

Digital Cancellation Event Options in Limit Order Markets with Automated Liquidity Self-Provisioning

Safraz Rampersaud and Daniel Grosu
Wayne State University
Department of Computer Science
Detroit, MI. 48202 USA
{safraz, dgrosu}@wayne.edu

Abstract—We propose a new limit order book (LOB) design by embedding a revenue-generating, options-based directive to manage malicious trading activities and increase liquidity in a high frequency trading (HFT) environment. The revenue collected as a result of market participants’ desire to purchase the right to cancel orders, is aggregated into an automated, self-managed LOB bank. The LOB accesses its bank and an accumulated order inventory in times of low liquidity. Low liquidity in this context is characterized by an imbalance of orders in the ask (bid) partition of the LOB relative to the bid (ask) partition. To compensate for this imbalance, the proposed LOB design is enhanced with a new market feature capable of assessing and improving its stability through absorbing selected orders. We perform extensive simulation experiments to characterize the ability of the proposed design to stabilize the LOB.

Keywords—limit order book; high frequency trading; simulation;

I. INTRODUCTION

The complexity of financial markets has reached new levels as a result of leveraging innovations in technology. Policy reforms are needed to manage these innovations particularly the millisecond trade execution environments. Regulators are taking strides to develop policies which constrain malicious trading activity and limit excess volatility [1].

Two recent episodes, the *Knight Trading Glitch* and the *Flash Crash*, demonstrate the need for stronger software configuration management controls [2] and innovative measures to counterbalance market volatility. The details of the *Flash Crash* of May 6th, 2010 have been studied extensively by academic and government institutions [3]. The summary reporting their findings [1] suggests several strategies to address similar market events: implementing stock-by-stock circuit breakers to stop trading activity over a short period of time, enhancing liquidity through make/take fees, information provisioning, reinterpreting market maker obligations, applying taxation, and setting speed limits, as well as limiting sponsored access and co-location strategies. A quote from the Joint Advisory Committee on Emerging Regulatory Issues [1] suggests a new direction in our understanding of current market interactions, “Liquidity in a high-speed world is not a given: market design and

market structure must ensure that liquidity provision arises continuously in a highly fragmented, highly interconnected trading environment.” Recently, the NYSE and NYSE/Arca exchanges have filed proposed changes to “prohibit manipulative or deceptive quotations or transactions” [4]. While these proposals are gaining momentum from regulators, the effects of malicious strategies such as quote-stuffing are still being detected. Quote stuffing [5] is the practice of submitting a large number of buy or sell orders to be canceled almost immediately for the purpose of causing congestion and limiting trading opportunities.

Quote stuffing as a strategy is made thoroughly effective through HFT. Research suggests that HFT makes up approximately 70 percent of the volume of all trading in active markets. Ye et al. [5] suggested that increasing HFT speeds from micro- to nano-seconds may not affect positively the market liquidity, price efficiency, and volume. Rather, the increase of speed supports a significant increase in cancellation orders which correlates to quote-stuffing strategies placed by HFT firms. One of the proposed solutions is the application of a tax or fee on market participants that overwhelm the environment’s bandwidth; correlating the activity to quote-stuffing. While this may impede malicious trading strategies, the tax/fee will reduce liquidity and depth in the market [6]. We believe that solutions to the problems affecting this environment should include either the design of new market mechanisms or embedding new features in the existing market mechanisms.

Our contributions. We propose a Limit Order Book (LOB) market mechanism incorporating new financial instruments designed to improve the stability of the LOB. The directives are new financial instruments resembling Digital options [7]; fitting into a new class of “action-or-nothing” type options categorized similarly to other option classes such as “cash-or-nothing” and “asset-or-nothing”. We call these new proposed instruments *Digital Cancellation Event Options* (DCEOs). They grant the market participants the right to submit cancellation orders but the execution of those orders are time and event dependent upon a liquidity metric characterizing the bid-ask spread within the LOB. The funds

paid to purchase the right to cancel are then accumulated in a bank managed by our LOB and accessed to absorb orders when severe imbalances occur between its ask and bid partitions.

We model our market with no central market authority nor any direct market making participants. Our approach is to create a self-provisioning LOB by automating the liquidity provisioning process. Liquidity provisioning is achieved by empowering the LOB to execute orders under sensitive conditions determined by using our proposed metric, the *density difference indicator*. While the complexities of real market interactions are not fully captured, we believe that our proposed design will have a considerable impact on how future LOB markets are designed.

Related work. Research in HFT and how liquidity is affected is a relevant and current topic. Harding and Ma [8] identified conditions which suggest that HFTs supply liquidity in the US treasury market. Wah and Wellman [9] observed the effects of latency arbitrage and efficiency in a HFT environment. Their proposed two-market model simulates the effects of market fragmentation using agent-based modeling and suggests a change in market design to a discrete-time call market.

