Reducing Price Fluctuation in Continuous Double Auctions through Pricing Policy and Shout Improvement

Jinzhong Niu
CUNY Guttman Community College

Kai Cai
CUNY Graduate Center

Simon Parsons
Brooklyn College

Elizabeth Sklar
Brooklyn College

How does access to this work benefit you? Let us know!

Follow this and additional works at: http://academicworks.cuny.edu/nc_pubs

Part of the Computer Sciences Commons

Recommended Citation
Reducing Price Fluctuation in Continuous Double Auctions through Pricing Policy and Shout Improvement

Jinzhong Niu and Kai Cai
Dept of Computer Science
Graduate Center
City University of New York
365, 5th Avenue
New York, NY 10016, USA
{niu,kcai}@gc.cuny.edu

Simon Parsons and Elizabeth Sklar
Dept of Computer and
Information Science
Brooklyn College
2900 Bedford Avenue
Brooklyn, NY 11210, USA
{parsons,sklar}@sci.brooklyn.cuny.edu

ABSTRACT
Auction mechanism design is a subfield of game theory dedicated to manipulating the rules of an auction so as to achieve specific goals. The complexity of the dynamics of auctions, especially continuous double auctions, makes it difficult to apply the usual analytic game-theoretic methods to do this. This paper takes an experimental approach, searching a parameterized space of possible auction types, and presents a new pricing policy for continuous double auctions. The paper further demonstrates how this policy, together with a shout improvement rule, helps to reduce the fluctuation of transaction prices in auctions involving agents with minimum intelligence while keeping the overall efficiency high.

Categories and Subject Descriptors
I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—Multiagent systems

General Terms
Algorithms, design, economics, experimentation, measurement, performance

Keywords
Continuous double auction, pricing policy, shout improvement rule

1. INTRODUCTION
Auctions, as a subclass of markets, have been widely used in solving real-world resource allocation problems, and in structuring stock or futures exchanges. This is due to the fact that auctions, when well designed [9], achieve desired economic outcomes like high allocative efficiency. Auctions are also an excellent scenario in which to study the dynamics and characteristics of micro-economics and they play an important role in computer-based control systems [2].

Research on auctions originally interested economists and mathematicians. They view auctions as games and have successfully applied traditional analytic methods from game theory to some kinds of auctions [11], for example the second-price sealed-bid auctions or Vickrey auctions [23]. The high complexity of the dynamics of some other auction types, especially double-sided auctions [5] or DAs¹, however makes it difficult to go further in this direction [10, 20, 25].

As a result, researchers turned to experimental approaches. Starting in 1955, Smith pioneered the experimental approach [21] conducting auctions involving human traders that revealed many of the properties of double auctions. For example, his work showed that in continuous double auctions or CDAs², even a handful of traders can lead to high overall efficiency, and transaction prices can quickly converge to the theoretical equilibrium. Smith’s focus was mainly on the convergence of transaction prices in different scenarios rather than examining why high efficiency is obtained—this is the same topic that interests us. In a computerized world, a question that arises naturally is whether these results can be replicated in electronic auctions. In Smith’s experiments, as in real markets traditionally, the traders are human beings, but computer programs are supposed to be automatic and work without human involvement. Obviously humans are intelligent creatures, but programs are not, at least for the foreseeable future. Is it intelligence that contributes to the high efficiency, or something else?

Gode and Sunder [7, 8] were among the first to address this question, claiming that no intelligence is necessary for the goal of achieving high efficiency; so the outcome is due to the auction mechanism itself. They introduced two trading strategies: zero intelligence without constraint (zi-u) and zero intelligence with constraint (zi-c), and showed that zi-u, the more naive version, which shouts an offer at a random price without considering whether it is losing money or not, performs poorly. In contrast, zi-c — which lacks the motivation of maximizing profit just like zi-u, but guarantees no

¹In DAs, both competing sellers and buyers can make offers, in contrast to the most common auction mechanisms, such as the English auction, where only buyers bid.
²A CDA is a continuous DA in which any trader can accept an offer and make a deal any time during the auction period.
loss — generates high efficiency solutions [7] and provides a lower bound on the efficiency of markets [8]. These results were however questioned by Cliff and Bruten [2], who thought Gode and Sunder’s conclusion was not convincing because the scenarios considered were not as comprehensive as in Smith’s experiments, and showed [2] that in different scenarios the zi-c agents performed poorly, especially in terms of convergence to the theoretical equilibrium.

