Automated Market Makers That Enable New Settings: Extending Constant-Utility Cost Functions

Abraham Othman and Tuomas Sandholm

Computer Science Department, Carnegie Mellon University,
{aothman,sandholm}@cs.cmu.edu

Abstract. Automated market makers are algorithmic agents that provide liquidity in electronic markets. We construct two new automated market makers that each solve an open problem of theoretical and practical interest. First, we formulate a market maker that has bounded loss over separable measure spaces. This opens up an exciting new set of domains for prediction markets, including markets on locations and markets where events correspond to the natural numbers. Second, by shifting profits into liquidity, we create a market maker that has bounded loss in addition to a bid/ask spread that gets arbitrarily small with trading volume. This market maker matches important attributes of real human market makers and suggests a path forward for integrating automated market making agents into markets with real money.

1 Introduction

Automated market makers are algorithmic agents responsible for maintaining open interest in electronic markets, and they have been applied successfully to create new prediction markets that would not exist without the intervention of liquidity-providing agents. For instance, the Gates Hillman Prediction Market successfully forecast a fine-grained probability distribution over potential opening dates for the new computer science buildings at Carnegie Mellon University [Othman and Sandholm, 2010], and Yoopick was a Facebook application that allowed users to predict point spreads with far more detail than in traditional sports betting [Goel et al., 2008]. In this paper, we develop two new automated market makers that can be applied to new domains.

First, we design a loss-bounded automated market maker for separable measure spaces. Automated market making research has centered on finite discrete event spaces, but the market maker we present here is suitable for countably infinite discrete event spaces and uncountably infinite continuous event spaces. This market maker opens up exciting possibilities for valuable new markets. For instance, a prediction market over earthquakes in California or hurricanes in Florida could be run; traders could select ranges of locations and dates and magnitudes (all of these being continuous variables), and get paid if their predictions are correct. More importantly, administrators could use market prices to

get a better picture of disaster risks, providing vital insight that could improve preparation and planning.

We are also interested in expanding the use of automated market makers beyond mediating prediction markets (that typically use artificial currency). We would like to see automated market makers used in real markets with real money. As we saw in the recent "Flash Crash", where open interest in popular equities dropped precipitously [Sterling, 2010], it would be of immense value to have trading regulated by algorithmic agents with predictable behavior. The second market maker we design replicates the desirable features of existing human-mediated markets, but in an algorithmic way. The market maker can run at a profit and is appropriate for use in markets where the level of liquidity is not known *a priori.* Our new market maker reduces its bid/ask spread to zero with sufficient trading volume while still retaining a finite bound on loss. No existing market maker in the literature has both of these properties.

Though the problems we solve sound distinct, our solutions are very similar; both involve extending the same existing framework in different ways. We proceed with an introduction to automated market making, and then discuss each of our new market makers in turn.

2 Setting

To construct an automated market maker, the space of possible futures is exhaustively partitioned into a set of events Ω, such that exactly one of the $\omega \in \Omega$ will be realized. The bets traders make with the market maker are represented as functions from the event space to the real numbers, so that bets are contracts specifying how much the market maker will pay out if each of the various futures is realized. Typically, automated market makers use *cost functions* to price trades.

Definition 1 *Over n events, a* cost function *is a scalar field $C : \Re^n \mapsto \Re$ that maps a vector of amounts the market maker must pay out (the* obligation vector*) to a scalar value (the* cost*).*

The automated market maker keeps track of their obligation vector over the sequence of bets traders make. A trader who wishes to move the obligation vector from $\mathbf{x}$ to $\mathbf{y}$ is charged $C(\mathbf{y}) - C(\mathbf{x})$.

(In our notation, x is a scalar and $\mathbf{x}$ is a vector. x_i refers to the i-th element of a discrete vector, and $\mathbf{x}(t)$ refers to the evaluation of a vector function at point t (when we move to general vector spaces). ∇f is the gradient of the function f and $\mathbf{1}$ refers to a vector in which every component is 1.)

