Pari-mutuel Markets: Mechanisms and Performance

Mark Peters, Yinyu Ye
Stanford University

Anthony Man–Cho So
The Chinese University of Hong Kong

May 9, 2008

Abstract

Recently, there has been an increase in the usage of centrally managed markets which are run by some form of pari-mutuel mechanism. A pari-mutuel mechanism is characterized by the ability to shield the market organizer from financial risk by paying the winners from the stakes of the losers. The recent introduction of new, modified pari-mutuel methods has spurred the growth of prediction markets as well as new financial derivative markets. Coinciding with this increased usage, there has been much work on the research front which has produced several mechanisms and a slew of interesting results. We will introduce a new pari-mutuel market-maker mechanism with many positive qualities including convexity, truthfulness and strong performance. Additionally, we will provide the first quantitative performance comparison of some of the existing pari-mutuel market-maker mechanisms.

1 Introduction

While pari-mutuel systems have long been one of the most popular means of organizing markets, recent innovations have created more applications for pari-mutuel market-making techniques. Lately, there has been a substantial increase in the number of markets being conducted by means of pari-mutuel techniques. It appears that this growth has been driven by the introduction of novel, non-standard pari-mutuel mechanisms that more easily facilitate the launch of a new market. While the mechanisms employed vary from market to market, they share the common bond of utilizing the pari-mutuel principle of paying the winners from the stakes collected from the losers.

The standard pari-mutuel market was developed in 1864 and is operated in a manner where market traders purchase shares for a specific possible outcome.

When the outcome is determined, the money collected is paid out to the winners in proportion to the number of winning shares that they hold. This technique protects the market organizer from sustaining a loss under any circumstance. Some of the earliest work in the development of modified pari-mutuel techniques was done by Bossaerts et al. (2) where the authors study a continuous double auction where a thin market exists and the prices do not reach an equilibrium. They solved a contingent claim call auction market with a linear programming formulation that maintains the pari-mutuel property of paying the winners with the stakes of the losers.

Prediction Markets Prediction markets are defined as speculative markets whose purpose is to create predictions for the outcome of a particular event. An important event in the growth of prediction markets was the development of the Logarithmic Market Scoring Rule (LMSR) by Hanson (7). We will describe the mechanics of the LMSR in section 3 but one key point is that this mechanism serves as an automated market-maker. The market-maker will calculate and post prices for all possible states. This allows the market organizer to immediately post prices for all states - rather than waiting for traders to post orders as in a continuous double auction. While this mechanism is generally pari-mutuel (except for a controllable amount of seed money that the organizer must provide), it differs in an important manner from the standard pari-mutuel market. The value of a winning order is fixed in Hanson's mechanism.

In 2006, Pennock et al. (12) built upon some previous work to introduce another mechanism called the share-ratio Dynamic Pari-mutuel Market-Maker (DPM). Their mechanism also operates as an automated market-maker with controlled risk to the market organizer. Powered by these research innovations, many new prediction markets have been introduced in recent years. The Yahoo! Tech Buzz Game uses the DPM mechanism to set prices. Hanson's LMSR is being employed by many online prediction markets including the TheWSX and InklingMarkets.

Financial Markets In 2002, Goldman Sachs and Deutsche Bank teamed up to create a market for their clients allowing them to trade claims over the potential values of economic indicators which would be announced in the future (Cass (4)). The technology for these markets is based on a call auction mechanism designed by a firm named Longitude (described in Baron and Lange (1) and Lange and Economides (8)). Since the market is conducted as a call auction, the organizer will collect all orders then close the market and determine which orders to accept and which to reject. The organizer has the advantage of seeing all the orders before needing to make commitments to accept any order. Therefore, the mechanism of Longitude is formulated as an optimization problem which the market organizer must solve. Constraints are used to ensure that the mechanism remains pari-mutuel. Fortnow et al. (5) have a linear programming formulation for this contingent claim call auction but it does not generate unique state prices. The mechanism developed by Bossaerts et al. (2) also addresses this auction but needs to be solved in multiple steps and also does not generate unique state prices. However, the call auction mechanism has one key disadvantage: the traders are not sure about the acceptance of their order until after the market is closed. Thus, if their order is rejected, there is no market to resubmit a modified order.

Let's pause to consider one key distinction between the mechanisms that we have discussed above. While these mechanisms are designed to be implemented in one of two different manners, we will show that it is possible to easily change the manner of implementation of these mechanisms. The mechanisms developed by Hanson and Pennock are designed to be implemented as posted price automated market-makers. On the other hand, the mechanism of Longitude must be implemented as a call auction. A fundamental difference in the posted price versus auction implementation is that the onus is placed on the trader to reveal a price for his order in the auction setting. Despite this difference, the posted price mechanisms can be implemented as an auction.

In Section 2, we will introduce a new mechanism which leverages the auction technology but is applied in a dynamic setting where the market organizer will make immediate, binding decisions about orders. A truthfulness property of the optimal bidding strategy in this auction setting will allow us to conduct a direct comparison with the posted price mechanisms. In Section 3, we will present an overview of the mechanisms developed by Hanson and Pennock plus some simulation results where we compare these mechanisms to the SCPM. We conclude with a discussion in Section 4.

2 The Sequential Convex Pari-mutuel Mechanism (SCPM)

The development of the SCPM followed directly from earlier work of the authors (Peters et al. (13)). In this section, we will examine the predecessor of the SCPM and describe the modifications used to create the SCPM. Longitude had previously developed a mechanism named the Pari-mutuel Derivative Call Auction (PDCA). See Lange and Economides (8) for details of the PDCA. The PDCA has a number of desirable characteristics but is formulated as a non–convex program. Thus, a special solver is required to find global solutions. We have developed an alternative convex call auction mechanism which maintains all the positive characteristics of the PDCA. First, let's describe in detail the format of the market before explaining the details of the market-making mechanisms.

The primary motivation of these market-making mechanisms is to help generate liquidity in small or new markets without needing to expose the market or-

Table 1: Notations Used in this Paper

Variable	Name	Description
$a_{i,j}$	State Order	Trader j's order on state i
q_j	Limit Quantity	Trader j's maximum number of orders requested
π_j	Limit Price	Trader j's maximum price for order
p_i	Price	Organizer's price level for state i
x_j	Order Fill	Number of trader j's orders accepted

ganizer to financial risk. As shown by Bossaerts et al. (2), a call auction which receives limit orders can avoid some of the thin market problems suffered by a continuous double auction and help the market achieve substantially better liquidity. This approach allows the organizer to see the full set of orders before determining acceptances while the limit provisions allow the traders to express the bounds of their willingness to pay for claims.