Cliff and Bruten further designed an adaptive trading strategy called zero intelligence plus or ZIP, and showed ZIP worked better than zi-C, generating high efficiency outcomes and converging to the equilibrium price. This led Cliff and Bruten to suggest that ZIP embodied the minimum intelligence required by traders. Subsequent work has led to the development of further trading strategies, including that proposed by Roth and Erev [18], which we call GD.

This work on trading strategies is only one facet of the research on auctions. Gode and Sunder’s results suggest that auction mechanisms play an important role in determining the outcome of an auction, and this is further borne out by the work of Walsh et al. [25] (which also points out that results hinge on both auction design and the mix of trading strategies used). For example, if an auction is strategy-proof, traders need not bother to conceal their private values and in such auctions complex trading agents are not required. While typical double auctions are not strategy-proof, McAfee [12] has derived a form of double auction that is strategy-proof (though this strategy-proofness comes at the cost of lower efficiency).

The work described here differs from the bulk of the literature. It differs because that bulk of work considers the design of trading strategies, looking for approaches that can achieve high efficiency [6, 17] or high profit [19, 22]. In contrast, we follow the work of Phelps et al. in looking to optimize several metrics — price fluctuation and efficiency — by manipulating the design of the auction itself, and we aim to do this, as Walia et al. [24] do, for populations of zi-c agents. In particular, this paper presents some preliminary experimental results in designing auction mechanisms, showing how a new pricing policy and a new shout improvement rule can augment a traditional continuous double auction involving zero intelligence agents to reduce the fluctuation of transaction prices, while keeping overall efficiency high.

This paper is organized as follows. Section 2 discusses recent related work in auction mechanism design. Section 3 introduces our new pricing policy and shout improvement rule, and describes the experimentation work and results. We follow that with a discussion of our results in Section 4 and conclude with a brief summary in Section 5.

2. AUCTION DESIGN

Mechanism design applied to auctions explores how to design the rules that govern auctions to obtain specific goals.

2.1 Auction types

Generally an auction involves buyers and sellers who aim to exchange commodities at agreed prices. The most common auction form is English auction, in which there is a single seller, and multiple buyers compete by making increasing bids for the commodity — the one who offers the highest price wins the right to purchase the good being auctioned. Since only one type of trader makes offers in an English auction, it belongs to the class of single-sided auctions. Accordingly, there are double-sided auctions or DAs, in which both sellers and buyers make shouts. Two most common forms of DAs are clearing house or CH and continuous double auction or CDA. In a CH, an auctioneer collects bids — shouts made by buyers — and asks — shouts made by sellers; the market-clearing (or equilibrium) price is then determined by maximizing the number of transactions and guaranteeing that if a trader is involved in a transaction, those with more competitive offers are also involved; and the market finally clears at that price. In a CDA, a trader can make a shout and accept an offer from someone at any time, which makes a CDA able to process many transactions in a short time. Both kinds of double auction are of practical importance as well, with, for example, CDA variants being widely used in real-world stock or trading markets including the New York Stock Exchange (NYSE) and Chicago Mercantile Exchange.

One can think of different forms of auctions as varying according to certain aspects of the auction rules, forming a parameterized auction space. For example, Wurman et al. [27] parameterized auction rules using the following classification:

- Bidding rules: How many sellers and buyers are there? Are both groups allowed to make shouts? How is a shout expressed? Does a shout have to beat the corresponding market quote if one exists?
- Information revelation: When and what market quotes are generated and announced? Are shouts visible to all traders?
- Clearing policy: When does clearing a market take place? When does a market close? How are crossing shouts matched? How is a transaction price determined?