Over a series of transactions, the cost function telescopes from some initial $\mathbf{x^0}$ (which is often just set to $\mathbf{0}$). Consequently, the scalar difference between the current payout vector $\mathbf{x}$ and the initial payout vector $\mathbf{x^0}$, $C(\mathbf{x}) - C(\mathbf{x^0})$, is the amount traders have paid to the automated market maker. When an event ω_i is realized, the market maker's profit or loss is $C(\mathbf{x}) - C(\mathbf{x^0}) - x_i$. If, for all

$\mathbf{x}$ and all ω_i this quantity is bounded, we say the market maker has *bounded worst-case loss* (or just *bounded loss*). A review of the mechanics of automated market makers can be found in Pennock and Sami [2007].

Most automated market makers use cost functions that are *convex risk measures*:

Definition 2 *A* convex risk measure *is a cost function C that satisfies three properties:*

Monotonicity If $\mathbf{x} \leq \mathbf{y}$, then $C(\mathbf{x}) \leq C(\mathbf{y})$.
Convexity For all $\mathbf{x}$ and $\mathbf{y}$, and $\lambda \in [0, 1]$

$$C(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda C(\mathbf{x}) + (1 - \lambda)C(\mathbf{y}).$$

Translation invariance For all $\mathbf{x}$ and α, $C(\mathbf{x} + \alpha \mathbf{1}) = C(\mathbf{x}) + \alpha$.

Monotonicity ensures that traders cannot profit without taking on some risk, convexity avoids pricing degeneracies, and translation invariance means that buying or selling a guaranteed profit of a dollar costs a dollar.

Examples of convex risk measures in the literature include the LMSR [Hanson, 2003, 2007], sequential convex parimutuel mechanisms [Peters et al., 2007, Agrawal et al., 2009], market makers derived from no-regret online learning algorithms [Chen et al., 2008, Chen and Vaughan, 2010], and constant-utility cost functions [Chen and Pennock, 2007], the notion we extend in this work.

One alternative to convex risk measures is the liquidity-sensitive cost function of Othman et al. [2010], which we discuss later in this paper.

Market makers that do not use cost functions at all include heuristic market makers [Das, 2008, Brahma et al., 2010] (which do not rule out manipulability), subsidized dynamic pari-mutuel mechanisms [Pennock, 2004] (which do not offer guaranteed payoffs), and self-financing static pari-mutuel mechanisms [Lambert et al., 2008] (which do not offer guaranteed payoffs or sequential interactions with a market maker). We will not discuss these alternatives to cost-function-based market makers further in this paper.

2.1 Constant-utility cost functions

In this section we describe *constant-utility cost functions*, an existing framework for building cost functions that was introduced by Chen and Pennock [2007]. Our new market makers are created by extending this framework.

Definition 3 *A* utility function *is an increasing concave function $u : \Re \mapsto \Re$.*

A constant-utility cost function works by charging traders the amount that keeps the market maker's utility at a constant level. Put another way, the market maker prices each bet so that he is indifferent between a trader declining and accepting it.

Definition 4 *Let $x^0 \in \mathsf{dom}\, u$ and let μ be a probability distribution over the ω_i, defining $p_i \equiv \mathbb{P}_\mu(\omega_i)$. A constant-utility cost function $C : \Re^n \mapsto \Re$ is defined implicitly as the solution to*

$$\sum_i p_i u(C(\mathbf{x}) - x_i) = u(x^0)$$

Since the cost function is given implicitly by a root-finding problem, rather than explicitly as a function of the vector $\mathbf{x}$, it is not immediately clear that costs exist, are unique, and are easily computable. However, because the utility function is monotonically increasing, the cost function exists for any input vector and is unique. Furthermore, because the utility function is increasing, we can compute b bits of the value of the cost function in b steps of a binary search over possible values.