Now, consider a market with one organizer and n traders. There are S states of the world in the future on which the market traders are submitting orders for contingent claims. For each order that is accepted by the organizer and contains the realized future state, the organizer will pay the trader some fixed amount of money w, which, without loss of generality, equals 1 in this paper. One caveat here is that one of the models which we will describe later in this paper, the Dynamic Pari-mutuel Market-Maker, does not pay a fixed amount to winners. The traders will submit orders to the organizer which specify the states which they want contingent claims over, the price at which they are willing to pay for the order, and the number of identical orders that they will buy. After all orders are submitted, the market is closed and the organizer will then decide whether to accept or reject each order. If the order is accepted, the organizer also decides the number of orders to accept and the price per order to be collected from the trader. As the reader might imagine, a wide range of markets from results of sporting events to elections can be organized in this manner. The challenge is to develop a mechanism for accepting and rejecting orders that doesn't expose the organizer to risk.

Throughout the analysis, we will use the notations in Table 1. The traders will supply the values of $a_{i,j}$, q_j and π_j for all i, j, which are denoted by the matrix A and vectors q and π. Thus, these data are considered given for the models. The market organizer will need to determine the decision variables p_i and x_j for all i, j.

2.1 Convex Pari-mutuel Call Auction Mechanism (CPCAM)

In previous work (13), we have developed an alternative formulation of the PDCA which has similar constraints but is also a convex program. The primary constraints are to ensure that the market is self–funding and that the quantities granted to each trader are consistent based on the relationship of their limit price and the calculated state price of the order. Furthermore, it is valuable that the model has a unique optimum. Below is our alternative pari-mutuel formulation, again, with $w = 1$ where w is the value of the fixed payoff:

$$\begin{array}{lll} \text{maximize} & \pi^T x - M + \sum_i \theta_i \log(s_i) & \\ \text{subject to} & \sum_j a_{i,j} x_j + s_i = M & \text{for } 1 \le i \le S \\ & 0 \le x \le q,\ s \ge 0 & \end{array} \quad (1)$$

In this formulation, θ represents a starting order needed to guarantee uniqueness of the state prices in the solution. The starting orders are not decision variables — in effect, the market organizer is seeding the market with this order. In some outcomes, the market organizer could actually lose some of this seed money. The objective function in this formulation has the following interpretation. The term $\pi^T x - M$ is the profit to the market organizer. On the other hand, the term $\sum_i \theta_i \log(s_i) = \sum_i \theta_i \log\left(M - \sum_j a_{i,j} x_j\right)$ can be viewed as a disutility function (or weighted logarithmic penalty function) for the market organizer that ensures she will find an allocation of accepted orders that is pari-mutuel. In our model, prices (p) are derived from the solution to the KKT conditions of the optimization model. They turn out to be the dual variables corresponding to the self-funding constraints. The KKT conditions also include the requirement that $\sum_i p_i = 1$.

2.1.1 Previous Results

It turns out that the CPCAM can be shown to have many valuable characteristics. In particular, our model yields the first fully polynomial–time approximation scheme (FPTAS) to the contingent claim call auction with unique prices problem. In prior work, we have established the following properties (see Peters et al. (13) for proofs):

- The CPCAM is a convex program that can be solved (up to any prescribed accuracy) in polynomial time using standard techniques.
- The market will be self-funding (other than the required starting orders).
- The optimal solution (x^*, p^*) from the CPCAM model would also be optimal if we replaced π_j with $(p^*)^T a_j$ in the objective function. Furthermore, the solution will remain optimal if we replace π_j with any c_j where $(p^*)^T a_j \le c_j \le \pi_j$.

- The state price vector p^* for any optimal solution to the CPCAM model is unique.
- As $\mu \searrow 0$, the solutions $p(\mu\theta)$ converge to the unique limit points $p(0) = p^*_\theta$.
- The set of feasible solutions of the PDCA model coincides with the set of optimal solutions of the CPCAM model (1) and they produce the identical state price vector. Furthermore, the PDCA model can be solved as a linear program after obtaining the state price vector p^*.

2.2 Sequential Convex Pari-mutuel Mechanism (SCPM)

While the CPCAM possesses many powerful properties, the fact that it must be implemented in a call auction setting is limiting. Two fundamental drawbacks exist. First, market traders do not know whether their order is accepted until the conclusion of the auction. At this point, the market is closed and there is no longer a central platform with which traders can submit a new order. In many situations, they would prefer to have an immediate decision. Second, there is no ability for market traders to lock-in gains by trading accepted orders while the market is still open. This trading is important as it allows traders to hedge positions if there is a swing in the state prices.

Thus, we would like to utilize some of the powerful properties of the CPCAM but change the format of order placement from a call auction setting. We have developed a simple modification to the mechanism that allows us to run the mechanism each time a new order is received by the market organizer. We call this new mechanism the Sequential Convex Pari-mutuel Mechanism (SCPM). Essentially, we run a slightly modified version of the CPCAM where, after receiving each order, we add a constraint to the model to lock in optimal order fills from the previous solution of the model. Thus, when the first order is received, we just run the CPCAM as normal. Next, when the second order is received, we run the CPCAM but we add the constraint that the first order must be filled to the level dictated when we solved the model for only the first order. This process continues with new constraints added after each new order is processed.

Thus, the market organizer can immediately tell the trader whether or not his order was accepted. When the nth order is received, we can formulate the SCPM as follows:

$$\begin{array}{lll} \text{maximize} & \pi^T x - M + \sum_i \theta_i \log(s_i) & \\ \text{subject to} & \sum_j a_{i,j} x_j + s_i = M & \text{for } 1 \le i \le S \\ & x_j = l_j & \text{for } 1 \le j \le n-1 \\ & 0 \le x_n \le q_n & \\ & s \ge 0 & \end{array} \quad (2)$$

This model will determine the amount of the nth order that should be filled. Here, l_j represents the order quantities found from previous solutions of the model. The SCPM will produce state prices which can be used to charge the nth trader for his order.

2.2.1 Properties of the SCPM

Fortunately, the SCPM will preserve many of the positive properties of the CPCAM. First, the market organizer's risk will be bounded by the the starting orders. The maximum possible loss by the market organizer will never exceed $\left(\max_G \sum_{i \in G} \theta_i\right)$ where G represents a set of $S-1$ states. It may be necessary to increase the magnitudes of each θ_i in comparison to the call auction setting since the dynamic setting will be more reliant on the seed money until a sufficient number of orders have been accepted.

In the CPCAM, the state prices would satisfy price consistency constraints for all orders whereby an order will only be accepted if its limit price is greater than or equal to the sum of p_i for all states included in the order. In the SCPM, we will find state prices but they will only satisfy price consistency for the current order. Thus, the prices are less useful in terms of providing information about all orders received previously. However, the current trader can be sure that his order will be accepted or rejected according to price consistency constraints based on the current price. The SCPM will also provide traders a certain payoff amount if one of their states is realized. The market organizer will pay each winning order a fixed amount w. Furthermore, the SCPM is easy to implement. As each order is received, we will be solving a slightly larger convex problem.

Truthfulness

In the call auction context, it is difficult to determine the optimal bidding strategy for the traders when the CPCAM is employed. However, the situation is different when we move to a sequential setting where the SCPM is implemented.