2.2 Performance metrics

Auctions with different values for the parameters described above may vary greatly in performance. Popular performance measurements include, but are not limited to, allocative efficiency and Smith’s coefficient of convergence g. The allocative efficiency, E_a, of an auction is used to measure how much social welfare is obtained through an auction. The actual overall profit, P_a, of an auction is:

\[ P_a = \sum_{i} [v_i - p_i] \] (1)

where \( p_i \) is the price of a trade made by agent \( i \) and \( v_i \) is the private value of agent \( i \) for all agents who trade. The theoretical or equilibrium profit, \( P_e \), is:

\[ P_e = \sum_{i} [v_i - p_0] \] (2)

for all agents whose private value is no less competitive than the equilibrium price, where \( p_0 \) is the equilibrium price, the price at which, if all agents bid their private value, supply would equal demand. Given these:

\[ E_a = \frac{100P_a}{P_e} \] (3)

Another term for offers made by traders.
By way of indicating typical efficiencies achieved in a CDA, Figure 1 shows the trend of the overall efficiencies of homogeneous CDAs lasting 10 days with 50 rounds per day\(^4\) in which 10 buyers and 10 sellers all use the same strategy, one of: TT\(^5\), Kaplan\(^6\), ZIP, RE, and GD. The results are averaged over 400 iterations and obtained in JASA—an extensible Java-based auction simulation environment [13].

The convergence coefficient, \(\alpha\), was introduced by Smith [21] to measure how far an active auction is away from the equilibrium point. It actually measures the deviation of transaction prices from the equilibrium price:

\[
\alpha = \frac{100}{p_0} \sqrt{\frac{1}{n} \sum_{i=1}^{n} (p_i - p_0)^2}
\]  

\[\text{(4)}\]

### 2.3 Auction design and performance

It is not yet clear how auction design, and thus the choice of parameter values, contributes to observed performance. Thus it is not possible to determine how to create an auction with a particular specification. It is possible to design simple mechanisms in a provably correct manner from a specification, as shown by Conitzer and Sandholm [3, 4]. However it is not obvious that this kind of approach can be extended to mechanisms as complex as double auctions. As a result, it seems that we will have to design double auction mechanisms experimentally, at least for the foreseeable future.

Of course, doing things experimentally does not solve the general problem. A typical experimental approach is to fix all but one parameter, creating a one-dimensional space, and then measure performance across a number of discrete sample points in the space, obtaining a fitness landscape that is expected to show how the factor in question correlates to a certain type of performance and how the auction can be optimized by tweaking the value of that factor [16]. In other words, the experimental approach examines one small part of a mechanism and tries to optimize that part\(^7\). The situation is complicated when more than one factor needs to be taken into consideration — the search space then becomes complex and multiple dimensional, and the computation required to map and search it quickly becomes prohibitive. Instead of manual search, some researchers have used evolutionary computation to automate mechanism design in these situations.

For example, Cliff [1] explored a continuous space of auction mechanisms by varying the probability of the next shout (at any point in time) being made by a seller, denoted by \(Q_s\). The continuum includes the CDA \((Q_s = 0.5)\) and also two purely single-sided mechanisms that are similar to the English auction \((Q_s = 0.0)\) and the Dutch flower auction \((Q_s = 1.0)\). Cliff’s experiments used genetic algorithms and found that a \(Q_s\) that corresponds to a completely new kind of auction led to a better \(\alpha\) value than that obtained for other auctions when all traders used the ZIP strategy. Walla, Byde and Cliff [24] continued with this work, showing that the new “irregular” mechanism can lead to high efficiency in 21-C markets as well. Taking another tack, Phelps et al. [14, 16] demonstrated how genetic programming can be used to find an optimal point in a space of pricing policies, where the notion of optimality is based on efficiency and trader market power, and transaction prices are determined from matching bids and asks and market quotes.

In this paper we make some preliminary steps towards an evolutionary computation analysis that extends the work of [24]. In particular, we demonstrate that a new policy for setting transaction prices and a new shout improvement rule are capable of lowering the fluctuations in transaction price while keeping efficiency high. Both these new mechanisms are parameterized, and so the space of all such mechanisms can be searched using evolutionary techniques.