Easley et al. [10] developed a new metric, VPIN[©], which has been shown to have accurately predicted the *Flash Crash*. VPIN[©] is a volume-based metric which identifies the order flow toxicity, i.e., when market makers in a HFT environment are providing liquidity at a loss causing their departure which results in the market becoming less liquid. Easley et al. also suggested the use of a VPIN[©] contract by market makers to hedge against toxic order flow. Our model implements similar components but are executed under different market conditions. Our density difference indicator metric is used to assess LOB stability in a shorter time window than VPIN[©]. In addition, the VPIN[©] relies on volume bucketing, i.e., selecting time intervals where there is an equal amount of volume, to assess overall market activity, whereas the DCEO contract is dependent on time intervals with respect to the activity of arriving cancellation orders. Our proposed DCEO contract is used to minimize the momentum of cancellation order executions while generating a revenue stream used to perform liquidity provisioning in the absence of direct market makers. Our design differs from VPIN[©] by enabling the LOB to re-stabilize itself assuming that no other market entity commits to a liquidity enhancing role within the market.

Our proposed DCEOs share some similar features with the Credit Event Binary Options (CEBOs) listed on the Chicago Board of Exchange [11]. The major difference from CEBOs is that in the case of DCEO the transfer from an LOB to the market participant is not cash or an asset, but rather an action.

Related work. The rest of the paper is organized as follows.

In Section II, we introduce the background and notation necessary to describe our LOB market design. In Section III, we present our LOB design and describe the features implemented in the design. In Section IV, we investigate the performance of our LOB design by conducting extensive experiments against a standard LOB considering the same market interactions. Lastly, in Section V, we conclude the paper and present directions for future work.

II. LOB MARKET PRELIMINARIES

Market participants are any entities capable of executing a trade order. Any order o_y executed on behalf of a market participant is defined as follows.

Definition 1: An order $o_y(p, v, t, u, x, z)$ consists of:

- p , the requested price of the order.
- v , the requested volume of the order.
- t , the time of the order.
- u , the proceeds to purchase the “right to cancel”.
- x , the action identifier, **A** for ask, or **B** for bid.
- y , the order type: *L* limit, *M* market, or *C* cancellation order, assigned initially by the market participant.
- z , the market participant ID.

Order arrival intensities have been studied under the assumption that they are Poisson arrival processes [12]. Since we are concentrating on the essential design rather than the market activity, we consider only interval, discrete-time arrivals.

A *limit order book* (LOB) is a structure which maintains a list of unexecuted orders for a unique asset in the market. The unexecuted orders represent the expected price-volume of market participants at time $t + \epsilon$, $\epsilon > 0$, and are referred to as *limit orders*. Limit orders differ from *market orders*, the latter being immediately executed at a price-volume combination at time t . The limit order is indicative of a market participant’s willingness to wait in the order execution process; usually for attaining a better queue position at time $t + \epsilon$ and/or anticipating price changes as a result of future news. LOBs are composed of one bid partition and one ask partition, which are used to match market participant orders. Figure 1 shows a standard LOB and its arrangement of orders by price (vertical axis) and volume (horizontal axis). For a comprehensive review of LOBs we refer the reader to Gould et al. [13].

Our market model is composed of a single LOB $\mathcal{Q}$. Any limit order in $\mathcal{Q}$ will be kept in $\mathcal{Q}$ until either of the following two conditions are met: (i) an order with an opposing action is executed, clearing or partially dissolving it as an entry in $\mathcal{Q}$; or (ii) our proposed features absorb the order to alleviate severe imbalances in the bid or ask partition.

We now define a new metric called the *density difference indicator* which characterizes the imbalances in the LOB partitions. In order to do that, we need to define the *density*

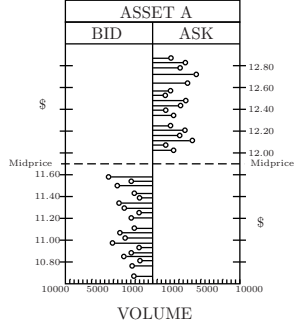


Figure 1. Limit Order Book.

of $\mathcal{Q}$. The density of $\mathcal{Q}$ is the aggregate price-volume product for all orders on both bid and ask partitions.