The partial derivatives of the cost function are known as *prices*. Where u is differentiable the prices of a constant-utility cost function have a closed form [Jackwerth, 2000, Chen and Pennock, 2007]:

Where u is differentiable the prices of a constant-utility cost function form a probability distribution:

$$\nabla_i C(\mathbf{x}) = \frac{p_i u'(C(\mathbf{x}) - x_i)}{\sum_j p_j u'(C(\mathbf{x}) - x_j)}.$$

The prices form a probability distribution because they sum to one and are non-negative (the utility function is non-decreasing and so its derivative is never negative).

We are particularly interested in a special class of utility functions that we dub *log-like*.

Definition 5 *A* log-like *utility function u is a differentiable utility function defined over $(0, \infty)$ that is unbounded from below from the right:* $\lim_{x \to 0^+} u(x) = -\infty$.

One example of a log-like utility function is log, and another is $-1/x$. The key property of log-like utility functions that we exploit is the extreme negative weight on input values close to zero.

3 Market making over measure spaces

In this section we extend constant-utility cost functions so that they apply over arbitrary measure spaces, and demonstrate how a particular class of utility functions over separable measure spaces has bounded loss. What makes this challenging is the tension between loss boundedness and sometimes offering traders bets that would be irrational to accept regardless of what event materializes. It is easy to create market makers with bounded loss if they charge agents as much as they could possibly win. On the other hand, it is also easy to create market makers that only offer bets a rational agent could accept, but have unbounded

loss. For instance, the most popular automated market maker in practice is the LMSR, which for n events has worst-case loss $\Theta(\log n)$. Over infinite event spaces, however, it has unbounded loss [Gao et al., 2009].

Because of this tension, it has been an open problem whether there exist market makers with bounded loss over arbitrary measure spaces that never offer agents bets which are immediately bad. In this section, we construct such a market maker. Our only restriction is that the spaces must be separable, a technical condition that is met naturally by all spaces of interest—such as the real line or the natural numbers, as well as geometric shapes like cubes and spheres.

3.1 Measure spaces and separability

We generalize the usual notation from settings with a finite number of events to settings with potentially an infinite number of events. In this context, the events are $t \in \Omega$, and bets still map from events to how much the trader gets paid based on what event materializes. However, bets and payout vectors (which were represented as points in $\Re^n$ in the finite case) are now functions over the event space; formally, $\mathbf{x} : \Omega \mapsto \Re$.

The cost function from a finite-event setting now generalizes to be a functional that maps these functions $\mathbf{x}$ to scalar values, i.e., $C : (\Omega \mapsto \Re) \mapsto \Re$. The notational similarity with the traditional setting that has only a finite number of events can be seen later because sums are simply replaced by integrals and probability mass functions are replaced by probability density functions.

We assume that the event space is *separable*, meaning that it possesses a natural notion of distance. With a separable event space we can assert the existence of a probability distribution over the events that has positive measure for every event, even though the space of events could be infinite. Formally, the following result is well known in measure theory.

Proposition 1 *In any separable measure space, there exists a probability measure μ such that $\mu(t) > 0$ for all t.*

We call such a μ an *always-positive measure*. Over the real line, an example of an always-positive measure is a Gaussian distribution. Measure spaces that are not separable are highly degenerate and are unlikely to be of practical importance.

3.2 Market making without offering bad bets

The aim of this section is to devise a market maker that has bounded loss over separable measure spaces that does not offer agents bets which are immediately bad. This latter quality is important because it is simple to construct a market maker with bounded loss over arbitrary measure spaces. The sup cost function (which can be thought of as an arbitrary-dimensional max)

$$C(\mathbf{x}) = \sup_t \mathbf{x}(t)$$

accomplishes this goal. This is easy to see because a trader is charged as much as she could possibly gain from the realization of any event. However, sup is one example of a cost function that *offers bets which are immediately bad.*

Definition 6 *A differentiable convex risk measure C offers bets which are immediately bad if there exists an obligation vector* **x** *where a proper subset of the event space has prices that sum to at least 1.*

As the name implies, trade with a cost function of this sort can sometimes be a dominated action: a trader pays at least as much as she could possibly make from any contract, but there also exist events for which the trader loses her bet, too.