### 3. EXPERIMENTAL SETUP

We start by looking at the impact of a new pricing policy and then go on to look at a shout improvement rule.

#### 3.1 A new pricing policy

A pricing policy defines how transaction prices are determined from the offers made by traders. In traditional CDAs, a typical pricing policy sets the price in the middle of the

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]

\[\text{prices.}\]
matching ask-bid pair using the following expression with 
\( k = 0.5 \)
\[ p = k \cdot p_a + (1 - k) \cdot p_b \] (5)
where \( p_a \) and \( p_b \) are prices offered respectively by the seller and the buyer. This is the usual \( k\)-double auction pricing rule [20], which is clearly a discriminatory policy, and which we denote as the \texttt{kPricingPolicy} hereafter. In contrast to this, \texttt{cda} usually adopt a non-discriminatory or uniform pricing policy. Phelps et al. [16] have shown that \( k = 0.5 \) is the best value to use when optimizing a combination of allocative efficiency and trader market power.

Determining the transaction price merely by the corresponding bid and ask, however, may be problematic. In a \texttt{cda} with \texttt{zi-c} traders using the \texttt{kPricingPolicy} (denoted as \texttt{kcda-ZiC} hereafter), the transaction prices fluctuate wildly as shown in Figure 3(a). This result, which is due to the randomness with which \texttt{zi-c} makes offers, is seen in Godde and Sunder’s results [7]. In contrast, in markets involving more complex trading strategies like \texttt{gd}, the phenomenon is less obvious. Figure 3(d) shows the same measurements as Figure 3(a) but for all-\texttt{gd} traders (a setup called \texttt{kcda-gd} hereafter).

The above results are obtained by running \texttt{kcda-ZiC} and \texttt{kcda-gd} in \texttt{JASA} with similar settings to the auctions that gave rise to Figure 1. Figure 3(a) and Figure 3(d) show the first 10 iterations, each with a different random seed, detailing the dynamic characteristics of the transaction prices set in these auctions. It is important to note that this is 10 separate runs over 10 days, with each run starting from scratch. If we did not start over from scratch at the end of each 10 days of trading, the prices for \texttt{gd} would fluctuate even less because there would be no “start of run” variation. The supply and demand schedules are fixed for all the iterations and those in Figure 2.

There are several reasons why we consider it worth studying the fluctuation in transaction prices in markets populated by \texttt{zi-c} traders. First, as Walia et al. [24] argue, auction mechanisms that do well with \texttt{zi-c} are likely to be “robust to trader-irrationality”, in other words they are markets that will work reliably well. Second, markets in which one can trade with minimal intelligence and still have good performance are nearly as good as markets that are strategy-proof — traders are free from some of the burden of having to make good offers. Finally, reducing fluctuation is good for participants in the market — participants in a low fluctuation market are guaranteed not to be given a transaction price too far from the equilibrium, and this may mean traders are more likely to use this kind of market than another with higher fluctuations.

To reduce the fluctuation, we used a new pricing policy that we call the \texttt{nPricingPolicy}. This keeps a sliding window of matching ask-bid pairs and uses the average price of those shouts to set the transaction price. If \( n \) is the window length and \( t \) the number of matching ask-bid pairs at a given point in time, then the transaction price for the \( t \)th transaction is given by:
\[ p_t = \frac{1}{2n} \sum_{i=(t-n+1)}^{t} (p_{a,i} + p_{b,i}) \] (6)
where \( p_{a,i} \) and \( p_{b,i} \) are the ask and bid prices corresponding to the \( i \)th transaction respectively. The \texttt{nPricingPolicy} generalizes the usual \( k\)-double auction pricing policy (\texttt{kPricingPolicy}) since when \( n = 1 \), it becomes the \texttt{kPricingPolicy} with \( k = 0.5 \), and when \( n \) is the number of matching ask-bid pairs in the entire auction, the \texttt{nPricingPolicy} is the rule commonly used to set prices in a \texttt{ch}.