Theorem 1 *When the SCPM is operated in the purely pari-mutuel manner (in* *which traders are charged the calculated state price of their accepted orders), the optimal bidding strategy for traders in the SCPM is to bid their true valuations.*

Proof Let's assume that trader j arrives at the market and seeks to submit an order. The states covered by this order can be represented by the vector a. We will assume that trader j seeks one order (so, his limit quantity is one). Finally, the trader j has a private valuation for his order of v_j. If the order is accepted, the trader will earn some surplus ($s = v_j - p$) where p is the charged price of the order. We will assume that the trader has a utility function, $u(s)$, which is monotone non-decreasing in surplus.

If the trader decides not to bid his true valuation, let's inspect his possible alternative bidding strategies and check for optimality. There are two cases that we must evaluate: bidding less than his valuation and bidding more than his valuation. Remaining consistent with previous notation, we will use $\pi_j = v_j$ to represent the limit price of his truthful bid. Whenever the SCPM is solved for the new order, the model will calculate new state prices. From these prices p, we can obtain a calculated state price $p^T a$. This is the price that the trader will be charged if his order is accepted. Now, let's examine the two cases of bidding strategies.

Case 1 (Bidding less than his valuation): Let π'_j be the new limit price for the trader. We know that $\pi'_j < v_j$. When a truthful bid would be rejected since it is lower than the calculated order price, then the bid of π'_j will obviously also be rejected since $\pi'_j < \pi_j$. This outcome is equivalent in terms of surplus to the outcome in the truthful bidding strategy. Next, when the truthful bid would have been partially accepted when $p^T a = v_j$, the bid of π'_j would be rejected since $\pi'_j < v_j$. Finally, in the case where the truthful bid was fully accepted, we have two outcomes. The bid of π'_j could also be fully accepted. In that case, the trader will be charged $p^T a$ for his order and will have some positive surplus for the accepted order since $(v_j - p^T a)$ is positive. This is an identical outcome to the truthful bid case. However, there is a chance that $\pi'_j < p^T a < v_j$. In this case, the order will be rejected and the trader will receive no positive surplus. If he had bid truthfully, he would have had his order fully accepted and received positive surplus.

Thus, we can see that bidding truthfully weakly dominates bidding less than v_j.

Case 2 (Bidding more than his valuation): Next assume that the bid is π''_j where $\pi''_j > v_j$. When a truthful bid would be rejected since it is lower than the calculated order price, then we have two possibilities. The bid of π''_j could also be rejected if $\pi''_j < p^T a$. However, the bid could be accepted (partially or fully) if $\pi''_j \geq p^T a$. This is a bad outcome for the trader because there is a chance that $p^T a > v_j$. In this case, he actually receives negative surplus since the net value of the accepted order will be $z(v_j - p^T a)$ where z is the order fill amount. This quantity will be negative. If he had bid truthfully, the order would have been rejected. If $v_j > p^T a$, then he will earn positive surplus of $(v_j - p^T a)$ but he would have received this if he had bid truthfully. Thus, the trader can only do worse by bidding more than his valuation in this instance. In the case where the truthful bid is partially accepted, the bid of π_j'' will be fully accepted. However, no positive surplus will be earned since $v_j = p^T a$. This is an equivalent outcome to bidding truthfully. In the case where the truthful bid is fully accepted, the bid of π_j'' will clearly be fully accepted. However, the utility earned will be equivalent since in both cases, the trader will be charged $p^T a$.

So, bidding truthfully also weakly dominates bidding more than v_j. Therefore, we have shown that truthful bidding is the dominant strategy when the purely pari-mutuel version of the SCPM is implemented. ∎

Thus, the mechanism is truthful in a myopic sense since we are only considering the optimal bid for the current order of the trader. It should be noted that it is easy to construct an example where the the optimal bid is not truthful when the market organizer charges the trader his limit price instead of the calculated pari-mutuel price.

Update Mechanism

When our mechanism is implemented, the market organizer must solve a convex program at each time step to determine the new state prices and the fill quantity for the current order. It turns out that the market organizer does not need to solve a convex optimization problem each time a new order is submitted. He can follow an update procedure that involves solving a small system of equations to determine if the order is accepted and to calculate the new prices.

The update procedure relies on insights from the KKT optimality conditions (and the previous solution). From the KKT conditions, we have the following relations:

$$\begin{array}{l} \sum_j a_{ij} x_j + s_i = M \\ s_i = \frac{\theta_i}{p_i} \end{array}$$

Now assume that we have already solved the SCPM for $(n-1)$ orders and the nth order has arrived. Let $\bar{p}$ be the optimal price vector for the previous solution. First, we observe that if $\pi_n \leq \bar{p}^T a_n$ then $x_n = 0$ and $p = \bar{p}$. This follows from the price consistency constraints of the model. If $\pi_n = \bar{p}^T a_n$, then we are assured that $x_n = 0$ since any increase in x_n would increase the prices of the states selected in the order and thus cause $\pi_n < p^T a_n$. If $\pi_n > \bar{p}^T a_n$, the update procedure would examine two cases to determine the new x and p:

1. Since we know that $\pi_n > \bar{p}^T a_n$, we also know that $0 < x_n \leq q_n$ from the price consistency property of the optimal solution. First, we assume that $x_n = q_n$ and see if we can find prices which will satisfy the optimality conditions. So, set $x_n = q_n$ and solve the following system of $S + 1$ equations to find values for M and p:

$$\begin{array}{l} \sum_j a_{ij} x_j + \frac{\theta_i}{p_i} = M \quad \forall i \\ \sum_i p_i = 1 \end{array}$$

   Let's rewrite this system into a form that will be easier to solve. Let $b(i) = \sum_{j=1}^{n-1} a_{ij} x_j$. This represents the number of claims which have been granted for state i. Now, based on our assumption that $x_n = q_n$, we can express $\sum_j a_{ij} x_j$ as $\big(b(i) + a_{(i,n)} q_n\big)$. Next, we can substitute for p_i and combine the equations into the following equation:

$$\sum_i \frac{\theta_i}{M - b(i) - a_{(i,n)} q_n} = 1$$

   Now, we have a function of M only. We can solve this system. If based on the solved value of M and the implied value of p, we have $\pi_n > \bar{p}^T a_n$, then we are done. However, if $\pi_n \leq \bar{p}^T a_n$ based on the solved values of p then this means that our assumption that $x_n = q_n$ was incorrect. Then, we proceed to the second case.

2. Now, we know that $0 < x_n < q_n$. Therefore, it must also be true that $\pi_n = p^T a_n$. Let T be the set of states where $a_{(i,n)} = 0$. We know that $\sum_{i \in T} \frac{\theta_i}{M - b(i)} = 1 - \pi_n$. This is true because $a_{(i,n)} x_n = 0$ for these states and the sum of the prices of these states must be equal to 1 minus the total prices of the states where $a_{(i,n)} \neq 0$. Again, we have a function of one variable (M). We can solve this equation and fix M.

   Next, for the other states, we have the following equation:

$$\sum_{i \notin T} \frac{\theta_i}{M - b(i) - a_{(i,n)} x_n} = \pi_n$$

   Since M is fixed, this is also an equation of one variable (x_n). Thus, we solve this equation and we have the order fill amount x_n. The prices can then be easily calculated.

