

# An Optimization-Based Framework for Automated Market-Making

Jacob Abernethy  
EECS Department  
UC Berkeley  
jake@cs.berkeley.edu

Yiling Chen  
School of Engineering and  
Applied Sciences  
Harvard University  
yiling@eecs.harvard.edu

Jennifer Wortman Vaughan  
Computer Science Department  
UCLA  
jenn@cs.ucla.edu

## ABSTRACT

We propose a general framework for the design of securities markets over combinatorial or infinite state or outcome spaces. The framework enables the design of computationally efficient markets tailored to an arbitrary, yet relatively small, space of securities with bounded payoff. We prove that any market satisfying a set of intuitive conditions must price securities via a convex cost function, which is constructed via conjugate duality. Rather than deal with an exponentially large or infinite outcome space directly, our framework only requires optimization over a convex hull. By reducing the problem of automated market making to convex optimization, where many efficient algorithms exist, we arrive at a range of new polynomial-time pricing mechanisms for various problems. We demonstrate the advantages of this framework with the design of some particular markets. We also show that by relaxing the convex hull we can gain computational tractability without compromising the market institution's bounded budget.

## Categories and Subject Descriptors

J.4 [Social and Behavioral Sciences]: Economics; G.1.6 [Numerical Analysis]: Optimization

## General Terms

Algorithms, Economics, Theory

## 1. INTRODUCTION

*Securities markets* play a fundamental role in economics and finance. A securities market offers a set of *contingent securities* whose payoffs depend on the future state of the world. For example, an Arrow-Debreu security pays \$1 if a particular state of the world is reached and \$0 otherwise [3, 4]. Consider an Arrow-Debreu security that will pay off in the event that a category 4 or higher hurricane passes through Florida in 2011. A Florida resident who worries about his home being damaged might buy this security as

a form of insurance to hedge his risk; if there is a hurricane powerful enough to damage his home, he will be compensated. Additionally, a risk-neutral trader who has reason to believe that the probability of a category 4 or higher hurricane landing in Florida in 2011 is  $p$  should be willing to buy this security at any price below  $p$  or (short) sell it at any price above  $p$  to capitalize his information. For this reason, the market price of the security can be viewed as the traders' collective estimate of how likely it is that a powerful hurricane will occur. Securities markets thus have dual functions: *risk allocation* and *information aggregation*.

Insurance contracts, options, futures, and many other financial derivatives are examples of contingent securities. A securities market primarily focused on information aggregation is often referred to as a *prediction market*. The forecasts of prediction markets have proved to be accurate in a variety of domains [5, 24, 32]. While our work builds on ideas from prediction market design [2, 10, 28], our framework can be applied to any contingent securities.

A securities market is said to be *complete* if it offers  $|\mathcal{O}| - 1$  independent securities over a set  $\mathcal{O}$  of mutually exclusive and exhaustive states of the world, which we refer to as *outcomes* [3, 4, 25]. For example, a prediction market with  $n$  Arrow-Debreu securities for  $n$  outcomes is complete. In a complete securities market without transaction fees, a trader may bet on any combination of the securities, allowing him to hedge any possible risk he may have. It is generally assumed that the trader may *short sell* a contract, betting against the given outcome; in a market with short selling, the  $n$ th security is not strictly necessary, as a trader can substitute the purchase of this contract by short selling all others. Furthermore, traders can change the market prices to reflect any valid probability distribution over the outcome space, allowing them to reveal any belief. Completeness therefore provides expressiveness for both risk allocation and information aggregation.

Unfortunately, completeness is not always achievable. In many real-world settings, the outcome space is exponentially large or even infinite. For instance, a competition among  $n$  candidates results in an outcome space of  $n!$  rank orders, while the future price of a stock has an infinite outcome space, namely  $\mathbb{R}_+$ . In such situations operating a complete securities market is not practical for two reasons: (a) humans are notoriously bad at estimating small probabilities and (b) it is computationally intractable to manage such a large set of contracts. Instead, it is natural to offer a smaller set of structured securities. For example, rather than offer a security corresponding to each rank ordering,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EC'11, June 5–9, 2011, San Jose, California, USA.

Copyright 2011 ACM 978-1-4503-0261-6/11/06 ...\$10.00.

in *pair betting* a market maker offers securities of the form “\$1 if candidate A beats candidate B” [11]. There has been a surge of recent research examining the tractability of running standard prediction market mechanisms (such as the popular Logarithmic Market Scoring Rule (LMSR) market maker [20]) over combinatorial outcome spaces by limiting the space of available securities [29]. While this line of research has led to a few positive results [1, 13, 14, 19], it has led more often to hardness results [12, 14] or to markets with undesirable properties such as unbounded loss of the market institution [16].

In this paper, we propose a general framework to design *automated market makers* for securities markets. An automated market maker is a market institution that adaptively sets prices for each security and is always willing to accept trades at these prices. Unlike previous research aimed at finding a space of securities that can be efficiently priced using an existing market maker like LMSR, we start with an arbitrary space of securities and design a *new* market maker tailored to this space. Our framework is therefore extremely general and includes both the LMSR and Quad-SCPM [2] market makers.

We take an axiomatic approach. Given a relatively small space of securities with bounded payoff, we define a set of intuitive conditions that a reasonable market maker should satisfy. We prove that a market maker satisfying these conditions must price securities via a convex potential function (the *cost function*), and that the space of reachable security prices must be precisely the convex hull of the payoff vectors for each outcome (that is, the set of vectors, one per outcome, denoting the payoff for each security if that outcome occurs). We then incorporate ideas from online convex optimization [22, 31] to define a convex cost function in terms of an optimization over this convex hull; the vector of prices is chosen as the optimizer of this convex objective. With this framework, instead of dealing with the exponentially large or infinite outcome space, we only need to deal with the lower-dimensional convex hull. The problem of automated market making is reduced to the problem of convex optimization, for which we have many efficient techniques to leverage.

To demonstrate the advantages of our framework, we provide two new computationally efficient markets. The first market can efficiently price *subset bets* on permutations, which are known to be #P-hard to price using LMSR [12]. The second market can be used to price bets on the landing location of an object on a sphere.

Finally, for situations where the convex hull cannot be efficiently represented, we show that we can relax the convex hull to gain computational tractability without compromising the market maker’s bounded budget. This allows us to provide a computationally efficient market maker for the aforementioned pair betting, which is also known to be #P-hard to price using LMSR [12]. In addition, this relaxation could potentially allow the market maker to charge transaction fees so that the depth of the market can be dynamically increased as the number of trades increases. This desirable property is discussed in Othman et al. [28] who proposed a corresponding market that extends the LMSR; utilizing our techniques for this purpose is left for future work.

This paper builds upon intuitions from earlier work [10] exploring the striking mathematical connections between complete, cost function based prediction markets and no-regret learning and online convex optimization. In this pa-

per, we pursue a general convex optimization framework for pricing an arbitrary set of securities over a potentially very large outcome space. Agrawal et al. [2] and Peters et al. [30] also use convex optimization for automated market making. In addition to only considering complete markets, they formulate the convex optimization quite differently – their formulation is for limit orders and does not explicitly make the connection with conjugate duality as our framework does. To the best of our knowledge, this paper is the first work on designing markets that are tailored to the *incomplete* security spaces without explicitly reasoning over a large outcome space.

## 2. PRELIMINARIES

A simple cost function based market maker [9, 10, 20, 21] offers  $|\mathcal{O}|$  Arrow-Debreu securities, each corresponding to a potential outcome. The market maker determines how much each security should cost using a differentiable *cost function*,  $C : \mathbb{R}^{|\mathcal{O}|} \rightarrow \mathbb{R}$ , which is simply a potential function specifying the amount of money currently wagered in the market as a function of the number of shares of each security that have been purchased. If  $q_{\sigma}$  is the number of shares of security  $\sigma$  currently held by traders, and a trader would like to purchase a bundle of  $r_{\sigma}$  shares for each security  $\sigma \in \mathcal{O}$  (where some  $r_{\sigma}$  could be zero or even negative, representing a sale), the trader must pay  $C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q})$  to the market maker. The instantaneous price of security  $\sigma$  (that is, the price per share of an infinitely small portion of a security) is then  $\partial C(\mathbf{q})/\partial q_{\sigma}$ , and is denoted  $p_{\sigma}(\mathbf{q})$ .