We replaced the \texttt{kPricingPolicy} with the \texttt{nPricingPolicy}, \( n = 4 \), in the same auctions as before, giving \texttt{ncda-ZiC} and \texttt{ncda-gd} respectively. Figures 3(b) shows that the price fluctuation is reduced for \texttt{zi-c} markets, but is still higher than that of \texttt{kcda-gd}.

11 This qualitative conclusion is supported by the quantitative data provided by measuring \( \alpha \) for the different markets. The means and standard deviations of \( \alpha \) and \( E_a \) in \texttt{cda-ZiC}s are plotted in Figures 4(a)–4(b), and Table 1 gives the numeric values of \( \alpha \) and \( E_a \) over the full 10 day run of these markets. Averages and standard deviations are obtained from 400 iterations of each 10 day run.

The values from Table 1 tell us \( \alpha = 12.2 \) for the \texttt{ncda-ZiC}. This value plus twice its standard deviation (1.5) is less than \( \alpha = 15.6 \) for the \texttt{kcda-ZiC}. Since the \( 95\% \) confidence interval is two standard deviations around the mean, we can conclude that the \texttt{nPricingPolicy} significantly reduces \( \alpha \) in \texttt{zi-c} markets. By the same calculation, the improvement of \( \alpha \) in the \texttt{ncda-gd} is not significant (it is less than one standard deviation).

We note that the \texttt{nPricingPolicy} merely sets transaction prices differently, which affects the distribution of profit across the traders but not the overall efficiency. This is confirmed by the fact that the \( E_{a,8} \) of the \texttt{ncda-ZiC} and \texttt{ncda-gd} are identical to their \texttt{kcda} counterparts as shown in Figure 4(b), where the values are superimposed on each other, and Table 1.

Overall, then, we can conclude for now that the \texttt{ncda} is significantly better than the traditional \texttt{cda} in terms of \( \alpha \) when populated by \texttt{zi-c} agents and no worse in terms of efficiency. The results for \texttt{gd} agents suggest that \texttt{ncda} is robust against different types of trading strategies. To confirm this we need to assess its performance with other trading strategies.

### 3.2 Shout improvement rule

Though the \texttt{nPricingPolicy} does well in reducing fluctuation of transaction prices in the \texttt{kcda-ZiC}s, the difference between \( \alpha = 12.2 \) for the \texttt{ncda-ZiC} vs. \( \alpha = 4.0 \) for the \texttt{ncda-gd} in Table 1 tells us that there may still be space to improve the performance of markets with \texttt{zi-c} traders. Why do we still get the obvious fluctuation of transaction prices in Figure 3(b) even when we use the \texttt{nPricingPolicy}? The answer is that the random selection of offers by \texttt{zi-c} causes some variation even though the \texttt{nPricingPolicy} does its best to smooth out prices. This variation is less than that

\[ ^{8} \text{Meaning it varies from trade to trade.} \]
\[ ^{9} [16] \text{ does not just search through possible values of } k. \text{ Instead it searches through all possible arithmetic combinations of shout prices and market quotes, evolving a horribly complex expression which is virtually indistinguishable from the } k = 0.5 \text{ version of (5).} \]
\[ ^{10} \text{Thus } n \text{ makes a continuum of auction spaces with } \texttt{cda} \text{ and } \texttt{ch} \text{ at either end, and exploring other points in this space is a subject of our current work.} \]
\[ ^{11} \text{The result for } \texttt{ncda-gd} \text{ is similar to that for } \texttt{kcda-gd} \text{ in Figure 3(d), so omitted here.} \]
Table 1: Metrics for kCDAs, nCDAs, and nCDAEEs measured over all 10 days of the auctions. Bold face indicates the corresponding market outperforms or equals its traditional kCDA counterpart. **Bold italic** points out the best result in the corresponding ZI-C or GD market group.