Agrawal et al. [2009] present several cost functions with a worst-case loss of $\Theta(1 - 1/n)$, which implies bounded loss as n gets large. However, they achieve their bounded worst-case loss by offering good bets only up to a certain amount of uncovered exposure. After reaching this limit (which is controlled by a parameter set *a priori* by the market administrator) the price of some events is set to zero, implying that the other events form a proper subset with unit marginal price. This makes the market maker offer bets which are immediately bad. (Furthermore, the market makers in Agrawal et al. [2009] rely on solving a relatively complex convex optimization, so it is not immediately clear how to generalize their technique over infinite-dimensional spaces, or whether such a generalization is even possible.)

3.3 Our loss-bounded market makers

We now show that using a constant-utility cost function with a log-like utility function and an always-positive measure produces a market maker with bounded loss over separable measure spaces that never offers bets to agents which are immediately bad. These are the first market makers to have both of these properties.

These market makers could be used over continuous spaces, for instance, to create time and location markets for natural disasters like earthquakes, tornadoes, and hurricanes. Furthermore, events that take place over unbounded discrete sets (such as population sizes, interest rates, or stock prices) can also be handled effectively with these market makers because countable sets are also separable measure spaces.

Our market maker requires solving an integral equation over the measure space, and so the feasibility of our scheme relies on how easy it is to evaluate the integral. We suggest that while it is possible to describe degenerate cases where it is not possible to compute this integral effectively, for natural domains like geometric shapes or number lines the integral can be computed using numerical techniques in a straightforward manner.

Proposition 2 *Let u be a log-like utility function, μ be an always-positive measure, and $x^0 > 0$. Then the constant-utility cost function given by the solution to*

$$\int_t u(C(\mathbf{x}) - \mathbf{x}(t))\mu(t)\,dt = u(x^0)$$

loses at most x^0 and never offers bets which are immediately bad.

Proof. To prove the bounded loss property, it suffices to show that there exists no $\mathbf{x}$ and t such that $\mathbf{x}(t) - C(\mathbf{x}) > x^0$, because if such an $\mathbf{x}$ and t existed it would imply that by moving the market maker to the payoff vector $\mathbf{x}$ and realizing the event t the market maker would lose more than x^0. Suppose there did exist such an $\mathbf{x}$ and t. By definition, evaluating the integrand at t would involve evaluating

$$u(C(\mathbf{x}) - \mathbf{x}(t)) = -\infty$$

because the utility function is log-like. Since the probability measure is always positive, $\mu(t) > 0$ and so the resulting integral is undefined rather than $u(x^0)$, a contradiction.

To prove the resulting cost function never offers bets which are immediately bad, consider that over a separable measure space a differentiable convex risk measure offer immediately bad bets if there exists a proper subset of the space A and an obligation vector $\mathbf{x}$ such that

$$\int_A \nabla C(\mathbf{x})\mu(t)\,dt = 1$$

while

$$\int_A \mu(t)\,dt \neq 1$$

For there to not exist such an $A, \mathbf{x}$ pair, it is suffices that for all $\mathbf{x}$ and t, $\nabla C(\mathbf{x})(t)\mu(t) > 0$ or, because μ is an always positive measure $\nabla C(\mathbf{x})(t) > 0$ for all $\mathbf{x}$ and t.

Since C is a constant-utility cost function, recall the closed-form expression for its gradient (which over general measure spaces is a function)

$$\nabla C(\mathbf{x})(t) = \frac{\mu(t)u'(C(\mathbf{x}) - \mathbf{x}(t))}{\int u'(C(\mathbf{x}) - \mathbf{x}(t))\mu(t)\,dt}$$

Because the utility function is strictly increasing and differentiable, its derivative exists and is always positive. Coupling this with the fact that μ is an always positive measure, we have that the numerator

$$\mu(t)u'(C(\mathbf{x}) - \mathbf{x}(t)) > 0$$

and so $\nabla C(\mathbf{x})(t) > 0$ for all $\mathbf{x}$ and t. Consequently, there never exists a proper subset A of the space that has unit marginal price. Thus, the cost function never offers bets that are immediately bad. ∎

4 Funneling profits into liquidity

In this section we construct a cost-function-based market maker with a fixed worst-case loss that also shrinks its bid/ask spread to zero (provided there is sufficient transaction volume). In these characteristics, and also in its ability to realize a profit, the market maker we describe in this section closely matches human market makers we see operating in the real world. This suggests that our automated market maker could prove to be a practical choice for markets with real money.

