellogg School of Management, Northwestern University, 2007.
- [Wil04] Andrea Wilson. Bounded memory and biases in information processing. Technical report, University of Chicago, 2004.
- [Yok03] Makoto Yokoo. The characterization of strategy/false-name proof combinatorial auction protocols: Price-oriented, rationing-free protocol. In *Proc. 18th Int. Joint Conf. on Art. Intell.*, pages 733–742, 2003.
- [YSM04] Makoto Yokoo, Y Sakurai, and S Matsubara. The effect of false-name bids in combinatorial auctions: New Fraud in Internet Auctions. *Games and Economic Behavior*, 46(1):174–188, 2004.
- [ZBS03] Martin Zinkevich, Avrim Blum, and Tuomas Sandholm. On polynomial-time preference elicitation with value queries. In *Proc. 4th ACM Conf. on Electronic Commerce*, pages 176–185, 2003.
- [ZN01] Edo Zurel and Noam Nisan. An efficient approximate allocation algorithm for combinatorial auctions. In *Proc. 3rd ACM Conference on Electronic Commerce*, pages 125–136. ACM Press, 2001.