<table>
<thead>
<tr>
<th>Auction Type</th>
<th>α</th>
<th>mean</th>
<th>stdev</th>
<th>$E_α$</th>
<th>mean</th>
<th>stdev</th>
</tr>
</thead>
<tbody>
<tr>
<td>kCDA-ZIC</td>
<td>15.613</td>
<td>1.352</td>
<td>95.732</td>
<td>1.537</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDA-ZIC</td>
<td>12.180</td>
<td>1.532</td>
<td>95.732</td>
<td>1.537</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kCDAEE-ZIC</td>
<td>9.360</td>
<td>1.039</td>
<td>93.296</td>
<td>2.061</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDAEE-ZIC</td>
<td><strong>6.028</strong></td>
<td>1.353</td>
<td>93.788</td>
<td>1.938</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDAEE05-ZIC</td>
<td>6.221</td>
<td>1.300</td>
<td><strong>96.072</strong></td>
<td>1.657</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDAEE10-ZIC</td>
<td>6.879</td>
<td>1.117</td>
<td><strong>96.868</strong></td>
<td>1.187</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDAEE15-ZIC</td>
<td>8.000</td>
<td>1.017</td>
<td><strong>96.864</strong></td>
<td>1.179</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDAEE20-ZIC</td>
<td>9.220</td>
<td>1.002</td>
<td><strong>96.744</strong></td>
<td>1.144</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kCDA-GD</td>
<td>4.198</td>
<td>1.140</td>
<td>99.224</td>
<td>0.548</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDA-GD</td>
<td><strong>4.000</strong></td>
<td>1.121</td>
<td><strong>99.224</strong></td>
<td>0.548</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kCDAEE-GD</td>
<td>5.374</td>
<td>4.909</td>
<td>84.612</td>
<td>26.165</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDAEE-GD</td>
<td>5.271</td>
<td>5.245</td>
<td>83.460</td>
<td>26.783</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDAEE05-GD</td>
<td>4.242</td>
<td>2.779</td>
<td>95.996</td>
<td>13.126</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDAEE10-GD</td>
<td><strong>4.054</strong></td>
<td>1.262</td>
<td>98.768</td>
<td>1.092</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDAEE15-GD</td>
<td>4.016</td>
<td>1.256</td>
<td>99.132</td>
<td>0.684</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nCDAEE20-GD</td>
<td><strong>3.994</strong></td>
<td>1.217</td>
<td><strong>99.224</strong></td>
<td>0.548</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Transaction price curves

when we do not smooth prices, but more than that typically generated by GD.

To see how this happens, consider $(p_{a(t-1)}, p_{b(t-1)})$ and $(p_{a}, p_{b})$ to be two consecutive matching ask-bid pairs. If the agents are GD, the two spreads — $[p_{a(t-1)}, p_{b(t-1)}]$ and $[p_{a}, p_{b}]$ — will likely be close if not overlapping since GD makes shouts that are close to previous shouts. This is not true for ZI-C agents for which $p_{a}$ and $p_{b}$ are selected randomly and independently. Thus for ZI-C agents, $[p_{a}, p_{b}]$ may easily be well away from the previous spread $[p_{a(t-1)}, p_{b(t-1)}]$. If this is the case, the transaction price, $p_t$, determined by (6) may fall outside of the spread and the transaction price is selected to be the closest boundary of $[p_{a}, p_{b}]$ to $p_t$ to make sure neither agents will make a loss.

Whichever of $p_{a}$ and $p_{b}$ is used in this case, it will be some distance from $[p_{a(t-1)}, p_{b(t-1)}]$. Since $p_{t-1}$ has to fall into $[p_{a(t-1)}, p_{b(t-1)}]$ for the same reason, we will get two transaction prices $p_t$ and $p_{t-1}$ that are not near to each other. If the auction mechanism forces ZI-C agents to “adapt” their shouts, then the fluctuation might be avoided.

Our approach is to use a shout improvement rule that forces buyers to bid above a running estimate of the equilibrium price, and sellers to ask below it. Thus any matching ask-bid pair must reside on the respective side of the estimate. When we use the NPricingPolicy, the price that is set will not vary far from the previous transaction price. If the estimate changes smoothly and converges to the equilibrium price, the transaction price curve is expected to be smooth as well. More specifically, the shout improvement rule works in a way that is similar to the NPricingPolicy, maintaining a sliding window over a series of $m$ transaction prices and using the average of the prices within the window as the estimate, $p_{e}$:

$$ p_{e} = \frac{1}{m} \sum_{i=t-m+1}^{t} p_i $$

(7)

where $p_i$ is the transaction price obtained at the time $i$ (de-
terminated by (6) for the \textit{N Pricing Policy}. Now, at the beginning of auctions, the transaction prices may deviate greatly from the equilibrium price, and so the rule may cut off those shouts that are actually on the right side of the equilibrium price. As a result, we introduce an additional parameter, $\delta$, to relax the restriction. All bids above $p_e - \delta$ and all asks below $p_e + \delta$ are permitted.