We achieve these properties through realizing the notion of "driving profits toward liquidity", an idea that has never been successfully and fully implemented in an automated market maker. Probably the closest prior effort is the market maker of Othman et al. [2010], which features a liquidity-sensitive market maker with a shrinking bid/ask spread. However, that market maker can only shrink its bid/ask spread to a positive constant determined *a priori*, not arbitrarily small.

4.1 Description of the problem

Liquidity is the amount of active interest in a market. It is intimately related to the *bid/ask spread*, or the difference between the price to buy and the price to sell a contract. Real world markets are characterized by an inverse relationship between transaction volumes and bid/ask spreads. *Deep* (aka *liquid*) markets have small bid/ask spreads; for instance, large cap equities generally have one-cent bid ask spreads for thousands of dollars of interest. These markets can sustain large trades without prices moving significantly. In contrast, a *shallow* (aka *illiquid*) market is characterized by large bid/ask spreads and prices that move significantly in response to trades. At an extreme, a very illiquid market may not have any open interest at all.

From an automated market making perspective, there exists a sharp tension between the sensitivity of the market maker to liquidity and the amount that the market maker can lose. Consider the case of a market maker that prices bets according to some probability distribution over the future state of the world, and does not change that distribution in response to the bets placed by traders. Such a market maker always has a bid/ask spread of zero, but also has unbounded worst-case loss. In contrast, the bounded-loss convex risk measures in the literature always have non-vanishing, positive bid/ask spreads for fixed-size bets.

We can also come up with more sophisticated variations on these schemes in an effort to shrink bid/ask spreads, but these also do not satisfy our desiderata. Consider a market maker operating based on a fixed probability distribution who also takes a diminishing profit cut, a notion we will mathematically formalize later in this section. With such a scheme, the bid/ask spread falls to zero with increasing liquidity, but the market maker still has unbounded worst-case loss. On the other hand, a standard bounded-loss convex risk measure equipped with a diminishing profit cut has bounded worst-case loss and a shrinking bid/ask spread, but this spread never goes away even as liquidity gets large.

What has not been resolved is the design of a market maker that (1) has bounded loss and (2) has a bid/ask spread that becomes arbitrarily small with increasing liquidity. In this section, we build such a market maker.

4.2 Diminishing profit cuts

In this section we introduce a framework for a market maker adding an additional charge to trades on top of a cost function. While it is easy to, for example, charge a trader 10 cents extra per transaction, making a profit cut that vanishes with liquidity requires more sophistication.

Recall that a constant-utility cost function solves

$$\sum_i p_i u(C(\mathbf{x}) - x_i) = u(x^0)$$

Now consider how to add profits into this calculation. The *profit function* (or *cut*) $\Pi : \Re_+ \mapsto \Re_+$, is a continuous increasing function that keeps track of the amount of profit we have collected. Π relies on the current state s; at the market's initiation we set $s = 0$, and since Π tracks profit, $\Pi(0) \equiv 0$. When the obligation vector of the market moves from $\mathbf{x}$ to $\mathbf{y}$ we set $s = s + ||\mathbf{y} - \mathbf{x}||$.

The following algorithm details how our market maker prices trades as the sum of the cost function and the profit function.