The market designer is free to choose any differentiable cost function  $C$  that satisfies a few basic properties. First, it must be the case that for every  $\sigma \in \mathcal{O}$  and every  $\mathbf{q} \in \mathbb{R}^{|\mathcal{O}|}$ ,  $p_{\sigma}(\mathbf{q}) \geq 0$ . This ensures that the price of a security is never negative. Second, assuming short sells are allowed, if the market designer wishes to prevent arbitrage, it must be the case that for every  $\mathbf{q} \in \mathbb{R}^{|\mathcal{O}|}$ ,  $\sum_{\sigma \in \mathcal{O}} p_{\sigma}(\mathbf{q}) = 1$ . That is, the sum of the instantaneous prices of the securities must always be 1. If the prices summed to something less than (respectively, greater than) 1, then a trader could purchase (respectively, sell) small equal quantities of each security for a guaranteed profit.<sup>1</sup> These conditions ensure that the current prices can always be viewed as a probability distribution over the outcome space. One example of a cost function based market that has received considerable attention is Hanson’s Logarithmic Market Scoring Rule (LMSR) [9, 20, 21]. The cost function of the LMSR is  $C(\mathbf{q}) = b \log \sum_{\sigma \in \mathcal{O}} e^{q_{\sigma}/b}$ , where  $b > 0$  is a parameter of the market controlling the rate at which prices change. The corresponding price function for each security  $\sigma$  is  $p_{\sigma}(\mathbf{q}) = \partial C(\mathbf{q})/\partial q_{\sigma} = e^{q_{\sigma}/b} / \sum_{\sigma' \in \mathcal{O}} e^{q_{\sigma'}/b}$ .

When  $|\mathcal{O}|$  is large or infinite, calculating the cost of a purchase becomes intractable in general. Recent research has focused on restricting the allowable securities over a combinatorial outcome space and examining whether LMSR prices can be computed efficiently in the restricted space. If the outcome space contains  $n!$  rank orders of  $n$  competing candidates, it is #P-hard to price pair bets (e.g., “\$1 if and only if candidate A beats candidate B”) or subset bets (e.g., “\$1 if one of the candidates in subset  $C$  finishes at position

<sup>1</sup>This no-arbitrage condition, while sometimes desirable, is not necessary. Othman et al. [28] recently analyzed a variation of LMSR in which  $\sum_{\sigma \in \mathcal{O}} p_{\sigma}(\mathbf{q}) \geq 1$ . We also explore relaxations of the no-arbitrage condition in Section 5.

$k$ ”) using LMSR on the full set of permutations [12]. If the outcome space contains  $2^n$  Boolean values of  $n$  binary base events, it is #P-hard to price securities on conjunctions of any two base events (e.g., “\$1 if and only if a Democrat wins Florida and Ohio”) using LMSR [12]. This line of research has led to some positive results when the uncertain event enforces particular structure on the outcome space. In particular, for a single-elimination tournament of  $n$  teams, securities such as “\$1 if and only if team A wins a  $k$ th round game” and “\$1 if and only if team A beats team B given they face off” can be priced efficiently using LMSR [13]. For a taxonomy tree on some statistic where the value of the statistic of a parent node is the sum of those of its children, securities such as “\$1 if and only if the value of the statistic at node A belongs to  $[x, y]$ ” can be priced efficiently using LMSR [19].

The framework we will introduce next takes a drastically different approach. Instead of searching for supportable spaces of securities for existing market makers, we design new market makers tailored to any security space of interest. Additionally, rather than requiring that securities have a fixed \$1 payoff when the underlying event happens, we allow more general contingent securities with arbitrary, efficiently computable and bounded payoffs.

### 3. A FRAMEWORK FOR MARKET-MAKING FOR COMPLEX SECURITIES

In the complete cost function based markets described above, the market maker offers an Arrow-Debreu security corresponding to each potential outcome. We consider a market-design scenario where the outcome space  $\mathcal{O}$  could potentially be quite large, or even infinite, making it infeasible to run such a market. Instead, we allow the market maker to offer a menu of  $K$  securities for some reasonably-sized  $K$ , with the (nonnegative) payoff of each security described by an arbitrary but efficiently-computable function  $\rho : \mathcal{O} \rightarrow \mathbb{R}_+^K$ . Specifically, if a trader purchases a share of security  $i$  and the outcome is  $\mathfrak{o}$ , then the trader is paid  $\rho_i(\mathfrak{o})$ . We call such security spaces *complex*. A complex security space reduces to the complete security space if  $K = |\mathcal{O}|$  and for each  $i \in \{1, \dots, K\}$ ,  $\rho_i(\mathfrak{o}) = 1$  if  $\mathfrak{o}$  is the  $i$ th outcome and 0 otherwise. We consider traders that purchase *security bundles*  $\mathbf{r} \in \mathbb{R}^K$ . Negative elements of  $\mathbf{r}$  encode sales. The payoff for  $\mathbf{r}$  upon outcome  $\mathfrak{o}$  is exactly  $\rho(\mathfrak{o}) \cdot \mathbf{r}$ , where  $\rho(\mathfrak{o})$  denotes the vector of payoffs for each security for outcome  $\mathfrak{o}$ . Let  $\rho(\mathcal{O}) := \{\rho(\mathfrak{o}) | \mathfrak{o} \in \mathcal{O}\}$ .

While it is easy to put intuitive constraints on market prices to ensure that a complete market is “well-behaved”, determining the comparable constraints on market prices for an arbitrary set of complex securities is challenging. In particular, while the no-arbitrage condition easily leads to the restriction that the security prices in a complete market need to sum to 1, it is much less clear what space of security prices should be allowed in order to prevent arbitrage in markets over complex securities. This is because the events under which complex securities will pay out are no longer mutually exclusive. This challenge leads us to take an axiomatic approach to understand what constraints a “well-behaved” market maker for complex securities must satisfy.

We do not presuppose a cost function based market. However, in Section 3.1, we show that the use of a convex cost function is *necessary* given the assumption of path indepen-

dence on the contract purchases. We then derive additional constraints on the form of this cost function. In Sections 3.2–3.4, we go on to show how to design an appropriate cost function by employing techniques from online convex optimization.

### 3.1 Imposing Some Natural Restrictions on the Market Maker

In this section we introduce a sequence of conditions or axioms that one might expect a market to satisfy, and show that these conditions lead to some natural mathematical restrictions on the costs of security bundles. (We consider relaxations of these conditions in Section 5.) Similar conditions were suggested for complete markets by Chen and Vaughan [10], who defined the notion of a *valid cost function*, and by Othman et al. [28], who discussed properties similar to our notions of path independence and expressiveness, among others. However, no similar conditions have been described for complex markets.

Imagine a sequence of traders entering the marketplace and purchasing security bundles. Let  $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots$  be the sequence of security bundles purchased. After  $t - 1$  such purchases, the  $t$ th trader should be able to enter the marketplace and query the market maker for the cost of arbitrary bundles. The market maker must be able to furnish a cost  $\text{Cost}(\mathbf{r} | \mathbf{r}_1, \dots, \mathbf{r}_{t-1})$  for any bundle  $\mathbf{r}$  given a previous trade sequence  $\mathbf{r}_1, \dots, \mathbf{r}_{t-1}$ . If the trader chooses to purchase  $\mathbf{r}_t$  at a cost of  $\text{Cost}(\mathbf{r}_t | \mathbf{r}_1, \dots, \mathbf{r}_{t-1})$ , the market maker may update the costs of each bundle accordingly. Our first condition requires that the cost of acquiring a bundle  $\mathbf{r}$  must be the same regardless of how the trader splits up the purchase.

**CONDITION 1 (PATH INDEPENDENCE).** *For any  $\mathbf{r}, \mathbf{r}'$ , and  $\mathbf{r}''$  such that  $\mathbf{r} = \mathbf{r}' + \mathbf{r}''$ , for any  $\mathbf{r}_1, \dots, \mathbf{r}_t$ ,*

$$\text{Cost}(\mathbf{r} | \mathbf{r}_1, \dots, \mathbf{r}_t) = \text{Cost}(\mathbf{r}' | \mathbf{r}_1, \dots, \mathbf{r}_t) + \text{Cost}(\mathbf{r}'' | \mathbf{r}_1, \dots, \mathbf{r}_t).$$

Path independence helps to reduce both arbitrage opportunities and the strategic play of traders, as they no longer need to reason about the optimal path leading to some target position. However, it is worth pointing out that there are interesting markets not satisfying this condition, for example, the continuous double auction (CDA) and the market maker for CDA considered by Das [15], which in our opinion deserve separate treatment.

It turns out that the path independence *alone* implies that prices can be represented by a cost function  $C$ , as illustrated in the following theorem. The proof is by induction on  $t$ .<sup>2</sup>

**THEOREM 1.** *Under Condition 1, there exists a cost function  $C : \mathbb{R}^K \rightarrow \mathbb{R}$  such that we may always write*

$$\begin{aligned} \text{Cost}(\mathbf{r}_t | \mathbf{r}_1, \dots, \mathbf{r}_{t-1}) \\ = C(\mathbf{r}_1 + \dots + \mathbf{r}_{t-1} + \mathbf{r}_t) - C(\mathbf{r}_1 + \dots + \mathbf{r}_{t-1}). \end{aligned}$$

With this in mind, we drop the cumbersome  $\text{Cost}(\mathbf{r} | \mathbf{r}_1, \dots, \mathbf{r}_t)$  notation from now on, and write the cost of a bundle  $\mathbf{r}$  as  $C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q})$ , where  $\mathbf{q} = \mathbf{r}_1 + \dots + \mathbf{r}_t$  is the vector of previous purchases.