Note that this shout improvement rule is very different from the common "NYSE rule" that forces subsequent offers to be improvements on one another. That rule applies to offers for the same good, and so the value is "reset" by a match between bid and ask. Our rule, in contrast, maintains continuity between the offers made for successive goods (and so, as discussed later, we eventually need to consider its response to price shocks).

We call the auction mechanism obtained by adding the shout improvement rule \textit{cDA with Estimated Equilibrium} or \textit{CDAEE}. We replace CDA with CDAEE with $n = 4$ and $\delta = 0$ in the 4 experiments described in the previous section and get markets we call \textit{kCDAEE-ZIC}, \textit{nCDAEE-ZIC}, \textit{kCDAEE-GD}, and \textit{nCDAEE-GD}. The results of the 2 zic-c markets are plotted in Figures 4(c)-4(d) and all numeric values are presented in Table 1. Figure 3(c) demonstrates the dynamic transaction price curve in nCDAEE-ZIC.

Clearly our results show average $\alpha$ is lower for nCDAEE-ZIC (6.0) than for nCDAE-ZIC (12.2) and the difference is even more significant than between nCDAE-ZIC and kCDA-E-ZIC (the 95% confidence intervals do not overlap), but average $E_a$ for nCDAEE-ZIC is slightly lower than for nCDAE-ZIC. Worse results are observed in nCDAEE-GD, where average $\alpha$ increases and average $E_a$ is down.

One reason for this may be that GD agents adapt their shouts gradually even if their previous shouts were on the wrong side of the estimated equilibrium price and got rejected. Within the limited amount of time as the auctions last, GD agents may not be able to adapt themselves enough to make transactions, which leads to low efficiency.

This indicates that it may be beneficial to relax the shout improvement rule a little and we do this using different positive values of $\delta = 5, 10, 15, 20$—in nCDAEE markets, getting 4 new auction mechanisms denoted as \textit{nCDAEE-D5}, \textit{nCDAEE-D10}, \textit{nCDAEE-D15}, and \textit{nCDAEE-D20}. We ran these mechanisms using zC-C and GD agents respectively and the results are shown in Figures 4(e)-4(f)\textsuperscript{13}, and Table 1.

The nCDAEE markets with $\delta \geq 5$ seem to avoid the low efficiency in the nCDAEES with $\delta = 0$ and the kCDAEES. Table 1 shows that the nCDAEE-D5, 10, 15, 20-ZIC markets perform better than the nCDAE-ZIC, and the nCDAE-D20-GD performs no worse than the nCDAE-GD in terms of both $\alpha$ and $E_a$. Though only the improvement on $\alpha$ in markets using zC-C traders is close to being statistically significant — obtaining low $\alpha$ in the Z-C markets is our major goal in this paper — we believe that these results show that the shout improvement rule has the potential to augment the improvements demonstrated by the nPricingPolicy.

### 3.3 Evolving DAs using genetic algorithms

The above manual parameter tweaking is obviously not an ideal method for auction design. Our ultimate goal is to automate the process. As the first step in this direction, we conducted a series of experiments by encoding various auction parameters into a binary vector of 24 bits and evolving a population of 100 DA individuals, each associated with a randomly generated vector at the very beginning.