Unfortunately, these systems of equations are nonlinear. One approach for solving these systems is to attempt to solve a geometric program with a null objective function and the constraint being the equation to be solved. This would basically be a geometric feasibility problem. However, the equation $\sum_{i \notin T} \frac{\theta_i}{M - b(i) - a_{(i,n)} x_n} = 1$ (when either M or x_n is the independent variable) is not a monomial so we cannot leverage solution techniques for standard geometric programs (see Boyd et al. (3)).

Fortunately, there are other efficient means of solving these equations. Simple bisection methods can be used to solve the systems in all instances - bisecting on the value of M or x_n as appropriate. While easy to implement, the bisection method will compute an approximate solution in $O(\log(\frac{1}{\varepsilon}))$ time where ε is the required error threshold. It turns out that there is actually a faster method to solving the update mechanism.

Theorem 2 *An ε-approximation of the optimal solution to the SCPM can be found in $O(\log\log(\frac{1}{\varepsilon}))$ time by using our update mechanism.*

Proof In the first case, we are trying to find a root a function of M.

$$f(M) = \sum_i \frac{\theta_i}{M - b(i) - a_{(i,n)} q_n} - 1$$

From Ye (14), we can see that this function satisfies the conditions of his Theorem 2. In fact, this problem is just a variant of his third example. Thus, Ye's Newton-step based algorithm will solve this particular problem in $O(\log\log(\frac{1}{\varepsilon}))$ time. As an initial point, we can set:

$$M_1 = \max_i \left(\theta_i + b(i) + a_{(i,n)} q_n\right)$$

This will ensure that $f(M_1) > 1$. From this starting point, we can apply Ye's algorithm and find the root.

The second case is very similar. We are now trying to find a root to a function of x_n.

$$f(x_n) = \sum_{i \notin T} \frac{\theta_i}{M - b(i) - a_{(i,n)} x_n} - \pi_n$$

Again, this equation is simply a variant of Ye's third example so we can apply his algorithm to find the root efficiently. ∎

Thus, we have a very efficient method for solving the SCPM for each new order. Rather than needing to solve a convex optimization problem at each iteration, we can simply apply the update mechanism which reduces the problem to simply solving a few equations. We have shown that Ye's algorithm can solve equations of our form very efficiently.

Additional Characteristics

We have discovered a couple other characteristics of the SCPM which are worth describing. We state them in the following theorem.

Theorem 3 *The solutions to the SCPM possess the following characteristics:*

1. *For a given accepted order n, if all values of $a_{(i,n)}$ are equal where $a_{(i,n)} > 0$, then the prices in the states where $a_{(i,n)} > 0$ will strictly increase from the previous solution.*

2. *For a given accepted order n, the state price p_i will be a continuous function of x_n.*

Proof

(Part 1)

Assume for order n that we have a set of states T where $a_{i,n} = a_{j,n} > 0$ if $i, j \in T$. Otherwise, $a_{i,n} = 0$. Assume that our prices before order n were $\bar{p}$ and our new prices will be p. Further, we assume that the optimal order acceptance $x_n > 0$. Our optimal solution will need to satisfy the following primal and dual feasibility conditions:

$$\begin{array}{l} \sum_j a_{i,j} x_j + s_i = M \quad \forall i \\ p_i = \frac{\theta_i}{s_i} \\ \sum_i p_i = 1 \end{array}$$

Now we will examine the possible changes to the prices of state k where $a_{k,n} = 0$ and its implications to the states in T. There are three possibilities. We will determine which possible price change will occur and what the impact will be on the prices of states in T.

1. Price in state k increases – By the second condition, this requires that s_i decreases. By the first condition, we know that M will also decrease. If M decreases then this would force all prices to increase in states where $a_{i,n} = 0$ since $a_{i,n} x_n = 0$ for these states. Now, let's look at the prices for states in T. If M decreases, then s_i where $i \in T$ must also decrease since $a_{i,n} x_n > 0$. Thus p_i would increase and we would have $\sum_i p_i > 1$. Therefore, this is not possible.

2. Price in state k remains constant – By the second condition, this requires that s_i to remain constant. By the first condition, we know that M will also

remain constant. If M remained constant then this would force all prices to remain constant in states where $a_{i,n} = 0$ since $a_{i,n}x_n = 0$ for these states. Now, let's look at the prices for states in T. If M remains constant, then s_i where $i \in T$ must decrease since $a_{i,n}x_n > 0$. Thus p_i would increase and we would have $\sum_i p_i > 1$. Therefore, this is not possible.

3. Price in state k decreases – In this case, s_i would increase for all states not in T. Also, M would increase. Since the prices in the states not in T have decreased, there must be an increase in prices in T to maintain the third condition. Let $\bar{s}$ and $\bar{M}$ represent the values of these variables for the previous solution of the SCPM. For each state in T, we have

$$\bar{s_i} = \bar{M} - \sum_{j=1}^{n-1} a_{i,j}x_j$$

$$s_i = M - \sum_{j=1}^{n} a_{i,j}x_j$$

Now, we can express

$$s_i - \bar{s_i} = M - \bar{M} - a_{i,n}x_n$$

Now, we see that the change in the value of s for all states in T is the same since $a_{i,n}x_n$ will be the same for all these states. Thus, since we require that the prices collectively increase in T to satisfy $\sum_i p_i = 1$, we know that all prices in T will increase.

Thus, we know that all states in T will experience an increase in price and all state not in T will experience a price decrease.

(Part 2)
First, assume that we have an order (π_n, a, q_n). For any $x_n \leq q_n$, we can find a value of π_n such that x_n orders will be accepted. We want to show that the prices are a continuous function of x_n. Let's use x_d to represent an arbitrary value for x_n. So assume we have a limit price π_d such that x_d orders are filled. Holding a constant, the state prices will be a function of x_d. Let the price vector for this filled order be $p(x_d)$. Now, assume that we have another order where x_n orders are filled. We want to show that $p_i(x_n) \to p_i(x_d)$ as $x_n \to x_d$ for all i.

We will use the notation from our update mechanism to show this fact. We can express $p_i(x_n)$ as $\frac{\theta_i}{M - b(i) - a_{i,n}x_n}$. Now, we can simply see that:

$$\lim_{x_n \to x_d} \frac{\theta_i}{M - b(i) - a_{i,n}x_n} = \frac{\theta_i}{\hat{M} - b(i) - a_{i,n}x_d}$$

Note that M is also a variable in this equation. As $x_n \to x_d$, it is possible that this value changes. We have reflected that fact by indicating the limit value of M as $\hat{M}$. At this point, the relationship between M and $\hat{M}$ is not clear. However, if we enforce the constraint that $\sum_i p_i(x_d) = 1$, we can see that $\hat{M}$ must be the optimal M for the problem when x_d orders are accepted. Thus, $\lim_{x_n \to x_d} \frac{\theta_i}{M - b(i) - a_{i,n} x_n} = p_i(x_d)$.