<sup>2</sup>An appendix containing omitted proofs is available in the version of this paper posted on the authors’ websites.

Now, recall that one of the functions of a securities market is to aggregate traders' beliefs into an accurate prediction. Each trader may have his own (potentially secret) information about the future, which we represent as a distribution  $\mathbf{p} \in \Delta_{|\mathcal{O}|}$  over the outcome space, where  $\Delta_n = \{\mathbf{x} \in \mathbb{R}_{\geq 0}^n : \sum_{i=1}^n x_i = 1\}$ , the  $n$ -simplex. The pricing mechanism should therefore incentivize the traders to reveal  $\mathbf{p}$ , but simultaneously avoid providing arbitrage opportunities. Towards this goal, we introduce four additional conditions on our pricing mechanism.

The first condition ensures that the gradient of  $C$  is always well-defined. If we imagine that a trader can buy or sell an arbitrarily small bundle, we would like the cost of buying and selling an infinitesimally small quantity of any particular bundle to be the same. If  $\nabla C(\mathbf{q})$  is well-defined, it can be interpreted as a vector of instantaneous prices for each security, with  $\partial C(\mathbf{q})/\partial q_i$  representing the price per share of an infinitesimally small amount of security  $i$ . Additionally, we can interpret  $\nabla C(\mathbf{q})$  as the traders' current estimates of the expected payoff of each security, in the same way that  $\partial C(\mathbf{q})/\partial q_o$  was interpreted as the probability of outcome  $o$  when considering the complete security space.

**CONDITION 2 (EXISTENCE OF INSTANTANEOUS PRICES).**  $C$  is continuous and differentiable everywhere.

The next condition encompasses the idea that the market should react to trades in a sensible way in order to incorporate the private information of the traders. In particular, it says that the purchase of a security bundle  $\mathbf{r}$  should never cause the market to lower the price of  $\mathbf{r}$ . It turns out that this condition is closely related to incentive compatibility for a myopic trader. It is equivalent to requiring that a trader with a distribution  $\mathbf{p} \in \Delta_{|\mathcal{O}|}$  can never find it simultaneously profitable (in expectation) to buy a bundle  $\mathbf{r}$  or to buy the bundle  $-\mathbf{r}$ . In other words, there can not be more than one way to express one's information.

**CONDITION 3 (INFORMATION INCORPORATION).** For any  $\mathbf{q}$  and  $\mathbf{r} \in \mathbb{R}^K$ ,  $C(\mathbf{q}+2\mathbf{r}) - C(\mathbf{q}+\mathbf{r}) \geq C(\mathbf{q}+\mathbf{r}) - C(\mathbf{q})$ .

The no arbitrage condition states that it is never possible for a trader to purchase a security bundle  $\mathbf{r}$  and receive a positive profit regardless of the outcome.

**CONDITION 4 (NO ARBITRAGE).** For all  $\mathbf{q}, \mathbf{r} \in \mathbb{R}^K$ , there exists an  $o \in \mathcal{O}$  such that  $C(\mathbf{q}+\mathbf{r}) - C(\mathbf{q}) \geq \mathbf{r} \cdot \rho(o)$ .

Finally, the expressiveness condition specifies that any trader can set the market prices to reflect his beliefs about the expected payoffs of each security if arbitrarily small portions of shares may be purchased. Here  $\bar{\mathbb{R}}$  denotes the extended real numbers,  $[-\infty, \infty]$ .

**CONDITION 5 (EXPRESSIVENESS).** For any  $\mathbf{p} \in \Delta_{|\mathcal{O}|}$ ,  $\exists \mathbf{q} \in \mathbb{R}^K$  for which  $\nabla C(\mathbf{q}) = \mathbb{E}_{o \sim \mathbf{p}}[\rho(o)]$ .

For any subset  $S$  of  $\mathbb{R}^d$ , let  $\mathcal{H}(S)$  denote the convex hull of  $S$ . Recall that  $\rho(\mathcal{O}) := \{\rho(o) | o \in \mathcal{O}\}$ . We characterize the form of the cost function under these conditions.

**THEOREM 2.** If  $\mathcal{H}(\rho(\mathcal{O}))$  is closed,<sup>3</sup> then under Conditions 2-5,  $C$  must be convex with  $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} = \mathcal{H}(\rho(\mathcal{O}))$ .

<sup>3</sup>The closure assumption is not strictly necessary, but is perfectly natural in typical settings. We include it for convenience. Without it, the conclusion can be modified to  $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} = \text{closure}(\mathcal{H}(\rho(\mathcal{O})))$ .

**PROOF.** We first prove convexity. Assume  $C$  is non-convex somewhere. Then there must exist some  $\mathbf{q}$  and  $\mathbf{r}$  such that  $C(\mathbf{q}) > (1/2)C(\mathbf{q}+\mathbf{r}) + (1/2)C(\mathbf{q}-\mathbf{r})$ . This means  $C(\mathbf{q}+\mathbf{r}) - C(\mathbf{q}) < C(\mathbf{q}) - C(\mathbf{q}-\mathbf{r})$ , which contradicts Condition 3, so  $C$  must be convex.

Now, Condition 2 trivially guarantees that  $\nabla C(\mathbf{q})$  is well-defined for any  $\mathbf{q}$ . To see that  $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} \subseteq \mathcal{H}(\rho(\mathcal{O}))$ , let us assume there exists some  $\mathbf{q}'$  for which  $\nabla C(\mathbf{q}') \notin \mathcal{H}(\rho(\mathcal{O}))$ . This can be reformulated in the following way: There must exist some halfspace, defined by a normal vector  $\mathbf{r}$ , that separates  $\nabla C(\mathbf{q}')$  from every member of  $\rho(\mathcal{O})$ . More precisely

$$\nabla C(\mathbf{q}') \notin \mathcal{H}(\rho(\mathcal{O})) \iff \exists \mathbf{r} \forall o \in \mathcal{O} : \nabla C(\mathbf{q}') \cdot \mathbf{r} < \rho(o) \cdot \mathbf{r},$$

where the strict inequality is because  $\mathcal{H}(\rho(\mathcal{O}))$  is closed. On the other hand, letting  $\mathbf{q} := \mathbf{q}' - \mathbf{r}$ , we see by convexity of  $C$  that  $C(\mathbf{q}+\mathbf{r}) - C(\mathbf{q}) \leq \nabla C(\mathbf{q}') \cdot \mathbf{r}$ . Combining these last two inequalities, we see that the price of bundle  $\mathbf{r}$  purchased with history  $\mathbf{q}$  is always smaller than the payoff for *any* outcome. This implies that there exists some arbitrage opportunity, contradicting Condition 4.

Finally, since  $\mathcal{H}(\rho(\mathcal{O})) = \{\mathbb{E}_{o \sim \mathbf{p}}[\rho(o)] | \mathbf{p} \in \Delta_{|\mathcal{O}|}\}$ , Condition 5 implies that  $\mathcal{H}(\rho(\mathcal{O})) \subseteq \{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\}$ .  $\square$

What we have arrived at from the set of proposed conditions is that (a) the pricing mechanism is always described precisely in terms of a convex cost function  $C$  and (b) the set of reachable prices, that is the set  $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\}$ , must be *identically* the convex hull of the payoff vectors for each outcome  $\mathcal{H}(\rho(\mathcal{O}))$ . For complete markets, those with a single security corresponding to each of the  $n$  distinct outcomes, this would imply that the set of reachable prices should be the convex hull of the  $n$  standard basis vectors. Indeed, this comports exactly with the natural assumption that the vector of security prices in complete markets should represent a probability distribution, or equivalently that it should lie in the  $n$ -simplex.

## 3.2 Designing the Cost Function via Conjugate Duality

The natural conditions we introduced above imply that to design a market for a set of  $K$  securities with payoffs specified by an arbitrary payoff function  $\rho : \mathcal{O} \rightarrow \mathbb{R}_+^K$ , we should use a cost function based market with a convex, differentiable cost function such that  $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} = \mathcal{H}(\rho(\mathcal{O}))$ . We now provide a general technique that can be used to design and compare properties of cost functions that satisfy these criteria. In order to accomplish this, we make use of tools from convex analysis.

It is well known<sup>4</sup> that any closed, convex, differentiable function  $C : \mathbb{R}^K \rightarrow \mathbb{R}$  can be written in the form  $C(\mathbf{q}) = \sup_{\mathbf{x} \in \text{dom}(R)} \mathbf{x} \cdot \mathbf{q} - R(\mathbf{x})$  for a strictly convex function  $R$  called the *conjugate* of  $C$ , where  $\text{dom}(R)$  denotes the domain of  $R$ . (The strict convexity of  $R$  follows from the differentiability of  $C$ .) Furthermore, any function that can be written in this form is convex. As we will show in Section 3.3, the gradient of  $C$  can be expressed in terms of this conjugate:  $\nabla C(\mathbf{q}) = \text{argmax}_{\mathbf{x} \in \text{dom}(R)} \mathbf{x} \cdot \mathbf{q} - R(\mathbf{x})$ . To generate a convex cost function  $C$  such that  $\nabla C(\mathbf{q}) \in \Pi$  for all  $\mathbf{q}$  for some set

<sup>4</sup>For a detailed discussion of convex conjugates and their properties, refer to a good text on convex optimization such as Boyd and Vandenberghe [6] or Hiriart-Urruty and Lemaréchal [23].