In an LOB $\mathcal{Q}$, there are up to N bid and ask orders. We denote the price of the order at position i in the bid partition of $\mathcal{Q}$ as P_i^B and the price of the order at position j in the ask partition of $\mathcal{Q}$ as P_j^A . To identify the position of the price on either partition, we must also define the *midpoint price* μ of $\mathcal{Q}$ as follows:

$$\mu = \frac{\max_i \{P_i^B\} + \min_j \{P_j^A\}}{2} \quad (1)$$

Considering the midpoint price, we define the midpoint price position π of $\mathcal{Q}$ as follows:

$$\pi = \left\lfloor \frac{\arg_i \max \{P_i^B\} + \arg_j \min \{P_j^A\}}{2} \right\rfloor \quad (2)$$

On the ask partition of $\mathcal{Q}$, the j^{th} order position is defined on the interval $j \in [1, \pi - 1]$. Likewise, on the bid partition of $\mathcal{Q}$, the i^{th} order position is defined on the interval $i \in [\pi + 1, N]$. The volume of the order at position i on the bid partition is $V_i^B > 0$ and the volume of the order at position j on the ask partition is $V_j^A > 0$.

Inferring information strictly from the density of $\mathcal{Q}$ may not be suitable since LOBs may have equivalent densities with drastically different price-volume combinations. In an attempt to mitigate this issue, we apply exponential smoothing to the density calculation of $\mathcal{Q}$ which we refer to as the *adjusted density*. The smoothing parameter takes the position of the order in the respective partitions of $\mathcal{Q}$ into consideration. As orders arranged by price in $\mathcal{Q}$ drift farther away from the *bid-ask spread*, (i.e., the space between the best bid and ask quotes in $\mathcal{Q}$), the smoothing parameter applies an increasing, continuous weighting effect throughout $\mathcal{Q}$. Orders that are weighted less are more likely to be transacted on by being closer to the bid-ask spread relative to the set of orders in the partition it resides in, i.e., aggressive limit orders. Orders that are weighted more are less likely to be transacted on by being farther away from

the bid-ask spread relative to the set of orders in the partition it resides in, i.e., passive limit orders.

The adjusted ask, respectively bid, partition densities are defined as follows:

$$\delta_A = \sum_{j=1}^{\pi-1} P_j^A \cdot V_j^A \cdot e^{-\frac{(j-1)}{((\pi-1)-1)}} \quad (3)$$

$$\delta_B = \sum_{i=\pi+1}^N P_i^B \cdot V_i^B \cdot e^{-\frac{(N-i)}{(N-(\pi+1))}} \quad (4)$$

The adjusted densities themselves only produce information with respect to each partition. In order to derive information about the partitions relative to the entire activity in $\mathcal{Q}$, we define the *relative adjusted densities* of $\mathcal{Q}$ as follows:

$$\overline{\delta_A} = \frac{\delta_A}{(\delta_A + \delta_B)} \quad (5)$$

$$\overline{\delta_B} = \frac{\delta_B}{(\delta_A + \delta_B)} \quad (6)$$

We define the *density-difference indicator* Δ of $\mathcal{Q}$ as the difference between ask and bid partition relative adjusted densities:

$$\Delta = |\overline{\delta_A} - \overline{\delta_B}| \quad (7)$$

A high value of Δ implies that $\mathcal{Q}$ may have a severe imbalance of orders on one of its partitions relative to the other partition. If these imbalances persist, it informs the market that the asset partially managed by $\mathcal{Q}$ is experiencing a loss of liquidity; making the asset more difficult to trade. At any time, LOB $\mathcal{Q}$ can be in one of two states: (i) *unstable*, suffering from severe imbalances; or (ii) *stable*, when the aggregate orders in the partitions are not severely imbalanced.

Given Δ , we define a boolean signal $\mathcal{S}$ indicating the stability of $\mathcal{Q}$, where $\mathcal{S}$ is 0 if $\Delta > \tau$; and 1 otherwise. Here, τ is the *partition imbalance threshold* i.e., the acceptable upper bound for the density-difference indicator. If $\Delta \leq \tau$, i.e., $\mathcal{S} = 1$, $\mathcal{Q}$ is not in need of liquidity provisioning and is considered stable. If $\Delta > \tau$, then $\mathcal{Q}$, attempting to stabilize the market, executes the proposed liquidity self-provisioning procedure (described in Section III) using proceeds submitted by market participants.

Digital Cancellation Event Options. We propose a class of simple options limiting cancellations for quote-stuffing execution named *Digital Cancellation Event Options* (DCEOs). Excessive cancellations in the HFT space are viewed as manipulative since their transactions by large distort the perception of a specific asset's trade flow and may be used to mislead other market participants. There are circumstances that do require cancellations due to incorrect trade entry but these cancellations are not generated with the momentum observed in malicious trading activities.

The proposed DCEOs are similar to the Credit Event Binary Options (CEBOs) listed on the Chicago Board of Exchange [11]. CEBOs are a European-style binary call option reflecting the probability that a credit event, i.e., bankruptcy, will occur prior to expiration. If the holder of a CEBO is “in-the-money”, then the holder receives cash payment for their position. If at expiry the specified credit event has not occurred, the holder’s CEBO position is nullified and no transfer of funds is executed. These instruments are classified as “cash-or-nothing” options. In our proposed DCEOs, the transfer from Q to the market participant is not cash or an asset, but rather an action. We classify this new class of options as “action-or-nothing” options. Rather than a credit event, Q offers DCEOs to market participants on both partitions under discrete time intervals, where the reference value is based on the best bid or ask price at time of expiry $\hat{T}$. For a market participant to successfully initiate and execute a cancellation order, it must pay for the right to cancel an order inline with how standard options operate.