Thus, $p_i(x_n) \to p_i(x_d)$ as $x_n \to x_d$ for all i. Thus the prices are a continuous function in the number of orders accepted.

∎

It is interesting to note that the prices on all states where $a_{i,n} > 0$ for an accepted order n will not necessarily increase when the values for $a_{i,n}$ are not equal for all states where $a_{i,n} > 0$.

3 Performance Comparison

As we mentioned earlier, the SCPM can actually be designed to operate as a posted price market-maker. This involves some simple additional solves of the model by the market organizer to calculate the required prices for orders to be accepted. This flexibility allows us to apply the mechanism to any market which is operated by a market-maker who posts prices. In this section, we will compare the performance of the SCPM against two of the more interesting pari-mutuel posted price mechanisms: the Logarithmic Market Scoring Rule (LMSR) of Hanson (7) and the Dynamic Pari-mutuel Market-Maker (DPM) of Pennock et al. (12). Two key properties that these mechanisms share are: 1) they allow the risk to the market organizer to be bounded and 2) they are based on pricing functions which allow the market to be priced up immediately. Both of these mechanisms are exceptionally easy to implement and are currently being used to operate various online prediction markets. We believe that our work is the first performance comparison testing that has been conducted amongst these mechanisms. Our initial results indicate that the optimal choice of mechanism depends on the objectives of the market organizer. Before we discuss results, we will quickly describe how these pari-mutuel mechanisms operate.

Logarithmic Market Scoring Rule (Hanson)

Hanson uses a market built on market scoring rules which avoid both thin and thick market problems. In particular, Hanson introduces a logarithmic market scoring rule in his work and we will focus on this rule. His market is organized as a market of contingent claims where claims are of the form "Pays \$ 1 if the state is i". Pennock (11) has computed cost and pricing functions for Hanson's logarithmic market scoring rule mechanism. These cost and pricing functions allow the mechanism to be implemented as an automated market-maker where orders will be accepted or rejected based on these functions. Let's assume that there are S states over which contingent claims are traded. We will use the vector $q \in \mathbb{R}^S$ to represent the number of claims on each state that have already been accepted by the market organizer. The total cost of all the orders already accepted is calculated via the cost function $C(q)$. Now, let a new trader arrive and submit an order characterized by the vector $r \in \mathbb{R}^S$ where r_i reflects the number of claims over state i that the trader desires. The market organizer will charge the new trader $C(q+r) - C(q)$ for his order. The pricing function is simply the derivative of the cost function with respect to one of the states. It represents the instantaneous price for an order over one state.

Here are the cost and pricing functions for the Logarithmic Market Scoring Rule.

$$C(q) = b \log \left(\sum_j e^{q_j/b} \right) \quad \text{and} \quad p_i(q) = \frac{e^{q_i/b}}{\sum_j e^{q_j/b}}$$

In this formulation, b is a parameter that must be chosen by the market organizer. It represents the risk that the organizer is willing to accept. The greater the value of b, the more orders the organizer is likely to accept. It turns out that the maximum possible loss to the market organizer is $(b \log S)$.

Share-Ratio DPM (Pennock)

Pennock et al. (12) have introduced a new market mechanism which combines some of the advantages of a traditional pari-mutuel market and a continuous double auction run by a market-maker. In this market, each dollar buys a variable share of the eventual payoff. The share is defined by a pricing function and the amount paid would be the integral of the pricing function over the number of shares purchased. In Pennock (10), the author explores several implementations of this market mechanism. Recently Pennock and his coauthors have introduced a modification where a share-ratio pricing function is used. We will now define P_i to be the current payoff to each holder of a share of state i if i is the eventual outcome. The following two relationships are required in this mechanism: $\frac{p_i}{p_k} = \frac{q_i}{q_k}$ and $P_i = \frac{M}{q_i}$, where M is the total money collected so far. The first ratio forces the states with more accepted shares to be more expensive. The second ratio calculates the pari-mutuel payoff for each state. From these, they derive a cost function which is the total cost of purchasing a vector q of shares. Additionally, p_i will be the instantaneous change in the cost function with respect to q_i. Thus we have:

$$C(q) = \kappa \sqrt{\sum_j q_j^2} \quad \text{and} \quad p_i = \frac{\kappa q_i}{\sqrt{\sum_j q_j^2}}$$

Table 2: Standardizing risk

Mechanism	Maximum Loss Formula	Parameter Values	Maximum Loss
SCPM	$\left(\max_G \sum_{i \in G} \theta_i\right)$	$\theta_i = 1$	2
LMSR	$(b \log S)$	$b \approx 1.82$	2
DPM	$\kappa\sqrt{\sum_j q_j^2}$	$\kappa = 1$ and $q_i^0 = \frac{2}{\sqrt{3}}$	2

Again, the market organizer must determine κ as well as an initial allocation of shares q^0. The market organizer will charge the new trader $C(q+r) - C(q)$ for his order. The maximum loss possible for the market organizer is $\kappa\sqrt{\sum_j (q_j^0)^2}$

These two pari-mutuel mechanisms share many of the same characteristics with the SCPM - the key difference from a trader's standpoint being that the payoffs of the DPM are uncertain (although they are bounded below by κ).

3.1 Sample Data Analysis

In order to compare the three mechanisms, we created some random orders and compared how the mechanisms handled the orders. In this analysis, we generated 10 datasets with 500 orders in each. These orders are constructed in the standard format for orders to the SCPM. For the DPM and LMSR which are based on a posted price mechanism, we will assume that the limit price and states indicated by the order dictate the rate of payoff required by the trader. Thus, we will use the pricing functions from the DPM or LMSR to determine the maximum order fill acceptable to the trader.

There are three possible states for these orders. For each order, the state which is covered by the order is equally likely to be any one of the three states. If the order covers the first or second state, the price limit is a random variable distributed uniformly over the interval [0.2, 0.6]. Otherwise, the price limit will be a random variable distributed uniformly over the interval [0.1, 0.3]. The quantity limit for all orders is 1.

In order to have a more fair comparison of these mechanisms, we would like to standardize the risk that each mechanism places on the market organizer. We have followed the approach of standardizing the worst case outcome for the market organizer of each mechanism. In Table 2, we describe the parameter settings used in our data analysis.

In the SCPM, the risk assumed by the market organizer is controlled by the values of θ_i. These θ_i represent starting orders for the market or seed money inserted by the market organizer. In the calculation of the maximum possible loss, G represents a set of $S-1$ states. In Pennock's share-ratio DPM, the choice of $\kappa = 1$

will guarantee that all orders will have an eventual payoff of at least 1.

3.2 Results

There are some challenges when implementing the DPM so that its results can be appropriately compared to those of the SCPM and LMSR. The primary difference is that the payoff for a winning order in the DPM is not guaranteed to be a fixed amount. In our implementation, we assume that the eventual payoff of an order will be equivalent to its current payoff. This is a slightly aggressive assumption as the lower bound for an order's payoff will be κ which is 1 in our case. By making our slightly more aggressive assumption, there is the possibility that the payoff will be lower than the current payoff. We'll see later that this doesn't appear to be a serious problem for our datasets.