$\Pi$ , it is therefore sufficient to choose an appropriate conjugate function  $R$ , restrict the domain of  $R$  to  $\Pi$ , and define  $C$  as

$$C(\mathbf{q}) = \sup_{\mathbf{x} \in \Pi} \mathbf{x} \cdot \mathbf{q} - R(\mathbf{x}). \quad (1)$$

We call such a market a *complex cost function based market*. To generate a cost function  $C$  satisfying our five conditions, we need only to set  $\Pi = \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$  and select a strictly convex function  $R$  to serve as the conjugate.

This method of defining  $C$  is convenient for several reasons. First, it leads to markets that are efficient to implement whenever  $\Pi$  can be described by a polynomial number of simple constraints. The difficulty with combinatorial outcome spaces is that actually enumerating the set of outcomes can be challenging or impossible. In our proposed framework we need only work with the convex hull of the payoff vectors for each outcome when represented by a low-dimensional payoff function  $\boldsymbol{\rho}(\cdot)$ . This has significant benefits, as one often encounters convex sets which contain exponentially many vertices yet can be described by polynomially many constraints. Moreover, as the construction of  $C$  is based entirely on convex programming, we reduce the problem of automated market making to the problem of optimization for which we have a wealth of efficient algorithms. Similar techniques have been applied to design learning algorithms in the online convex optimization framework [22, 31], where  $R$  plays the role of a regularizer, and have been shown to be efficient in a variety of combinatorial applications, including online shortest paths, online learning of perfect matchings, and online cut set [7].

Second, this method yields simple formulas for properties of markets that help us choose the best market to run. Two of these properties, worst-case monetary loss and worst-case information loss, are analyzed below.

Note that both the LMSR and Quad-SCPM [2] are examples of complex cost function based markets, though they are designed for the complete market setting only.

### 3.3 Bounding Market Maker Loss and Loss of Information

Before discussing market properties, it is useful to review some helpful properties of conjugates. The first is a convenient duality: For any convex, closed function  $C$ , the conjugate of the conjugate of  $C$  is  $C$  itself. This implies that if  $C$  is defined as in Equation 1, we may write

$$R(\mathbf{x}) = \sup_{\mathbf{q} \in \mathbb{R}^K} \mathbf{q} \cdot \mathbf{x} - C(\mathbf{q}).$$

Since this maximization is unconstrained, and we are assuming that our cost function  $C$  is differentiable, the maximum occurs when  $\nabla C(\mathbf{q}) = \mathbf{x}$ . (Note that this may hold for many different values of  $\mathbf{q}$ .) Suppose for a particular pair  $(\mathbf{x}^*, \mathbf{q}^*)$  we have  $\nabla C(\mathbf{q}^*) = \mathbf{x}^*$ . We can then rewrite this equation as  $R(\mathbf{x}^*) = \mathbf{q}^* \cdot \mathbf{x}^* - C(\mathbf{q}^*)$ , which gives us that  $C(\mathbf{q}^*) = \mathbf{q}^* \cdot \mathbf{x}^* - R(\mathbf{x}^*)$ . From Equation 1, this tells us that  $\mathbf{x}^*$  must be a maximizer of  $\mathbf{x} \cdot \mathbf{q} - R(\mathbf{x})$ . In fact, it is the unique maximizer due to strict convexity. This implies, as mentioned above, that  $\nabla C(\mathbf{q}) = \operatorname{argmax}_{\mathbf{x} \in \Pi} \mathbf{x} \cdot \mathbf{q} - R(\mathbf{x})$ .

By a similar argument we have that for any  $\mathbf{q}$ , if  $\nabla R(\mathbf{x}) = \mathbf{q}$  then  $\mathbf{x}$  maximizes  $\mathbf{x} \cdot \mathbf{q} - R(\mathbf{x})$  and therefore, as we have just shown,  $\mathbf{x} = \nabla C(\mathbf{q})$ . However, the fact that  $\mathbf{x} = \nabla C(\mathbf{q})$  does not imply that  $\nabla R(\mathbf{x}) = \mathbf{q}$ ; in the markets we consider, it is generally the case that  $\mathbf{x} = \nabla C(\mathbf{q})$  for multiple  $\mathbf{q}$ .

We will make use of the notion of Bregman divergence. The *Bregman divergence* with respect to a convex function  $f$  is given by

$$D_f(\mathbf{x}, \mathbf{y}) := f(\mathbf{x}) - f(\mathbf{y}) - \nabla f(\mathbf{y})(\mathbf{x} - \mathbf{y}).$$

It is clear by convexity that  $D_f(\mathbf{x}, \mathbf{y}) \geq 0$  for all  $\mathbf{x}$  and  $\mathbf{y}$ .

Additionally, we will require notions like the interior and boundary of a convex set, yet we must introduce more careful terminology in order to consider the case that our price space is not of full rank. Given a convex set  $S \subset \mathbb{R}^d$  that occupies  $d'$ -dimensional linear subspace, where potentially  $d' < d$ , we define the *relative interior* of  $S$ ,  $\operatorname{relint}(S)$ , which is the interior of  $S$  when restricted to the  $d'$ -dimensional space (typically known as the *affine hull* of  $S$ ). In addition, we define the *relative boundary* of  $S$ ,  $\operatorname{relbnd}(S)$ , as the set  $\operatorname{closure}(S) \setminus \operatorname{relint}(S)$ .

With these tools in hand, we are ready to examine some market properties, and see how they depend on  $R$ .

#### 3.3.1 Bounding the Market Maker's Monetary Loss

When comparing market mechanisms, it is useful to consider the market maker's worst-case monetary loss,

$$\sup_{\mathbf{q} \in \mathbb{R}^K} \left( \sup_{\mathbf{o} \in \mathcal{O}} (\boldsymbol{\rho}(\mathbf{o}) \cdot \mathbf{q}) - C(\mathbf{q}) + C(\mathbf{0}) \right).$$

This quantity is simply the worst-case difference between the maximum amount that the market maker might have to pay the traders ( $\sup_{\mathbf{o} \in \mathcal{O}} \boldsymbol{\rho}(\mathbf{o}) \cdot \mathbf{q}$ ) and the amount of money collected by the market maker ( $C(\mathbf{q}) - C(\mathbf{0})$ ). The following theorem provides a bound on this loss in terms of the conjugate function.

**THEOREM 3.** *Consider any complex cost function based market with  $\Pi = \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ . The worst-case monetary loss of the market maker is no more than*

$$\sup_{\mathbf{x} \in \boldsymbol{\rho}(\mathcal{O})} R(\mathbf{x}) - \inf_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}),$$

and when  $\mathbf{o}$  is such that  $\boldsymbol{\rho}(\mathbf{o}) = \operatorname{argmax}_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x})$  then this loss is realized in the limit for  $\mathbf{q}$  as  $\nabla C(\mathbf{q}) \rightarrow \boldsymbol{\rho}(\mathbf{o})$ .

This theorem tells us that as long as the conjugate function is bounded on  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ , the market maker's worst-case loss is also bounded<sup>5</sup>. It says further that this loss is actually realized, for a particular outcome  $\mathbf{o}$ , at least when the price vector approaches  $\mathbf{o}$ . This suggests that loss to the market maker is worst when the traders are the most certain about the outcome.

#### 3.3.2 Bounding Information Loss

Information loss can occur when securities are sold in discrete quantities (for example, single units), as they are in most real-world markets. Without the ability to purchase arbitrarily small bundles, traders may not be able to change the market prices to reflect their true beliefs about the expected payoff of each security, even if expressiveness is satisfied. We will argue that the amount of information loss is captured by the market's bid-ask spread

<sup>5</sup>In Section 5, we will state a more general, stronger bound on market maker loss capturing the intuitive notion that the market maker's profits should be higher when the distance between the final vector of prices and the payoff vector  $\boldsymbol{\rho}(\mathbf{o})$  of the true outcome  $\mathbf{o}$  is large; see Theorem 6.

for the smallest trading unit. Given some  $\mathbf{q}$ , the current bid-ask spread of security bundle  $\mathbf{r}$  is defined to be  $(C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q})) - (C(\mathbf{q}) - C(\mathbf{q} - \mathbf{r}))$ . This is simply the difference between the current cost of buying the bundle  $\mathbf{r}$  and the current price at which  $\mathbf{r}$  could be sold.