In current markets, the order types have significant degrees of variation. For instance, the BATS Exchange [14] offers several order types that embed a cancellation concept: (i) Immediate or Cancel (IOC) limit order, if partial orders are not filled, cancel the remaining order; (ii) Good till Canceled (GTC) limit order, cancel the order by 4 p.m. on current trading day; (iii) Cancel and Replace, modify current order parameters. All these order types assume a primary order or condition before executing a cancel order instruction. Our order type allows the order to enter into the LOB without having any previous existing orders or conditions before canceling.

We assume that the market participant is aware of her desire to initiate a quote-stuffing strategy. We allow the market participant to select the order initially as a cancellation order unbeknownst to the other market participants. While Q does have an incentive to increase its liquidity, Q has no preference or bias in accepting the cancellation order o_C ; just as it would with a limit order o_L , or a market o_M . In a real world sequence, the market participant executes orders, then executes orders which cancel those orders. In our design, we reverse the execution sequence whereby the market participant will execute cancellation orders and our mechanism will duplicate and store them as temporary limit orders to be re-assigned or deleted at DCEO expiry.

III. LOB WITH DCEO DESIGN

In this section, we present the proposed design of LOB with DCEO mechanism. The mechanism, called DCEO-Mechanism is presented in an algorithmic form in Algorithm 1. The DCEO-Mechanism is responsible for order management within Q . The mechanism is executed continuously and thus, we include all the statements describing it in an infinite while loop in Algorithm 1.

At the start of the trading day, $t = 0$ (Line 2) and $\hat{T}$ the expiry of the DCEO contract is set to an initial value, e.g., $\hat{T}_0 = 2$ seconds (Line 3). The expiry time of the DCEO is announced to the market by calling the function `Announce` (Line 4). When the current clock time t has surpassed the current DCEO expiry time, the DCEO expiry time is updated (Lines 5-6). The function `Announce` is then called to communicate the change in DCEO expiry time to the market (Line 7).

Next, the orders are organized into: (i) either one of the two sub-LOBs, Q_L or Q_C ; or (ii) enter the main LOB Q . The orders are initially inserted or matched based on their order type set by the market participant sending the order to Q (Lines 8-18). The while-loop continuously accepts orders until $t > \hat{T}$, i.e., when the clock time has surpassed the DCEO contract expiry time. The while-loop allows for orders to be entered into $\hat{Q}_L$, $\hat{Q}_C$, or Q during $[t, \hat{T}]$. The sub-LOBs $\hat{Q}_L$, $\hat{Q}_C$ are temporary copies of Q_L , Q_C which are used to aggregate orders under the current DCEO expiry time. At the time of expiration, the temporary copies $\hat{Q}_L$, $\hat{Q}_C$ transfer orders to Q_L , Q_C . This implementation provides a separation of order processing whereby the temporary sub-books $\hat{Q}_L$, $\hat{Q}_C$ are initialized and ready to accept orders under a new DCEO expiry time immediately after the transfers occur, while concurrently Q_L , Q_C perform order re-assignments on all orders under the previous DCEO expiry time. Without this construction, accepting and re-assigning the correct set of orders would not be possible.

If $y = C$, then this indicates that the newly arrived order is a cancellation order. Prior to o_C being accepted into $\hat{Q}_C$, the market participant submits the cost to purchase the DCEO contract, u , within the order parameters. The amount u is a fixed value per share in our essential model. The DCEO purchase amount u is then added to the current bank balance B managed by Q (Line 11). Following the deposit, a temporary, duplicate order $\hat{o}_L$ of o_C is created (Line 12). Order $\hat{o}_L$ is then inserted into $\hat{Q}_L$ (Line 13) and o_C is inserted into $\hat{Q}_C$ (Line 14). This order duplication allows for order transparency, whereby $\hat{o}_L$ is made public as an order temporarily and the original cancellation order o_C is hidden but is kept for audit trails. If $y = M$, then this indicates that the newly arrived order is a market order to be met immediately at the best available price. The function `Match( $o_M$ )` in Line 16 uses a time price priority matching scheme commonly found in financial markets. Lastly, if $y = L$, the order o_L is stored in $\hat{Q}_L$ as a limit order (Line 18).