Each vector specifies (1) what policy is used to determine transaction prices, \textit{N Pricing Policy} or \textit{N Pricing Policy}, (2) what rule is used to regulate shout improvement, the NYSE rule, the CDAEE rule using sliding window mechanism as in Section 3.2, or the CDAEE rule using the Widrow-Hoff learning algorithm\textsuperscript{14} \textsuperscript{26} to adapt the equilibrium price estimate, and (3) when to clear the market, or, in particular, a probability $\tau \in [0, 1]$ with which the market clears whenever a new shout arrives. The search space is thus multi-dimensional, and includes points corresponding to the usual CDA ($\tau = 1$), CH ($\tau = 0$), and all the other auction mechanisms discussed above. This enables us to optimize auction mechanisms automatically by searching the space.

We first experimented up to 100 generations with minimizing $\alpha$ as the goal\textsuperscript{15} and all DA individuals were evaluated in the same setting as before using homogeneous ZC traders. The results showed that the CDAEE shout improvement rule with the sliding window mechanism not only dominated the other choices across the whole population over generations but also appeared in the fittest mechanism individual found during evolution. This fittest individual takes 0 for $\delta$, which is consistent with the trend exhibited by the nCDAE2-zic markets in Table 1, where nCDAE-ZIC, or nCDAE-D0-ZIC, has the lowest average $\alpha$ (4.0). We also tested with maximizing $E_a$ as the goal and found that a similar shout improvement rule prevailed with $\delta = 12$. This result is also confirmative since Table 1 shows that nCDAE-D10-zic has the highest average $E_a$, among nCDAE-zic markets.

Besides $\delta$, the evolutionary experiments identified the values of other parameters for the fittest auction mechanisms as well. For example in the case of $E_a$ defined as fitness, $\tau = 0.69$ was found to be optimal. This is unexpected since a CH is more efficient than a CDA with the same setup and intuitively clearing a market less frequently helps avoid less competitive shouts being involved in transactions. We leave this issue and others as our future work.

### 4. Discussion

The main result of this paper, then, is that we have identified a new variation on the double auction, a continuous double auction with a new pricing policy and a new shout improvement rule, in which zero intelligence agents can perform well. As Walia et al. argue, that we get high efficiency and low $\alpha$ even for zC-C agents, agents that are on the border-line between rational and irrational trading behavior, suggests that the new auction is pretty robust.

However, there is more we can do to test the robustness of this auction. For a start, we need to run further iterations of the existing experiments to identify the significance or otherwise of the improvements due to the shout improvement rule on top of the new pricing policy. We also need to examine how the new auction performs on other performance metrics and when using trading strategies other than zC-C and GD, and we are currently running experiments with the other trading strategies in JASA, namely ZIP, Kaplan, RE,\textsuperscript{16}.

\textsuperscript{13}Only ZC-C markets are shown, again to save space, and standard deviations were omitted from these figures to make them cleaner.

\textsuperscript{14}The Widrow-Hoff learning algorithm is used in the ZIP and PVT strategies to calculate the target price towards which the shout price should move.

\textsuperscript{15}The fitness function is $F = 100 - \alpha$.
Figure 4: Values of $\alpha$ and $E_a$ for ZI-C markets, (a)-(c) with standard deviations, the values in (b) are so close that the points are superimposed on one another. 

5. SUMMARY

This paper has described some experiments in mechanism design for auctions. As a first step towards new work on automated mechanism design, we have explored by hand some aspects of the continuous double auction. In particular, we looked at setting the transaction price by averaging over $n$ different matched bids and asks, and incorporating a shout improvement rule that forces offers to beat the last $m$ offers plus or minus some value $\delta$, and showed that these modifications created an auction that can achieve high efficiency and low variation in transaction price even for agents with zero intelligence.

This new auction type is interesting on its own, as a counterpart to the mechanisms discovered by [1] and [24], but more interesting, we believe, is that $n$, $m$, and $\delta$, along with $Q_s$ from [1] provide a parameter space that can be searched automatically for additional new kinds of auction that optimize performance on metrics other than $E_a$ and $\alpha$, and for agents employing other kinds of trading strategies. The final contribution of this paper is to demonstrate that such an automated search can discover interesting new types of auction.

6. ACKNOWLEDGMENTS

This work was partially supported by NSF IIS-0329037. We thank the anonymous reviewers for their valuable suggestions.
7. REFERENCES