1. We are currently at obligation vector $\mathbf{x}$ and profit state s. A trader wishes to take on a transaction that would leave the market maker with the obligation vector $\mathbf{y}$.
2. The value of the cost function, $C(\mathbf{y})$, is solved through

$$\sum_i p_i u(C(\mathbf{y}) - y_i + \Pi(s + ||\mathbf{y} - \mathbf{x}||)) = u(x^0)$$

   (In this equation, $C(\mathbf{y})$ is the variable and all of the other values are constants.)
3. The total transaction cost quoted to the trader (the *quoted price*) is

$$C(\mathbf{y}) - C(\mathbf{x}) + \Pi(s + ||\mathbf{y} - \mathbf{x}||) - \Pi(s)$$

4. If the trade is accepted, the current profit state is updated, $s \doteq s + ||\mathbf{y} - \mathbf{x}||$.

In order to design our new market maker, we require our profit function to meet some additional requirements. First, it must not have a bound on the total amount it collects $\lim_{s \to \infty} \Pi(s) = \infty$. Second, the marginal profit cut must vanish as liquidity gets large; for differentiable Π this condition is $\lim_{s \to \infty} \Pi'(s) = 0$.

There are many functions that satisfy these conditions. Any such function would constitute a market maker in our desired class, and the propositions in the rest of this section would go through. However, in what follows, we will focus on one particularly simple such function: $\Pi(s) = \alpha \log(s + 1)$ for some $\alpha > 0$. The corresponding marginal profit cut is $\Pi'(s) = \frac{\alpha}{s+1}$.

Equipped with this profit side, we can now introduce our market maker.

4.3 Our new market maker

Our market maker uses a constant-utility cost function with the profit cut framework we described above. The utility function is $u(x) = \log(x)$ and the profit function is $\Pi(s) = \alpha \log(s + 1)$.

The market administrator has two parameters to set: the $x^0 > 0$ chosen to initially seed the constant utility level, and the $\alpha > 0$ chosen to moderate the profit function. Higher x^0 leads to higher worst-case loss but more initial liquidity-sensitivity. Higher α makes the profit cut larger.

Our market maker has loss bounded by its initial cost seed:

Proposition 3 *Let $\min_i p_i > 0$, so the probability mass function is always positive. Then our market maker loses no more than the x^0 chosen to initially seed the constant utility level $\log x^0$.*

Proof. Because we are using a log-like utility function and the probability mass function is never negative, the costs, profits, and obligation vector always obey the relationship $C(\mathbf{x}) + \Pi(s) > \max_i x_i$ because if this were not the case the sum used to calculate expected utility would be undefined. But our loss if the market expires at state $(\mathbf{x}, s)$ is at most

$$\max_i x_i + C(\mathbf{0}) - C(\mathbf{x}) - \Pi(s)$$

Consequently, loss can be no greater than $C(\mathbf{0})$, which for constant-utility cost functions is just x^0 by definition. ∎

Our market maker's bid/ask spread becomes arbitrarily small with increasing activity:

Proposition 4 *The bid/ask spread of our market maker becomes arbitrarily small as $s \to \infty$.*

Proof. We prove that as $s \to \infty$, the quoted price of any fixed trade can be arbitrarily well-approximated by a linear extrapolation from the marginal prices of the cost function. Since the cost function is a differentiable convex risk measure, these marginal prices sum to unity, and so the bid/ask spread becomes arbitrarily small as both the buy and sell sides approach the same quoted prices.

Let the market maker be in state $(\mathbf{x}, s)$, let $\mathbf{y}$ be some trade, and let C_a denote a logarithmic constant-utility cost function seeded with initial cost a. Since profits are funneled into liquidity, we have that the quoted price to move to state $(\mathbf{x} + \mathbf{y}, s + ||\mathbf{y} - \mathbf{x}||)$ is no larger than

$$C_{x^0 + \Pi(s)}(\mathbf{x} + \mathbf{y}) - C_{x^0 + \Pi(s)}(\mathbf{x}) + \Pi(s + ||\mathbf{y} - \mathbf{x}||) - \Pi(s)$$

Now consider each pair of these terms in turn as s grows large. For the cost function terms, a logarithmic constant-utility market maker with large initial cost becomes increasingly insensitive to the second-order effects (i.e., changing prices) when taking on a fixed size bet. This is because utility grows arbitrarily

large in initial cost, while marginal utility shrinks arbitrarily small in initial cost. Consequently, since Π gets arbitrarily large in s we have that for large s

$$C_{x^0 + \Pi(s)}(\mathbf{x} + \mathbf{y}) - C_{x^0 + \Pi(s)}(\mathbf{x}) \approx \nabla C_{x^0 + \Pi(s)}(\mathbf{x}) \cdot \mathbf{y}$$

or, that the difference in costs can be arbitrarily well-approximated by a linear extrapolation from marginal prices.