After $t > \hat{T}$, the transfer of orders from $\hat{Q}_L$, $\hat{Q}_C$ to Q_L , Q_C allows simultaneous processing of orders under different DCEO expiry times, i.e., orders aggregated under the new DCEO expiry time and orders to be re-assigned under the previous DCEO expiry time. The bid-ask spreads of Q_C and Q_L are determined using the function `Spread` (Lines 21-22). Next, the DCEO-Mechanism considers the

Algorithm 1 DCEO-Mechanism

```

1: while (1) do
2:   if  $t = 0$  then                                ▷ Internal clock time  $t$ .
3:      $\hat{T} = \hat{T}_0$  ▷ Initializes start of day expiry, e.g.  $\hat{T}_0 = 2$  secs.
4:     Announce( $\hat{T}$ ) ▷ Announces DCEO expiry time to market.
5:   if  $t \geq \hat{T}$  then ▷ Updates expiry if current time exceeds expiry.
6:      $\hat{T} = \hat{T} + \hat{T}_0$ 
7:     Announce( $\hat{T}$ ) ▷ Announces DCEO expiry time to market.
8:   while ( $t \leq \hat{T}$ ) do
9:     Receive-order( $o_y(p, v, t, u, x, z)$ )
10:    if  $y = C$  then                                ▷ Cancellation order  $o_C$ .
11:       $B = B + u$                                 ▷ Update bank balance  $B$  with  $u$ 
12:       $\hat{o}_L \leftarrow o_C$                             ▷ Create temporary limit order.
13:       $\hat{Q}_L = \hat{Q}_L \cup \hat{o}_L$ 
14:       $\hat{Q}_C = \hat{Q}_C \cup o_C$ 
15:    else if  $y = M$  then                                ▷ Market order  $o_M$ .
16:      Match( $o_M$ )                                ▷ Orders matched in  $Q$  or  $Q_L$ .
17:    else                                            ▷ Limit order  $o_L$ .
18:       $\hat{Q}_L = \hat{Q}_L \cup o_L$ 
19:    [ $Q_L, Q_C$ ]  $\leftarrow$  [ $\hat{Q}_L, \hat{Q}_C$ ] ▷ Transfers accumulated orders
20:    [ $\hat{Q}_L, \hat{Q}_C$ ] =  $\emptyset$  ▷ Initializes for new DCEO time frame
21:    [ $\text{bid}_{Q_C}, \text{ask}_{Q_C}$ ] = Spread( $Q_C$ )
22:    [ $\text{bid}_{Q_L}, \text{ask}_{Q_L}$ ] = Spread( $Q_L$ )
23:    if  $\text{bid}_{Q_C} < \text{bid}_{Q_L}$  then                                ▷  $Q$  “in-the-money”.
24:      for  $\forall \hat{o}_L \in \{Q_L \mid \hat{o}_L = o_L(\cdot, \cdot, \cdot, \cdot, B, \cdot)\}$  do ▷ Bid.
25:         $o_L(\cdot, \cdot, \cdot, \cdot, B, \cdot) \leftarrow \hat{o}_L$ 
26:    else                                            ▷  $Q$  “out-the-money”.
27:      for  $\forall \hat{o}_L \in \{Q_L \mid \hat{o}_L = o_L(\cdot, \cdot, \cdot, \cdot, B, \cdot)\}$  do ▷ Bid.
28:         $Q_L = Q_L \setminus \hat{o}_L$ 
29:    if  $\text{ask}_{Q_C} > \text{ask}_{Q_L}$  then                                ▷  $Q$  “in-the-money”.
30:      for  $\forall \hat{o}_L \in \{Q_L \mid \hat{o}_L = o_L(\cdot, \cdot, \cdot, \cdot, A, \cdot)\}$  do ▷ Ask.
31:         $o_L(\cdot, \cdot, \cdot, \cdot, A, \cdot) \leftarrow \hat{o}_L$ 
32:    else                                            ▷  $Q$  “out-the-money”.
33:      for  $\forall \hat{o}_L \in \{Q_L \mid \hat{o}_L = o_L(\cdot, \cdot, \cdot, \cdot, A, \cdot)\}$  do ▷ Ask.
34:         $Q_L = Q_L \setminus \hat{o}_L$ 
35:    for  $\forall o_L \in Q_L$  do
36:       $Q = Q \cup o_L$ 
37:    [ $Q_L, Q_C$ ] =  $\emptyset$ 
38:    [ $\delta_A, \delta_B, \Delta$ ] = Calc-Density( $Q$ )
39:    if  $\Delta \leq \tau$  then
40:       $S = 1$                                 ▷  $Q$  boolean signal indicating stability.
41:    else
42:       $\Delta = \text{Self-Provisioning}(\overline{\delta_A}, \overline{\delta_B}, \Delta, Q)$ 
43:      if  $\Delta \leq \tau$  then
44:         $S = 1$                                 ▷ Stable signal following Self-Provisioning.
45:      else
46:         $S = 0$                                 ▷ Unstable signal following Self-Provisioning.
47:    Output  $S$ 