To see how the bid-ask spread relates to information loss, suppose that the current vector of quantities sold is  $\mathbf{q}$ . If securities must be sold in unit chunks, a rational, risk-neutral trader will not buy security  $i$  unless she believes the expected payoff of this security is at least  $C(\mathbf{q} + \mathbf{e}_i) - C(\mathbf{q})$ , where  $\mathbf{e}_i$  is the vector that has value 1 at its  $i$ th element and 0 everywhere else. Similarly, she will not sell security  $i$  unless she believes the expected payoff is at most  $C(\mathbf{q}) - C(\mathbf{q} - \mathbf{e}_i)$ . If her estimate of the expected payoff of the security is between these two values, she has no incentive to buy or sell the security. In this case, it is only possible to infer that the trader believes the true expected payoff lies somewhere in the range  $[C(\mathbf{q}) - C(\mathbf{q} - \mathbf{e}_i), C(\mathbf{q} + \mathbf{e}_i) - C(\mathbf{q})]$ . The bid-ask spread is precisely the size of this range.

The bid-ask spread depends on how fast instantaneous prices change as securities are bought or sold. Intuitively, the bid-ask spread relates to the depth of the market. When the bid-ask spread is large, new purchases or sales can change the prices of the securities dramatically; essentially, the market is shallow. When the bid-ask spread is small, purchases or sales may only move the prices slightly; the market is deep. Based on this intuition, for complete markets, Chen and Pennock [9] use the inverse of  $\partial^2 C(\mathbf{q}) / \partial q_i^2$  to capture the notion of market depth for each security  $i$  independently. In a similar spirit, we define a *market depth parameter*,  $\beta$ , for our complex securities markets with twice-differentiable  $C$ . Larger values of  $\beta$  correspond to deeper markets. We will bound the bid-ask spread in terms of this parameter, and use this parameter to show that there exists a clear tradeoff between worst-case monetary loss and information loss; this will be formalized in Theorem 4 below.

**DEFINITION 1.** *For any complex cost function based market, if  $C$  is twice-differentiable, the market depth parameter  $\beta(\mathbf{q})$  for a quantity vector  $\mathbf{q}$  is defined as  $\beta(\mathbf{q}) = 1/V_c(\mathbf{q})$ , where  $V_c(\mathbf{q})$  is the largest eigenvalue of  $\nabla^2 C(\mathbf{q})$ , the Hessian of  $C$  at  $\mathbf{q}$ . The worst-case market depth is  $\beta = \inf_{\mathbf{q} \in \mathbb{R}^K} \beta(\mathbf{q})$ .*

Let  $\text{relint}(\Pi)$  be the relative interior of  $\Pi$ . If  $C$  is twice-differentiable, then for any  $\mathbf{q}$  such that  $\nabla C(\mathbf{q}) \in \text{relint}(\Pi)$ , we have a correspondence between the Hessian of  $C$  at  $\mathbf{q}$  and the Hessian of  $R$  at  $\nabla C(\mathbf{q})$ . More precisely, we have that  $\mathbf{u}^\top \nabla^2 C(\mathbf{q}) \mathbf{u} = \mathbf{u}^\top \nabla^2 R(\nabla C(\mathbf{q})) \mathbf{u}$  for any  $\mathbf{u} = \mathbf{x} - \mathbf{x}'$  with  $\mathbf{x}, \mathbf{x}' \in \Pi$ . (See, for example, Gorni [17] for more.) This means that  $\beta(\mathbf{q})$  is equivalently defined as the smallest eigenvalue of  $\nabla^2 R(\nabla C(\mathbf{q}))|_{\Pi}$ ; that is, where we consider the second derivative only within the price region  $\Pi$ .

The definition of worst-case market depth implies that  $1/\beta$  is an upper bound on the curvature of  $C$ , which implies that  $C$  is locally bounded by a quadratic with Hessian  $1/\beta$ . We can derive the following.

**LEMMA 1.** *Consider a complex cost function based market with worst-case market depth  $\beta$ . For any  $\mathbf{q}$  and  $\mathbf{r}$  we have*

$$D_C(\mathbf{q} + \mathbf{r}, \mathbf{q}) \leq \frac{\|\mathbf{r}\|^2}{2\beta}.$$

It is easy to verify that the bid-ask spread can be written in terms of Bregman divergences. In particular,  $C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q}) - (C(\mathbf{q}) - C(\mathbf{q} - \mathbf{r})) = D_C(\mathbf{q} + \mathbf{r}, \mathbf{q}) + D_C(\mathbf{q} - \mathbf{r}, \mathbf{q})$ . This implies that the worst-case bid-ask spread of a market with market depth  $\beta$  can be upperbounded by a constant times  $1/\beta$ . That is, as the market depth parameter increases, the bid-ask spread must decrease. The following theorem shows that this leads to an inherent tension between worst-case monetary loss and information loss. Here  $\text{diam}(\mathcal{H}(\rho(\mathcal{O})))$  denotes the diameter of the hull of the payoff vectors for each outcome.

**THEOREM 4.** *For any complex cost function based market with worst-case market depth  $\beta$ , for any  $\mathbf{r}, \mathbf{q}$  meeting the conditions in Lemma 1, the bid-ask spread for bundle  $\mathbf{r}$  with previous purchases  $\mathbf{q}$  is no more than  $\|\mathbf{r}\|^2/\beta$ . The worst-case monetary loss of the market maker is at least  $\beta \cdot \text{diam}^2(\mathcal{H}(\rho(\mathcal{O}))) / 8$ .*

We can see that there is a direct tradeoff between the upper bound<sup>6</sup> of the bid-ask spread, which shrinks as  $\beta$  grows, and the lower bound of the worst-case loss of the market maker, which grows linearly in  $\beta$ . This tradeoff is very intuitive. When the market is shallow (small  $\beta$ ), small trades have a large impact on market prices, and traders cannot purchase too many shares of the same security without paying a lot. When the market is deep (large  $\beta$ ), prices change slowly, allowing the market maker to gain more precise information, but simultaneously forcing the market maker to take on more risk since many shares of a security can be purchased at prices that are potentially too low. This tradeoff can be adjusted by scaling  $R$ , which scales  $\beta$ . This is analogous to adjusting the “liquidity parameter”  $b$  of the LMSR.

### 3.4 Selecting a Conjugate Function

We have seen that the choice of the conjugate function  $R$  impacts market properties such as worst-case loss and information loss. We now explore this choice in more detail.

In many situations, the ideal choice of the conjugate is a function of the form

$$R(\mathbf{x}) := \frac{\lambda}{2} \|\mathbf{x} - \mathbf{x}_0\|^2. \quad (2)$$

Here  $R(\mathbf{x})$  is simply the squared Euclidean distance between  $\mathbf{x}$  and an initial price vector  $\mathbf{x}_0 \in \Pi$ , scaled by  $\lambda/2$ . By utilizing this quadratic conjugate function, we achieve a market depth that is uniformly  $\lambda$  over the entire security space. Furthermore, if  $\mathbf{x}_0$  is chosen as the “center” of  $\Pi$ , namely  $\mathbf{x}_0 = \arg \min_{\mathbf{x} \in \Pi} \max_{\mathbf{y} \in \Pi} \|\mathbf{x} - \mathbf{y}\|$ , then the worst-case loss of the market maker is  $\max_{\mathbf{x} \in \Pi} R(\mathbf{x}) = (\lambda/8) \text{diam}^2(\Pi)$ . While the market maker can tune  $\lambda$  appropriately according to the desired tradeoff between worst-case market depth and worst-case loss, the tradeoff is tightest when  $R$  has a Hessian that is uniformly a scaled identity matrix, or more precisely where  $R$  takes the form in Equation 2.

Unfortunately, by selecting a conjugate of this form, or any  $R$  with bounded derivative, the market maker does inherit one potentially undesirable property: security prices

<sup>6</sup>Strictly speaking, as we are emphasizing the necessary tradeoff between bid-ask spread and worst-case loss, we should have a *lower bound* on the bid-ask spread. On the other hand, if the worst-case market depth parameter is  $\beta$  then there is some  $\mathbf{q}$  and  $\mathbf{r}$  such that  $D_C(\mathbf{q} + \mathbf{r}, \mathbf{q}) / \|\mathbf{r}\|^2 \approx 1/(2\beta)$  and this approximation can be made arbitrarily tight for small enough  $\mathbf{r}$  when  $C$  is twice differentiable.

may become constant when  $\nabla C(\mathbf{q})$  reaches a point at  $\text{relbnd}(\Pi)$ , the relative boundary of  $\Pi$  (see Section 3.3). That is, if we arrive at a total demand  $\mathbf{q}$  where  $\nabla C(\mathbf{q}) = \rho(\mathbf{o})$  for some outcome  $\mathbf{o}$ , our mechanism begins offering securities at a price *equal to the best-case payoff*, akin to asking someone to bet a dollar for the chance to possibly win a dollar. The Quad-SCPM for complete markets is known to exhibit this behavior [2].