Another issue is that the shares in the DPM are not necessarily equivalent to shares in the LMSR or SCPM since the current payoff for a DPM share will often be greater than one. While a share in the LMSR or SCPM will pay 1 if the correct state is realized, a share in the DPM will pay some amount greater than or equal to 1. Thus, shares in the DPM are actually worth more. Our traders have limit quantities of 1 for all orders, so we want to require that they are only allocated shares whose best case outcome would be a payment of 1. To achieve this goal, we must do two things. First, we must only fill DPM orders so that $xP(x) \leq 1$ where x is the number of DPM shares allocated and $P(x)$ is the current payoff for those shares. Second, for our comparison purposes, we must convert the number of shares allocated in the DPM into a number of shares which would have a payoff of 1 per share. We will use these converted share numbers when we compare the mechanisms below.

For our comparison, we have examined three situations with different profit implications for the market organizer.

1. **Purely pari-mutuel** - All monies collected are redistributed to winners. The market organizer will earn no profit.

2. **Full charge** - Each trader is charged his full limit price. The pricing functions from the DPM and LMSR will still determine if an order is to be accepted. However, after the order fill quantity is determined, the trader will be charged his limit price times the number of shares accepted.

3. **Tax penalty** - Each order is taxed at a certain percent which will guarantee the organizer a profit percentage. In essence we will divide each share that is bought by a trader into two portions. One portion of the share will be retained by the trader. The other portion (equivalent in size to the required profit

Table 3: Performance comparison under different profit settings

Mechanism	Revenue	Orders Accepted	Worst Profit	Profit %
Purely Parimutuel				
SCPM	73.3	201	NA	NA
LMSR	86.8	254	NA	NA
DPM	81.6	240	NA	NA
Full Charge				
SCPM	84.5	201	15.9	18.9%
LMSR	99.3	254	14.0	14.1%
DPM	99.2	240	17.6	17.7%
Tax Penalty				
SCPM	84.5	201	15.9	18.9%
LMSR	69.7	168	13.2	18.9%
DPM	92.7	221	17.4	18.9%

percentage) will be retained by the market organizer. This profit percentage must be specified a priori.

Table 3 displays the performance data from our simulations under these profit settings. From the table, we see that the SCPM actually performs worse than the other mechanisms in the purely parimutuel and full charge settings in terms of revenue collected and orders accepted. However, it does outperform the LMSR in the tax penalty setting (where the profit percentage is set to the profit percentage of the SCPM in the full charge setting). Since they both offer certain payoffs to winners, the LMSR and SCPM are the most comparable mechanisms. Adoption of the DPM requires the market traders to accept uncertain payoffs. Thus, the market organizer will need to determine the importance of certain payoffs and organizer profitability when selecting the most appropriate mechanisms for his market.

It's not clear why the LMSR outperforms the DPM in the purely pari-mutuel setting. One hypothesis for the improved LMSR performance in comparison to the DPM is that the LMSR has a less stable pricing function which may allow it to accept more orders. Later, we will explore why the prices of the DPM demonstrate more stability.

In the full charge setting, we can see that the DPM and LMSR both collect significantly more revenue and orders than the SCPM. The LMSR actually accepts more orders than the DPM while still collecting similar amounts of revenue. Due to the fact that the LMSR accepts so many orders, its gross profit and profit percentage are actually the lowest of the three mechanisms. We believe that the volatility of the LMSR's prices is responsible for its acceptance of lower revenue orders.

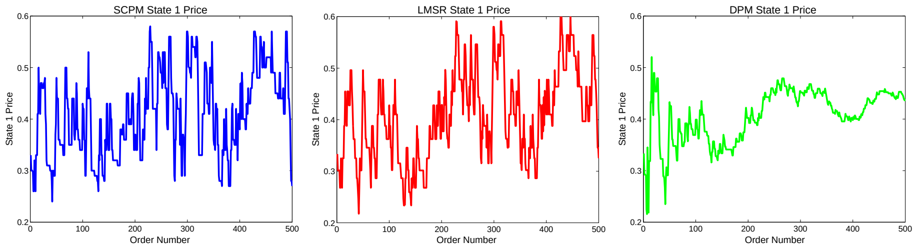

Figure 1: State 1 Price Stability in the SCPM, LMSR and DPM

In the tax penalty setting, the DPM continues to display strong performance. We see that the SCPM outperforms the LMSR in terms of orders accepted and revenue collected. This is not surprising since the SCPM is solving an optimization problem with an objective function that is partially geared towards maximizing worst case profitability. The LMSR is simply accepting orders in a manner consistent with its pricing function. Again, we believe that the rapid change of the LMSR's prices leads to lower levels of order acceptance when the tax is imposed.

We have also tested these mechanisms with random bids generated from a normal distribution. The results are very similar when the variance of the normal distribution is equivalent to the variance of the uniform. However, changing the variance has a substantial impact on the models' performance. The DPM tends to be less effective in the parimutuel and full charge settings when variance is reduced. In general, profits drop quickly as variance is reduced. Due to space constraints, we have omitted more details on this issue but we recognize this as an important consideration for mechanism selection.

Price Stability

One interesting feature of the mechanisms is that they provide updated state prices after each new order is accepted. In the DPM, the prices tend to stabilize to values close to the expected values of the limit prices for each state. However, the LMSR and SCPM both have prices that fluctuate greatly and don't appear to stabilize. Figure 1 shows the evolution of the state prices as orders were received in the first of our sample datasets.

The price of the SCPM is more volatile because it is only required to satisfy the price consistency constraints for the current order. Thus, its value is not tethered to the previous orders. To gain a better estimate of overall demand for a state, the SCPM organizer could choose to solve the CPCAM model for the current orders. The prices in the CPCAM stabilize towards the mean order price. So, while the SCPM prices are not stable, there is a way to gain some further insight into the actual demand for the states. In contrast to the SCPM, the price of the LMSR is calculated by a formula that includes previous orders. To better understand the lack of stability in the LMSR's prices, we need to observe the derivative of its pricing function with respect to a new order being received in one state. From the LMSR pricing function, we have the following:

$$\frac{\partial p_i}{\partial q_i} = \frac{\frac{1}{b}e^{q_i/b}}{\sum_j e^{q_j/b}} - \frac{\frac{1}{b}e^{2q_i/b}}{(\sum_j e^{q_j/b})^2}$$

Now, we can also calculate the derivative of the DPM pricing function. We have:

$$\frac{\partial p_i}{\partial q_i} = \frac{\kappa}{(\sum_j e^{q_j/b})^{\frac{1}{2}}} - \frac{\kappa q_i^2}{(\sum_j e^{q_j/b})^{\frac{3}{2}}}$$

The value of the DPM price derivative will approach zero as the value of q, holding all q_j's in the same proportion, increases. Thus, as more orders are accepted, the price stabilizes. However, the derivative of the LMSR is actually independent of the number of orders received. It will remain constant if q increases as long as the proportion of the values of the q_j's is the same. This explains why the LMSR prices are so volatile. Given that the LMSR is widely used in prediction markets, it is surprising to find the lack of stability of its prices.