As for the Π terms, by construction the derivative of the profit cut Π' gets arbitrarily small in s, so for large s the difference of the two terms is arbitrarily small. Furthermore, the upper bound on the quoted price of

$$C_{x^0 + \Pi(s)}(\mathbf{x} + \mathbf{y}) - C_{x^0 + \Pi(s)}(\mathbf{x}) + \Pi(\mathbf{x} + \mathbf{y}) - \Pi(\mathbf{x})$$

gets arbitrarily tight as s gets large, because as profits get arbitrarily small, the impact of adding them back into the cost function of our market maker gets arbitrarily small, too.

Putting all of these arguments together, the quoted price can be approximated arbitrarily well as s gets large by $\nabla C_{x^0 + \Pi(s)}(\mathbf{x}) \cdot \mathbf{y}$, which is a linear extrapolation of the marginal prices of the cost function. ∎

Acknowledgments

References

S. Agrawal, E. Delage, M. Peters, Z. Wang, and Y. Ye. A unified framework for dynamic pari-mutuel information market design. In *ACM Conference on Electronic Commerce (EC)*, pages 255–264, 2009.

A. Brahma, S. Das, and M. Magdon-Ismail. Comparing Prediction Market Structures, With an Application to Market Making. Technical report, Dept. of Computer Science, Rensselaer Polytechnic Institute, 2010.

Y. Chen and D. M. Pennock. A utility framework for bounded-loss market makers. In *Proceedings of the 23rd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 49–56, 2007.

Y. Chen and J. W. Vaughan. A new understanding of prediction markets via no-regret learning. In *ACM Conference on Electronic Commerce (EC)*, pages 189–198, 2010.

Y. Chen, L. Fortnow, N. Lambert, D. M. Pennock, and J. Wortman. Complexity of combinatorial market makers. In *ACM Conference on Electronic Commerce (EC)*, pages 190–199, 2008.

S. Das. The effects of market-making on price dynamics. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 887–894, 2008.

X. Gao, Y. Chen, and D. M. Pennock. Betting on the real line. In *International Workshop On Internet And Network Economics (WINE)*, pages 553–560, 2009.

S. Goel, D. Pennock, D. Reeves, and C. Yu. Yoopick: a combinatorial sports prediction market. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1880–1881, 2008.

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

R. Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, 1(1):1–15, 2007.

J. Jackwerth. Recovering risk aversion from option prices and realized returns. *Review of Financial Studies*, 13(2):433, 2000.

N. Lambert, J. Langford, J. Wortman, Y. Chen, D. Reeves, Y. Shoham, and D. Pennock. Self-Financed Wagering Mechanisms for Forecasting. In *ACM Conference on Electronic Commerce (EC)*, pages 170–179, 2008.

A. Othman and T. Sandholm. Automated market-making in the large: the Gates Hillman prediction market. In *ACM Conference on Electronic Commerce (EC)*, pages 367–376, 2010.

A. Othman, D. M. Pennock, D. M. Reeves, and T. Sandholm. A practical liquidity-sensitive automated market maker. In *ACM Conference on Electronic Commerce (EC)*, pages 377–386, 2010.

D. Pennock and R. Sami. Computational Aspects of Prediction Markets. In *Algorithmic Game Theory*, chapter 26, pages 651–674. Cambridge University Press, 2007.

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

M. Peters, A. M.-C. So, and Y. Ye. Pari-mutuel markets: Mechanisms and performance. In *International Workshop On Internet And Network Economics (WINE)*, pages 82–95, 2007.

B. Sterling. Flash Crash. http://www.wired.com/beyond_the_beyond/2010/05/flash-crash/, 2010.