To avoid these undesirable pricing scenarios, it is sufficient to require that our conjugate function satisfy one condition. We say that a convex function  $R$  defined on  $\Pi$  is a *pseudo-barrier*<sup>7</sup> for  $\Pi$  if  $\|\nabla R(\mathbf{x}_t)\| \rightarrow \infty$  for any sequence of points  $\mathbf{x}_1, \mathbf{x}_2, \dots \in \Pi$  which tends towards  $\text{relbnd}(\Pi)$ . If we require our conjugate function  $R$  to be a pseudo-barrier, we are guaranteed that the instantaneous price vector  $\nabla C(\mathbf{q})$  always lies in  $\text{relint}(\Pi)$ , and does not become constant near the boundary.

It is important to note that, while it is desirable that  $\|\nabla R(\mathbf{x}_t)\| \rightarrow \infty$  as  $\mathbf{x}_t$  approaches  $\text{relbnd}(\Pi)$ , it is generally not desirable that  $R(\mathbf{x}_t) \rightarrow \infty$ . Recall that the market maker’s worst-case loss grows with the maximum value of  $R$  on  $\Pi$  and thus we should hope for a conjugate function that is bounded on the domain. A perfect example of convex function that is simultaneously bounded and a pseudo-barrier is the *negative entropy function*  $H(\mathbf{x}) = \sum_i x_i \log x_i$ , defined on the  $n$ -simplex  $\Delta_n$ . It is perhaps no surprise that LMSR, the most common market mechanism for complete security spaces, can be described by the choice  $R(\mathbf{x}) := \lambda H(\mathbf{x})$  where the price space  $\Pi = \Delta_n$  [2, 10].

## 4. EXAMPLES OF COMPUTATIONALLY EFFICIENT MARKETS

In the previous section, we provided a general framework for designing markets on combinatorial or infinite outcome spaces. We now provide some examples of markets that can be operated efficiently using this framework.

### 4.1 Subset Betting

Consider the scenario in which the outcome is some *ranking* of a set of  $n$  competitors, such as  $n$  horses in a race. The outcome of such a race is a permutation  $\pi : [n] \rightarrow [n]$ , where  $\pi(i)$  is the final position of  $i$ , with  $\pi(i) = 1$  being best, and  $[n]$  denotes the set  $\{1, \dots, n\}$ . A typical market for this setting might offer  $n$  Arrow-Debreu securities, with the  $i$ th security paying off if and only if  $\pi(i) = 1$ . Additionally, there might be separate, *independent* markets allowing bets on horses to place (come in first or second) or show (come in first, second, or third). However, running independent markets for sets of outcomes with clear correlations is wasteful in that information revealed in one market does not automatically propagate to the others.

Chen et al. [11] proposed a betting language, *subset betting* in which traders can place bets  $(i, j)$ , for any candidate  $i$  and any slot  $j$ , that pay out \$1 in the event that  $\pi(i) = j$  and \$0 otherwise.<sup>8</sup> Chen et al. [12] showed that pricing bets of this

<sup>7</sup>We use the term pseudo-barrier to distinguish this from the typical definition of a barrier function on a set  $\Pi$ , which is a function that grows without bound towards the boundary of  $\Pi$ . The term *Legendre* was introduced by Cesa-Bianchi and Lugosi [8] for a similar notion, yet their definition requires the stronger condition that  $\Pi$  contain a nonempty interior.

<sup>8</sup>The original definition of subset betting allowed bets of the form “any candidate in set  $S$  will end up in slot  $j$ ” or

form using LMSR is #P-hard and provided an algorithm for approximating the prices by exploiting the structure of the market. Using our framework, it is simple to design a computationally efficient market for securities of this form.

In order to set up such a combinatorial market within our framework, we must be able to efficiently work with the convex hull of the payoff vectors for each outcome. Notice that, for an outcome  $\pi$ , the associated payoff can be described by a matrix  $M_\pi$ , with  $M_\pi(i, j) = I[\pi(i) = j]$ , where  $I[\cdot]$  is the indicator function. Taking this one step further, it is easy to verify that the convex hull of the set of permutation matrices is precisely the set of *doubly stochastic matrices*, that is the set

$$\Pi = \left\{ X \in \mathbb{R}_{\geq 0}^{n \times n} : \sum_{i'=1}^n X(i', j) = \sum_{j'=1}^n X(i, j') = 1 \quad \forall i, j \right\},$$

where  $X(i, j)$  represents the element at the  $i$ th row and  $j$ th column of the matrix  $X$ . Notice, importantly, that this set is described by only  $n^2$  variables and  $O(n)$  constraints.

To fully specify the market maker, we must also select a conjugate function  $R$  for our price space. While the quadratic conjugate function is an option, there is a natural extension of the entropy function, whose desirable properties were discussed in the previous section, for the space of stochastic matrices. For any  $X \in \Pi$ , let us set

$$R(X) = \lambda \sum_{i,j} X(i, j) \log X(i, j).$$

The worst-case market depth is computed as the minimum of the smallest eigenvalue of the Hessian of  $R$  within the  $\text{relint}(\Pi)$ . This occurs at the matrix with all values  $1/n$ , hence the worst-case depth is  $n\lambda$ . The worst-case loss, on the other hand, is easily computed as  $\lambda n \log n$ .

### 4.2 Sphere Betting

We now consider an example in which the outcome space is infinite. An object orbiting the planet, perhaps a satellite, is predicted to fall to earth in the near future and will land at an unknown location, which we would like to predict. We represent locations on the earth as unit vectors  $\mathbf{u} \in \mathbb{R}^3$ . The difficulty of this example arises from the fact that the outcome must be a unit vector, imposing constraints on the three coordinates. We will design a market with three securities, each corresponding to one coordinate of the final location of the object. In particular, security  $i$  will pay off  $u_i + 1$  dollars if the object lands in location  $\mathbf{u}$ . (The addition of 1, while not strictly necessary, ensures that the payoffs, and therefore prices, remain positive, though it will be necessary for traders to sell securities to express certain beliefs.) This means that traders can purchase security bundles  $\mathbf{r} \in \mathbb{R}^3$  and, when the object lands at a location  $\mathbf{u}$ , receive a payoff  $(\mathbf{u} + \mathbf{1}) \cdot \mathbf{r}$ . Note that in this example, the outcome space is infinite, but the security space is small.

The price space  $\mathcal{H}(\rho(\mathcal{O}))$  for this market will be the 2-norm unit ball centered at  $\mathbf{1}$ . To construct a market for this scenario, let us make the simple choice of  $R(\mathbf{x}) = \lambda \|\mathbf{x} - \mathbf{1}\|^2$  for some parameter  $\lambda > 0$ . When  $\|\mathbf{q}\| \leq 2\lambda$ , there exists an  $\mathbf{x}$  such that  $\nabla R(\mathbf{x}) = \mathbf{q}$ . In particular, this is true for  $\mathbf{x} = (1/2)\mathbf{q}/\lambda + \mathbf{1}$ , and  $\mathbf{q} \cdot \mathbf{x} - R(\mathbf{x})$  is minimized at this “candidate  $i$  will end up in one of the slots in set  $S$ .” A bet of this form can be constructed easily using our betting language by bundling multiple securities.

point. When  $\|\mathbf{q}\| > 2\lambda$ ,  $\mathbf{q} \cdot \mathbf{x} - R(\mathbf{x})$  is minimized at an  $\mathbf{x}$  on the boundary of  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ . Specifically, it is minimized at  $\mathbf{x} = \mathbf{q}/\|\mathbf{q}\| + \mathbf{1}$ . From this, we can compute

$$C(\mathbf{q}) = \begin{cases} \frac{1}{4\lambda} \|\mathbf{q}\|^2 + \mathbf{q} \cdot \mathbf{1}, & \text{when } \|\mathbf{q}\| \leq 2\lambda, \\ \|\mathbf{q}\| + \mathbf{q} \cdot \mathbf{1} - \lambda, & \text{when } \|\mathbf{q}\| > 2\lambda. \end{cases}$$

The market depth parameter  $\beta$  is  $2\lambda$ ; in fact,  $\beta(\mathbf{x}) = 2\lambda$  for any price vector  $\mathbf{x}$  in the interior of  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ . By Theorem 3, the worst-case loss of the market maker is no more than  $\lambda$ , which is precisely the lower bound implied by Theorem 4. Finally, the divergence  $D_C(\mathbf{q} + \mathbf{r}, \mathbf{q}) \leq \|\mathbf{r}\|^2/(4\lambda)$  for all  $\mathbf{q}, \mathbf{r}$ , with equality when  $\|\mathbf{q}\|, \|\mathbf{q} + \mathbf{r}\| \leq 2\lambda$ , implying that the bid-ask spread scales linearly with  $\|\mathbf{r}\|^2/\lambda$ .

## 5. COMPUTATIONAL COMPLEXITY AND RELAXATIONS

In Section 3, we argued that the space of feasible price vectors should be precisely  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ , the convex hull of the payoff vectors for each outcome. In each of our examples, we have discussed market scenarios for which this hull has a polynomial number of constraints, allowing us to efficiently set prices via convex optimization<sup>9</sup>. Unfortunately, one should not necessarily expect that a given payoff function and outcome space will lead to an efficiently describable convex hull. In this section, we explore a couple of approaches to overcome such complexity challenges. First, we discuss the case in which  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$  has exponentially (or infinitely) many constraints yet gives rise to a *separation oracle*. Second, we show that the price space  $\Pi$  can indeed be *relaxed* beyond  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$  without increasing the risk to the market maker. Finally, we show how this relaxation applies in practice.