Actually, this form can be called 'complete set' equivalent since the derivative of the price for q and $\bar{q}$ will be equivalent if $q + \beta 1 = \bar{q}$ where $\beta > 0$ and 1 is a vector of ones. Thus, we can remove β complete sets of orders from q and the derivative is equivalent. This is an important idea in this setting since these mechanisms are designed to have accepted a similar number of orders for each state at any given time. This allows them to maintain a risk-controlled position.

It turns out the SCPM also shares this 'complete set' equivalence property.

Theorem 4 *Let $\beta \in \mathbb{R}^+$ and $\beta \leq \min_i k_i$ where k_i represents the number of accepted orders for state i already accepted by the SCPM. If (x_n, p) is the optimal price and order fill for the SCPM then (x_n, p) will also be optimal for the SCPM where we remove β complete sets from the accepted orders.*

Proof First, let's list the KKT optimality conditions for the SCPM:

$$
\begin{aligned}
\pi_n - \textstyle\sum_i p_i a_{i,n} + \gamma_n &\leq 0 \\
x_n \left(\pi_n - \textstyle\sum_i p_i a_{i,n} + \gamma_n\right) &= 0 \\
\textstyle\sum_i p_i &= 1 \\
\tfrac{\theta_i}{s_i} - p_i &\leq 0 \qquad \text{for } 1 \leq i \leq S \\
s_i \left(\tfrac{\theta_i}{s_i} - p_i\right) &= 0 \qquad \text{for } 1 \leq i \leq S \\
\gamma_n (x_n - q_n) &= 0 \\
k_i + a_{i,n} + s_i &= M \\
\gamma_n &\leq 0
\end{aligned}
\tag{3}
$$

where $k_i = \sum_{j=1}^{n-1} a_{i,j} l_j$.
Now, assume (x_n, p, γ_n, s, M) is a KKT point for this system. By removing β complete sets from the accepted orders, we are essentially setting $\bar{k}_i = k_i - \beta$. Now, let $\bar{M} = M - \beta$.

It is easy to see that $(x_n, p, \gamma_n, s, \bar{M})$ is a KKT point to the system where $\bar{k}_i$ replaces k_i. ∎

Thus, this explains that the SCPM prices should be volatile like those of the LMSR since the 'complete set' equivalence property means that the increase in accepted orders will not impact price volatility behavior.

Improving SCPM Price Stability

While the SCPM prices will be volatile due to the 'complete set' equivalence property, there are some approaches that can help generate more stable prices. The key idea for generating more stable prices comes from inspecting one of the KKT optimality conditions: $p_i = \frac{\theta_i}{s_i}$. Our hypothesis is that increasing the value of θ_i (the starting order) will increase the stability of price i. While there are drawbacks to increasing the magnitude of the starting orders, the value of producing more stable state prices may justify the decision.

First, let's validate that increasing the magnitude of the starting prices will create more stable prices based on a small example. Assume that we have 3 states and 1 order specifying the third state with a price of 0.5. Now, we will solve the SCPM for varying sizes of θ. Assume that $\theta_i = \theta$ for all i. With equal value starting orders on each of the three states, the initial prices will be $\frac{1}{3}$ for each state. The following table shows the order acceptance and final state prices for three values of θ.

θ Value		State 1 Price	State 2 Price	State 3 Price		Quantity Accepted
0.1		0.25	0.25	0.5		0.2
1		0.29	0.29	0.42		1
10		0.33	0.33	0.34		1

We can see from the table that increasing the magnitude of θ does seem to have an impact on creating more stable prices. The larger starting orders allow the market organizer to accept more orders without needing to adjust the prices to prevent over-exposure.

Next, we will compare several methods of actually increasing the magnitude of θ while running the SCPM and demonstrate the impact on price stability over a sample dataset.

We will use a dataset of 2,500 orders where the state chosen and limit price is randomly determined in the same fashion that we have used for the earlier analysis. First, we will implement a regime where the market organizer increase the value of θ from 1 to 20 in a linear manner over the orders. Second, we will use the profits generated by the market organizer to fund the increase in θ. The profit generated is equal to $\pi^T x - \max_i \sum_j a_i j x_j$. In our implementation, we will take 50% of the guaranteed profit and divide it equally among the states as the starting order. In our third implementation, we will follow a similar approach as the second where we will collect 50% of the guaranteed profit and use it to fund the starting orders. In this implementation, however, we will not divide the funds equally to each state. We will proportionally divide the funds based on the current state prices. So, if state 1 has a price of 0.5 while the other states have prices of 0.25, we will set $\theta_1 = 2\theta_2 = 2\theta_3$. We are adopting this approach to see if we converge to stable prices faster by incorporating prior information about the appropriate state prices into our starting orders.

In figure 2, we have the results from each of these three strategies in addition to a baseline strategy of leaving $\theta = 1$ throughout the entire run. As we had previously observed, the state 1 price in the baseline strategy is not stable. The average order price for state 1 is 0.4. However, the prices fluctuates from 0.23 to 0.59 in the baseline example. On the other hand, our first modified strategy (where we linearly increase theta to 20) creates a greater degree of stability. Here, we observe the prices converging to a much tighter band centered at 0.4 as θ increases in value.

Now, we will examine the strategies which involve using 50% of the profits generated to fund the starting orders. In the second strategy where the profits are distributed evenly over the states, we do see a reduction in the variance of the state 1 price. At the conclusion of this market, the value of the starting orders is [6.3,

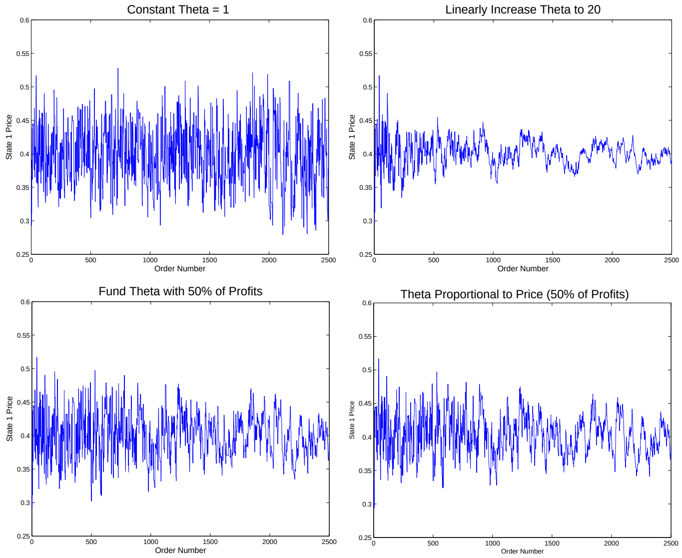

Figure 2: State 1 Price Stability for various starting order policies

6.3, 6.3]. Thus, the value of θ doesn't reach 20 in this simulation so we would not expect the same amount of price stability as our first strategy. Interesting, the third strategy produces state prices almost identical to the second strategy. The final value of the starting orders are [8.9, 9.2, 4.5]. The profits are distributed among the states in a manner which we would expect since the average limit price for orders on state 1 or 2 is 0.4 and 0.2 for orders on state 3. Thus, we would expect the profits to be distributed more heavily on states 1 and 2. We would have expected the third strategy to converge more quickly to a tighter band of state 1 prices around 0.4.