```

bid partition and begins comparing the best bid prices between Q_C and Q_L . If $\text{bid}_{Q_L} > \text{bid}_{Q_C}$, Q will achieve a tighter bid-ask spread as a result of the cancellation orders attracting limit orders within the time interval $[t, \hat{T}]$. As a result, Q has increased its liquidity and is therefore “in-the-money”. When Q is “in-the-money”, all duplicate orders $\hat{o}_L$ under the same DCEO contract are re-assigned to a limit order (Lines 24-25); thereby eliminating the market participants right to cancel. If $\text{bid}_{Q_L} \leq \text{bid}_{Q_C}$, Q is “out-the-money” and every cancellation order under the same DCEO contract are then removed from Q_L (Lines 27-28). A similar procedure is used on the ask partition (Lines 29-34), where

Q is “in-the-money” when $\text{ask}_{Q_C} > \text{ask}_{Q_L}$, and “out-the-money” otherwise. Re-assigning cancellation orders to limit orders when Q is “in-the-money” provides more trading opportunities at prices closer to the bid-ask spread which translates into an increase in the overall liquidity in Q . The lack of trading opportunities at DCEO expiry is a result of being “out-the-money”. In this case, the cancellation orders that have attracted other orders successfully at prices closer to the bid-ask spread, will cancel their orders to meet the better priced orders. As a result, this will widen the bid-ask spread and reduce the amount of trading opportunities in Q .

Following all order post-expiry re-assignment, every $o_L \in Q_L$ is stored and transferred to the main LOB Q (Lines 35-36) waiting to be met. Both Q_L, Q_C are then initialized and prepared for another set of orders to be re-assigned under a new DCEO contract (Line 37). Then the DCEO-Mechanism calls the Calc-Density function which calculates δ_A, δ_B , and Δ using equations (1) through (7).

Once δ_A, δ_B , and Δ are determined by calling Calc-Density and $\Delta > \tau$, the DCEO-Mechanism calls the Self-Provisioning function (given in Algorithm 3). Self-Provisioning is responsible for verifying the stability of Q and absorbing orders when the partitions are severely imbalanced. Self-Provisioning executes a while-loop conditioned on the bank balance B being non-negative and the current density difference indicator being greater than the partition imbalance threshold τ . Self-Provisioning manages order modification in the ask partition (Lines 4-15) and order modification in the bid partition (Lines 16-31). In both partitions, Self-Provisioning randomly selects (using rand, a uniform random number generator) a passive limit order and records its position r . The parameters of rand are the lower and upper order position bounds of the least liquid portion of each partition containing passive limit orders. Given our smoothing effect applied increasingly as the orders drift away from the bid-ask spread, it would be natural to select the absolute least liquid order within either partition to absorb. Using this method, market participants have an incentive to send an unusually high ask priced order or unusually low bid priced order to Q ; which may eventually get absorbed not from other market participants, but from Q when $\Delta > \tau$. By randomizing the selected order to be absorbed within the least liquid quarter set of orders in each partition, we minimize market participant opportunities for “gaming” our design and concentrate on a set of possible passive limit orders rather than a single passive limit order.

The share volume from the selected order is incrementally aggregated until either B does not have enough proceeds to purchase another share or all shares within the selected order have been accounted for (Lines 7-9). The function E-Transfer-Funds transfers the amount $\hat{V} \cdot P_r^A$ from Q to the market participant identified by z in exchange for market shares. A temporary limit order $\hat{o}_L$ containing the aggregate share volume $\hat{V}$ is transferred to an order inventory container