### 5.1 Separation Oracles

If we encounter a convex hull  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$  with exponentially-many constraints, all may not be lost. Recall that, in order to set prices, we need to solve the optimization problem  $\max_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} \mathbf{q} \cdot \mathbf{x} - R(\mathbf{x})$ . Under certain circumstances this can still be solved efficiently.

Consider a convex optimization problem with a concave objective function  $f(\mathbf{x})$  and constraints  $g_i(\mathbf{x}) \leq 0$  for all  $i$  in some index set  $I$ . That is, we want to solve:

$$\begin{aligned} \max \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{R}^d \\ & g_i(\mathbf{x}) \leq 0 \quad \forall i \in I \end{aligned}$$

This can be converted to a problem with a linear objective in the standard way:

$$\begin{aligned} \max \quad & c \\ \text{s.t.} \quad & \mathbf{x} \in \mathbb{R}^d, c \in \mathbb{R} \\ & f(\mathbf{x}) \geq c \\ & g_i(\mathbf{x}) \leq 0 \quad \forall i \in I \end{aligned}$$

Of course, if  $I$  is an exponentially or infinitely large set we will have trouble solving this problem directly. On the other

<sup>9</sup>A convex program can be solved with arbitrarily small error  $\epsilon$  in time polynomial of  $1/\epsilon$  and the size of the problem input using standard techniques such as the interior-point method. In this paper, we do not worry about finding the exact solution to the convex programs.

hand, the constraint set may admit an efficient *separation oracle*, defined as a function that takes as input a point  $(\mathbf{x}, c)$  and returns **true** if all the necessary constraints are satisfied or, otherwise, returns **false** and specifies a violated constraint<sup>10</sup>. Given an efficient separation oracle one has access to alternative methods for optimization, the most famous being Khachiyan's *ellipsoid method*, that run in polynomial time. For more details see, for example, Grötschel et al [18].

This suggests that a fruitful direction for designing computationally efficient market makers is to examine the pricing problem on an instance-by-instance basis, and for a particular instance of interest, leverage the structure of the instance to develop an efficient algorithm for solving the specific separation problem. We leave this for future research.

### 5.2 Relaxation of the Price Space

When dealing with a convex hull  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$  that has a prohibitively large constraint set and does not admit an efficient separation oracle we still have one tool at our disposal: we can modify  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$  to get an alternate price space  $\Pi$  which we *can* work with efficiently. Recall that in Section 3, we arrived at the requirement that  $\Pi = \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$  as a necessary conclusion of the proposed conditions on our market maker. If we wish to violate this requirement, we need to consider which conditions must be weakened and revise the resulting guarantees from Section 3.

We will continue to construct our markets in the usual way, via the tuple  $(\mathcal{O}, \boldsymbol{\rho}, \Pi, R)$  where  $\mathcal{O}$  is the outcome space,  $\boldsymbol{\rho}$  is the payoff function,  $\Pi \subseteq \mathbb{R}^d$  is a convex compact set of feasible prices, and  $R: \mathbb{R}^d \rightarrow \mathbb{R}$  is a strictly convex function with domain  $\Pi$ . The market's cost function  $C$  will be the conjugate of  $R$  with respect to the set  $\Pi$ , as in Equation 1. The only difference is that we now allow  $\Pi$  to be distinct from  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ . Not surprisingly, the choice of  $\Pi$  will affect the interest of the traders and the market maker. We prove several claims which will aid us in our market design. Theorem 5 tells us that the expressiveness condition should not be relaxed, while Theorem 6 tells us that the no-arbitrage condition can be. Together, these imply that we may safely choose  $\Pi$  to be a *superset* of  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ .

**THEOREM 5.** *For any complex cost function based market, the worst-case loss of the market maker is unbounded if  $\boldsymbol{\rho}(\mathcal{O}) \not\subseteq \Pi$ .*

This (perhaps surprising) theorem tells us that expressiveness is not only useful for information aggregation, it is actually *necessary* for the market maker to avoid unbounded loss. The proof involves showing that if  $\mathbf{o}$  is the final outcome and  $\boldsymbol{\rho}(\mathbf{o}) \notin \Pi$ , then it is possible to make an infinite sequence of trades such that each trade causes a constant amount of loss to the market maker.

In the following theorem, which is a simple extension of Theorem 3, we see that including additional price vectors in  $\Pi$  does not adversely impact the market maker's worst-case loss, despite the fact that the no-arbitrage condition is violated.

**THEOREM 6.** *Consider any complex cost function based market with  $R$  and  $\Pi$  satisfying  $\sup_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) < \infty$  and*

<sup>10</sup>More precisely, a separation oracle returns any *separating hyperplane* that divides the input from the feasible set.

$\mathcal{H}(\boldsymbol{\rho}(\mathcal{O})) \subseteq \Pi$ . Assume that the initial price vector satisfies  $\nabla C(\mathbf{0}) \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ . Let  $\mathbf{q}$  denote the vector of quantities sold and  $\mathbf{o}$  denote the true outcome. The monetary loss of the market maker is no more than

$$R(\boldsymbol{\rho}(\mathbf{o})) - \min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} R(\mathbf{x}) - D_R(\boldsymbol{\rho}(\mathbf{o}), \nabla C(\mathbf{q})).$$

This tells us that expanding  $\Pi$  can only help the market maker; increasing the range of  $\nabla C(\mathbf{q})$  can only increase the divergence term. This may seem somewhat counterintuitive. We originally required that  $\Pi \subseteq \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$  as a consequence of the no-arbitrage condition, and by relaxing this condition, we are providing traders with potential arbitrage opportunities. However, these arbitrage opportunities do not hurt the market maker. As long as the initial price vector lies in  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ , any such situations where a trader can earn a guaranteed profit are effectively created (and paid for) by other traders! In fact, if the final price vector  $\nabla C(\mathbf{q})$  falls outside the convex hull, the divergence term will be strictly positive, improving the bound.

To elaborate on this point, let's consider an example where  $\Pi$  is strictly larger than  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ . Let  $\mathbf{q}$  be the current vector of purchases, and assume the associated price vector  $\mathbf{x} = \nabla C(\mathbf{q})$  lies in the interior of  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ . Consider a trader who purchases a bundle  $\mathbf{r}$  such that the new price vector leaves this set, i.e.,  $\mathbf{y} := \nabla C(\mathbf{q} + \mathbf{r}) \notin \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ . We claim that this choice can be strictly improved in the sense that there is an alternative bundle  $\mathbf{r}'$  whose associated profit, for any outcome  $\mathbf{o}$ , is strictly greater than the profit for  $\mathbf{r}$ .

For simplicity, assume  $\mathbf{y}$  is an interior point of  $\Pi \setminus \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$  so that  $\mathbf{q} + \mathbf{r} = \nabla R(\mathbf{y})$ . Define  $\pi(\mathbf{y}) := \arg \min_{\mathbf{y}' \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} D_R(\mathbf{y}', \mathbf{y})$ , the minimum divergence projection of  $\mathbf{y}$  into  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ . The alternative bundle we consider is  $\mathbf{r}' = \nabla R(\pi(\mathbf{y})) - \mathbf{q}$ . Our trader pays  $C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q} + \mathbf{r}')$  less to purchase  $\mathbf{r}'$  than to purchase  $\mathbf{r}$ . Hence, for any outcome  $\mathbf{o}$ , we see that the increased profit for  $\mathbf{r}'$  over  $\mathbf{r}$  is

$$\begin{aligned} & \boldsymbol{\rho}(\mathbf{o}) \cdot (\mathbf{r}' - \mathbf{r}) - C(\mathbf{q} + \mathbf{r}') + C(\mathbf{q} + \mathbf{r}) \\ & > \boldsymbol{\rho}(\mathbf{o}) \cdot (\mathbf{r}' - \mathbf{r}) + \nabla C(\mathbf{q} + \mathbf{r}') \cdot (\mathbf{r} - \mathbf{r}') \\ & = (\boldsymbol{\rho}(\mathbf{o}) - \pi(\mathbf{y})) \cdot (\mathbf{r}' - \mathbf{r}). \end{aligned} \quad (3)$$

Notice that we achieve strict inequality precisely because  $\nabla C(\mathbf{q} + \mathbf{r}') \neq \nabla C(\mathbf{q} + \mathbf{r})$ . Now use the optimality condition for  $\pi(\mathbf{y})$  to see that, since  $\boldsymbol{\rho}(\mathbf{o}) \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ ,  $\nabla_{\pi(\mathbf{y})}(D_R(\pi(\mathbf{y}), \mathbf{y})) \cdot (\boldsymbol{\rho}(\mathbf{o}) - \pi(\mathbf{y})) \geq 0$ . It is easy to check that  $\nabla_{\pi(\mathbf{y})}(D_R(\pi(\mathbf{y}), \mathbf{y})) = \nabla R(\pi(\mathbf{y})) - \nabla R(\mathbf{y}) = \mathbf{r}' - \mathbf{r}$ . Combining this last expression with the inequality above and (3) tells us that the profit increase is strictly greater than  $(\boldsymbol{\rho}(\mathbf{o}) - \pi(\mathbf{y})) \cdot (\mathbf{r}' - \mathbf{r}) \geq 0$ . Simply put, the trader receives a guaranteed positive increase in profit for any outcome  $\mathbf{o}$ .