While this is only one dataset, we do feel that this demonstrates a general solution to the issue of generating more stable state prices. The SCPM has the built-in mechanism of the starting orders which can be leveraged as a means to create greater stability in the state prices. We feel that the second and third implementation are very reasonable methods for a market organizer to both run a profitable market and also extract valuable state price information. In many markets (such as prediction markets), the organizer's objective is not to create profit but rather to accept as many orders as possible or create meaningful state price information. For these markets, the concept of "re-investing" the profits into the starting orders makes a lot of sense.

When implementing the third approach, there may be some risk of converging to an unrepresentative state price at some point. We use the term unrepresentative to characteristic a price that is not close to the mean values expressed by traders for the particular state. In the third approach, over time the guaranteed profit could become quite large. When this occurs, the current state price will be used to anchor the value of the starting orders. If the current state price is not representative (due to a recent series of unusual orders or some other situation), the method might end up with a state price close to the current state price for a long duration due to the large magnitude of the starting orders.

4 Discussion

We believe that the SCPM is a valuable and interesting new pari-mutuel mechanism for contingent claim markets. Our analysis indicates that the best choice of mechanism depends on the objectives of the market organizer. In prediction markets, the organizer is typically not interested in generating a profit but instead wants to maximize the number of orders accepted. Our results show that the LMSR will accept more orders and collect more revenue than the SCPM or DPM. In financial markets, the organizer may actually want to guarantee himself a certain profit level or even maximize his profit. The DPM or SCPM would seem to be the better choice in these cases. A second question that the market organizer should consider is how important certain payoffs are to the traders. If the traders exhibit risk averse preferences, then they may significantly discount the value of shares with uncertain payoffs. If the market organizer believes that having a certain payoff for a winning order is important for traders, then he will rule out the use of the DPM. It's important to remember that the posted price versus bid implementations of the mechanisms can be interchanged and are broadly equivalent. While a bid format allows the market organizer to charge the full limit price, the posted price mechanism can be adapted to impose a tax on each share to guarantee the organizer a fixed profit percentage.

One of our earliest concerns about our dynamic mechanism was the amount of time required to determine acceptance decisions for each order and update the state prices. In a standard implementation, the SCPM requires the solution of a convex optimization program as each order is received. While not extremely problematic, this situation was somewhat less desirable than the implementations of the LMSR or the DPM since these mechanisms are based on simpler formulas. Solving a convex optimization problem is not necessary – the market organizer is only required to numerically solve a nonlinear equation. However, as we demonstrated, it turns out that the SCPM can be solved through an easy update mechanism. Since the update mechanism only requires the solution of a few equations which can be solved very efficiently with a Newton step-based approach, the SCPM is very comparable with the other mechanisms from the standpoint of runtime.

One of the most interesting outputs from these market-makers are the state prices. In prediction markets, the generation of these prices is the raison d'être for the market. However, it is very interesting to discover that the LMSR actually doesn't produce very stable prices. The SCPM also suffers from less price stability when compared to the DPM. As one means for generating prices that are more representative of the entire market, one could take all the orders and solve the CPCAM. Since the limit prices on the orders should be truthful, the prices calculated by solving the CPCAM would most likely give a more accurate read of the demand for various states.

While this approach works, it might create a bit of confusion with the traders since the market organizer would be generating two sets of prices. Earlier, we have explored other methods of creating stable prices by increasing the value of the starting orders over time. We believe our technique of funding the starting orders with the 'profits' is a very reasonable way to create more stable prices. It would be valuable to better characterize the relationship between starting order magnitude and price stability in future work.

One assumption that we made when comparing the mechanisms was that traders would assume that the DPM eventual payoffs will be equal to the current payoffs. This might be an aggressive assumption since traders would probably want to discount the payoffs since there is some uncertainty in them. As a quick test, we also implemented the DPM in the conservative setting where the trader assumes that the actual payoff will be 1. However, this mechanism performed very poorly against our sample datasets - hardly any orders were accepted in this conservative setting. It would make sense that most traders would prefer certain payoffs but it is not clear that the traders would assume the worst case payoff when utilizing this mechanism. It would be interesting to see how the DPM performs under various levels of discounting.

References

[1] Baron, K., J. Lange: From Horses to Hedging. Risk Magazine, Volume 16, Number 2 (2003)

[2] Bossaerts, P., L. Fine, J. Ledyard: Inducing Liquidity in Thin Financial Markets Through Combined–Value Trading Mechanisms. European Economic Review, Volume 46, Number 9 (2002) 1671–1695

[3] Boyd, S., S.-J. Kim, A. Hassibi, L. Vanderberghe: A Tutorial on Geometric Programming. Optimization and Engineering, (2005)

[4] Cass, D.: Goldman and Deutsche to Launch Economic Data Options. Risk August 2002 (2002)

[5] Fortnow, L., J. Killian, D. Pennock, M. Wellman: Betting Boolean-Style: A Framework for Trading in Securities Based on Logical Formulas. Proceedings of Electronic Commerce '03 (2003)

[6] Goldman, A. J., A.W. Tucker: Theory of linear programming. In H. W. Kuhn and A. W. Tucker, editors, Linear Inequalities and Related Systems, volume 38 of Annals of Mathematical Studies, (1956) 53–97

[7] Hanson, R.: Combinatorial Information Market Design. Information Systems Frontiers, Volume 5, Number 1 (2003) 107–119

[8] Lange, J. and N. Economides: A Parimutuel Market Microstructure for Contingent Claims. European Financial Management, Volume 11, Number 1 (2005) 25–49

[9] Morrison, D.: Deutsche Bank adopts auction technology to trade economic statistics. Waters Magazine May 2002 (2002)

[10] Pennock, D.: A Dynamic pari-mutuel market for hedging, wagering, and information aggregation Proceedings of Electronic Commerce '04 (2004)

[11] Pennock, D.: Implementing Hanson's Market Maker http://blog.oddhead.com/2006/10/30/implementing-hansons-market-maker/ (2006)

[12] Pennock, D., Y. Chen, M. Dooley: Dynamic Pari-mutuel Market Working Paper (2006)

[13] Peters, M., A. So, Y. Ye A Convex Parimutuel Formulation for Contingent Claim Markets http://www.stanford.edu/ yyye/cpcam-ec.pdf, Working Paper (2005)

[14] Ye, Y.: Combining binary search and Newton's method to compute real roots for a class of real functions, Journal of Complexity, Volume 10 (1994) 271–280

[15] Ye, Y.: A Path to the Arrow–Debreu Competitive Market Equilibrium Mathematical Programming, Online, December (2006)