Algorithm 2 Self-Provisioning($\bar{\delta}_A, \bar{\delta}_B, \Delta, \mathcal{Q}$)

```

1: Input:  $\mathcal{Q}$  ask & bid relative-adjusted densities,  $\mathcal{Q}$  density
   difference indicator, LOB: (  $\bar{\delta}_A, \bar{\delta}_B, \Delta, \mathcal{Q}$  )
2: Output:  $\mathcal{Q}$  density difference indicator: (  $\Delta$  )
3:  $\hat{V}, r = 0$  ▷ Volume and random position variables.
4: while ( $\mathcal{B} \geq 0$ ) || ( $\Delta > \tau$ ) do
5:   if  $\bar{\delta}_A > \bar{\delta}_B$  then
6:      $r = \text{rand}(1, \lceil \frac{(\pi-1)}{4} \rceil)$ 
7:     while ( $\mathcal{B} \geq P_r^A$ ) & ( $V_r^A > \hat{V}$ ) do
8:        $\hat{V} = \hat{V} + 1$  ▷ Increment share volume.
9:        $\mathcal{B} = \mathcal{B} - P_r^A$  ▷ Decrement bank by share price.
10:      E-Transfer-Funds(  $\hat{V} \cdot P_r^A, \mathcal{Q}, z$  )
11:       $\hat{\mathcal{O}} = \hat{\mathcal{O}} \cup \hat{o}_L(0, \hat{V}, 0, 0, 0, 0)$  ▷ Transfer shares.
12:      if  $V_r^A - \hat{V} = 0$  then ▷ All shares absorbed.
13:         $\mathcal{Q} = \mathcal{Q} \setminus o_L(P_r^A, V_r^A, \cdot, \cdot, \cdot, \cdot)$ 
14:      else
15:         $o_L(P_r^A, V_r^A, \cdot, \cdot, \cdot, \cdot) \leftarrow o_L(P_r^A, V_r^A - \hat{V}, \cdot, \cdot, \cdot, \cdot)$ 
16:      else
17:         $V = 0$ 
18:         $r = \text{rand}((\pi + 1) + \lfloor \frac{N - (\pi + 1)}{4} \rfloor, N)$ 
19:        for  $\forall \hat{o}_L \in \hat{\mathcal{O}}$  do
20:           $V = V + \hat{V}$ 
21:           $\hat{\mathcal{O}} = \hat{\mathcal{O}} \setminus \hat{o}_L(0, \hat{V}, 0, 0, 0, 0)$ 
22:        while ( $V > 0$ ) & ( $V_r^B > 0$ ) do
23:           $o_L(P_r^B, V_r^B, \cdot, \cdot, \cdot, \cdot) \leftarrow o_L(P_r^B, V_r^B - 1, \cdot, \cdot, \cdot, \cdot)$ 
24:           $V = V - 1$ 
25:           $\hat{V} = \hat{V} + 1$ 
26:        E-Transfer-Funds(  $\hat{V} \cdot P_r^B, z, \mathcal{Q}$  )
27:         $\mathcal{B} = \mathcal{B} + \hat{V} \cdot P_r^B$  ▷ Increment bank.
28:        if  $V_r^B = 0$  then ▷ All shares absorbed.
29:           $\mathcal{Q} = \mathcal{Q} \setminus o_L(P_r^B, V_r^B, \cdot, \cdot, \cdot, \cdot)$ 
30:        if  $V \neq 0$  then
31:           $\hat{\mathcal{O}} = \hat{\mathcal{O}} \cup \hat{o}_L(0, V, 0, 0, 0, 0)$ 
32:         $\hat{V} = 0$ 
33:         $[\bar{\delta}_A, \bar{\delta}_B, \Delta] = \text{Calc-Density}(\mathcal{Q})$ 
34: Output  $\Delta$ 

```

$\hat{\mathcal{O}}$ (Line 11). This sequence simulates the transaction of a requesting ask order to a market participant. Price, time, and other order attributes are not relevant in our transfer as the focus is on the number of available shares to redistribute if needed regardless of the price paid per share. If the entire share volume has been accounted for in the transfer from the randomly selected order to the temporary order, then the randomly selected order is deleted from $\mathcal{Q}$ (Lines 12-13). If only a portion of the volume has been transferred, then the share volume of the randomly selected order is updated (Lines 14-15).

The bid partition undergoes similar modifications. All shares in the order container are aggregated into a volume inventory variable V and then all orders are deleted (Lines 19-21). Self-Provisioning incrementally reduces the share volume from the selected passive limit order, decrements the available inventory, and counts the number of transferred orders (Lines 22-25). The function E-Transfer-Funds transfers the amount $\hat{V} \cdot P_r^B$ from the market participant identified by z to $\mathcal{Q}$ in exchange for inventory shares. Since there is no reduction of $\mathcal{B}$, a one time update on the amount per share by

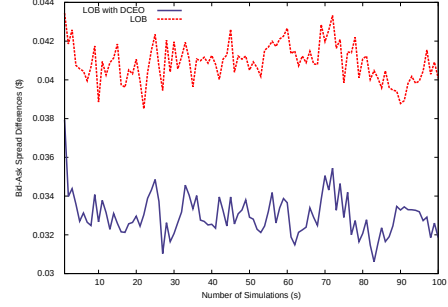


Figure 2. Bid-Ask Spread Differences.

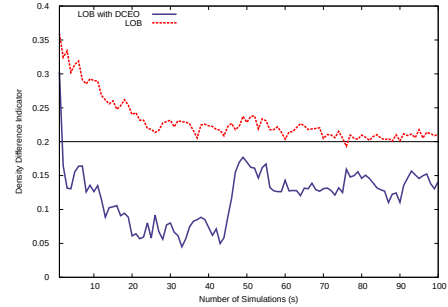


Figure 3. Density Difference Indicator.

volume is executed rather than incremental updates, where checking for a non-negative balance is not necessary (Line 27). The order transitioning (Lines 28-31) follows the same sequence of actions as in the case of the ask partition (Lines 12-15).