The next theorem shows that any time the price vector lies outside of  $\boldsymbol{\rho}(\mathbf{o})$ , traders could profit by moving it back inside. The proof uses a nice application of minimax duality for convex-concave functions.

**THEOREM 7.** *For any complex cost function based market, given a current quantity vector  $\mathbf{q}_0$  with current price vector  $\nabla C(\mathbf{q}_0) = \mathbf{x}_0$ , a trader has the opportunity to earn a guaranteed profit of at least  $\min_{\mathbf{x} \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))} D_R(\mathbf{x}, \mathbf{x}_0)$ .*

When  $\mathbf{x}_0 \in \mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ ,  $D_R(\mathbf{x}, \mathbf{x}_0)$  is minimized when  $\mathbf{x} = \mathbf{x}_0$  and the bound is vacuous, as we would expect. The more

interesting case occurs when the prices have fallen outside of  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ , in which case a trader is guaranteed a riskless profit by moving  $\nabla C(\mathbf{q})$  to the closest point in  $\mathcal{H}(\boldsymbol{\rho}(\mathcal{O}))$ .

### 5.3 Pair Betting via Relaxation

We return our attention to the scenario where the outcome is a ranking of  $n$  competitors, as described in Section 4.1. Consider a complex market in which traders make arbitrary *pair bets*: for every  $i, j$ , a trader can purchase a security which pays out \$1 whenever  $\pi(i) > \pi(j)$ . Like subset bets, pricing pair bets using LMSR is known to be #P-hard [12].

We can represent the payoff structure of any such outcome  $\pi$  by a matrix  $M_\pi$  defined by

$$M_\pi(i, j) = \begin{cases} 1, & \text{if } \pi(i) > \pi(j) \\ \frac{1}{2}, & \text{if } i = j \\ 0, & \text{if } \pi(i) < \pi(j). \end{cases}$$

We would like to choose our feasible price region as the set  $\mathcal{H}(\{M_\pi : \pi \in S_n\})$ , where  $S_n$  is the set of permutations on  $[n]$ . Unfortunately, the computation of this convex hull is necessarily hard: if given only a separation oracle for the set  $\mathcal{H}(\{M_\pi : \pi \in S_n\})$ , we could construct a linear program to solve the ‘‘minimum feedback arc set’’ problem, which is known to be NP-hard.

On the positive side, we see from the previous section that the market maker can work in a larger feasible price space without risking a larger loss. We thus relax our feasible price region  $\Pi$  to the set of matrices  $X \in \mathbb{R}^{n \times n}$  satisfying

$$\begin{aligned} X(i, j) & \geq 0 & \forall i, j \in [n] \\ X(i, j) & = 1 - X(j, i) & \forall i, j \in [n] \\ X(i, j) + X(j, k) + X(k, i) & \geq 1 & \forall i, j, k \in [n] \end{aligned}$$

This relaxation was first discussed by Meggido [26], who referred to such matrices as *generalized order matrices*. He proved that, for  $n \leq 4$ , we do have  $\Pi = \mathcal{H}(\{M_\pi : \pi \in S_n\})$ , but gave a counterexample showing strict containment for  $n = 13$ . By using this relaxed price space, the market maker allows traders to bring the price vector outside of the convex hull, yet includes a set of basic (and natural) constraints on the prices. Such a market could be implemented with any strongly convex conjugate function (e.g., quadratic).

## 6. CONCLUSION

We conclude by mentioning one nice direction of work. As we discussed, there is an inherent tradeoff between the bid-ask spread and the worst-case loss of the market maker. But if the market maker chooses to sell securities with an additional *transaction cost* for each security sold, then this money can not only help to cover the worst-case loss, but can also lead to a profit. Furthermore, if a market becomes popular, the market-maker may wish to increase the market depth. This idea has been explored by Othman et al. [28] for the case of complete markets, introducing a *liquidity sensitive* market maker, and they provide a new model with profit guarantees. Othman and Sandholm [27] recently extend the work and characterize a family of market makers that are liquidity sensitive. Via our framework, we can define an alternative method for simultaneously including transaction costs and guaranteeing profit. In particular, this is achieved through relaxing the price space, as discussed in Section 5.2. We leave the details to future work.

## 7. ACKNOWLEDGEMENTS

This material is based upon work supported by NSF under CNS-0937060 to the CRA for the CIFellows Project, CCF-0953516, NSF grant DMS-070706, DARPA grant FA8750-05-2-0249, and a Yahoo! PhD Fellowship. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors alone.

## References

- [1] S. Agrawal, Z. Wang, and Y. Ye. Parimutuel betting on permutations. In *Proceedings of the 4th International Workshop On Internet And Network Economics*, 2008.
- [2] S. Agrawal, E. Delage, M. Peters, Z. Wang, and Y. Ye. A unified framework for dynamic prediction market design. *Operations Research*, to appear, 2010.
- [3] K. J. Arrow. The role of securities in the optimal allocation of risk-bearing. *Review of Economic Studies*, 31(2):91–96, 1964.
- [4] K. J. Arrow. *Essays in the Theory of Risk Bearing*. North Holland, Amsterdam, 1970.
- [5] J. E. Berg, R. Forsythe, F. D. Nelson, and T. A. Rietz. Results from a dozen years of election futures markets research. In C. A. Plott and V. Smith, editors, *Handbook of Experimental Economic Results*. 2001.
- [6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [7] N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. In submission, January 2010.
- [8] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [9] Y. Chen and D. M. Pennock. A utility framework for bounded-loss market makers. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, 2007.
- [10] Y. Chen and J. Wortman Vaughan. A new understanding of prediction markets via no-regret learning. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, 2010.
- [11] Y. Chen, L. Fortnow, E. Nikolova, and D.M. Pennock. Betting on permutations. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 326–335. ACM, 2007.
- [12] Y. Chen, L. Fortnow, N. Lambert, D. M. Pennock, and J. Wortman. Complexity of combinatorial market makers. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, 2008.
- [13] Y. Chen, S. Goel, and D. M. Pennock. Pricing combinatorial markets for tournaments. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, 2008.
- [14] Yiling Chen, Lance Fortnow, Evdokia Nikolova, and David M. Pennock. Betting on permutations. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 326–335, 2007.
- [15] S. Das. The effects of market-making on price dynamics. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 887–894, 2008.
- [16] X. Gao, Y. Chen, and D. M. Pennock. Betting on the real line. In *Proceedings of the 5th Workshop on Internet and Network Economics*, 2009.
- [17] G. Gorni. Conjugation and second-order properties of convex functions. *Journal of Mathematical Analysis and Applications*, 158(2):293–315, 1991.
- [18] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. ISSN 0209-9683.
- [19] M. Guo and D. M. Pennock. Combinatorial prediction markets for event hierarchies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, 2009.
- [20] R. Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1):105–119, 2003.
- [21] R. Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, 1(1):3–15, 2007.
- [22] E. Hazan. A survey: The convex optimization approach to regret minimization. Draft, Sept. 2009.
- [23] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer, 2001.
- [24] J. Ledyard, R. Hanson, and T. Ishikida. An experimental test of combinatorial information markets. *Journal of Economic Behavior and Organization*, 69:182–189, 2009.
- [25] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomics Theory*. Oxford University Press, New York, NY, 1995.
- [26] N. Megiddo. Mixtures of order matrices and generalized order matrices. *Discrete Mathematics*, 19(2):177–181, 1977. ISSN 0012-365X.
- [27] A. Othman and T. Sandholm. Homogeneous risk measures and liquidity-sensitive automated market makers. Carnegie Mellon University working paper, March 2011.
- [28] A. Othman, T. Sandholm, D. M. Pennock, and D. M. Reeves. A practical liquidity-sensitive automated market maker. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, 2010.
- [29] D. M. Pennock and R. Sami. Computational aspects of prediction markets. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [30] M. Peters, A. Man-Cho So, and Y. Ye. Pari-mutuel markets: Mechanisms and performance. In *Proceedings of the 3rd International Workshop on Internet and Network Economics*, pages 82–95, 2007.
- [31] A. Rakhlin. Lecture notes on online learning. Draft, April 2009.
- [32] J. Wolfers and E. Zitzewitz. Prediction markets. *Journal of Economic Perspective*, 18(2):107–126, 2004.