IV. EXPERIMENTAL RESULTS

In this section, we perform extensive simulations to compare our proposed LOB with DCEO design against a standard LOB. Both LOBs receive the same set of orders throughout the experiments.

Experimental setup. Our simulations are executed on a 2.4 GHz Intel® Core™ i7-3630 QM CPU 64-bit Windows operating system using 8192 MB RAM. All simulations are coded in C++ and built using a version 4.6.4 GCC compiler. Our trade order data is composed of randomly generated orders with minimal constraints and initially created with single share volume. We impose a fixed maximum price of \$47 and a minimum price of \$37 boundary for both LOBs in our experiment. We classify different order types dependent on price and stage of execution. Limit orders initially fill both LOBs and market orders are generated to match a portion of those limit orders simulating realistic demand around the bid-ask spread. These market orders, once matched to available orders in both LOBs, widen the bid-ask spread and allow for market participants to execute trades with the intent to cancel and attract orders closer to the bid-ask spread. Orders that enter both LOBs following the cancellation orders are considered attracting orders as

their presence is dependent on the opportunity to transact with those orders intend to cancel.

Following this execution, the standard LOB releases those orders identified as cancellation orders regardless of the liquidity contribution from the attracting orders while the LOB with DCEO design compares the bid-ask spread and determines if the liquidity has improved on either partition. Both LOBs then receive another set of market orders simulating more demand. After the cancellation order management, both LOBs calculate their density difference indicator. If the LOB with DCEO design is unstable, the Self-Provisioning function is invoked and performs liquidity provisioning.

Each simulation s is comprised of 250 orders entering both LOBs in the following sequence; 100 initial limit orders, 20 market orders, 100 cancellation orders, 10 attracting orders, and ending with 20 market orders. We plot the average results for both metrics by executing 10 sets of 100 simulations, accounting for the management of 250,000 orders evolving throughout the execution.

Analysis of results. We compare the performance of the LOB with DCEO design and the standard LOB by using the bid-ask spread (Figure 2) and the density difference indicators (Figure 3).

In Figure 2, we plot the average bid-ask spread difference against the number of simulations s . The LOB with DCEO design bid-ask spread difference is represented by the solid line plot while the standard LOB bid-ask spread difference is represented by the dashed line plot. A tighter bid-ask spread indicates more liquid trading opportunities. The LOB with DCEO design shows a tighter bid-ask spread than the standard LOB in every simulation suggesting this market design sustains better liquidity when compared against the standard LOB.

In Figure 3, we plot the average density difference indicator against the number of simulations s . The enhanced LOB with the Self-Provisioning feature is represented by the solid line plot while the standard LOB is represented by the dashed line plot. The partition imbalance threshold is identified by the horizontal solid line and is equal to .20. The data points below this line indicate LOB stability while the data points above it indicate LOB instability. Both LOBs are unstable initially. As the LOBs evolve through the simulations, the LOB with DCEO design keeps the density difference below the threshold by executing Self-Provisioning. We observed simulations where the LOB with DCEO design remains unstable. These are either attributed to not enough proceeds to execute Self-Provisioning and/or irregularity of the incoming orders. In the average case, the LOB with DCEO design remains stable.

V. CONCLUSION

We proposed a new LOB design using a new financial instrument called DCEO. We performed extensive experiments

that show that the proposed LOB design performs better than a standard LOB by offsetting quote-stuffing strategies and empowering the LOB to respond to its own stable state.

While the results showed that our design is an improvement over the standard LOB; the results are dependent on the trade order activity which is randomly generated without considering the market participants' decision making. As future work, we plan to expand our simulation studies to investigate the effects of the trade order activity and market decision making on the performance of the proposed LOB design.

ACKNOWLEDGMENTS

This research has been supported in part by NSF Grant DGE-0654014.

REFERENCES

- [1] Joint Advisory Committee on Emerging Regulatory Issues, "Recommendations Regarding Regulatory Responses to the Market Events of May 6, 2010," Commodity Futures Trading Commission (CFTC) and Securities and Exchange Commission (SEC), Tech. Rep., February 2011.
- [2] B. Aiello, "What Happened at Knight?" 2012. [Online]. Available: <http://cmbestpractices.com/articles/what-happened-at-knight>
- [3] A. A. Kirilenko, A. P. Kyle, M. Samadi, and T. Tuzun, "The flash crash: The impact of high frequency trading on an electronic market," *SSRN Working Paper Series*, Oct. 2011.
- [4] Nanex. (2011, December) Quote stuffing banned. [Online]. Available: nanex.net/Research/QuoteStuffingBanned/QuoteStuffingBanned.html
- [5] M. Ye, C. Yao, and J. Gai, "The externality of high frequency trading," *SSRN Working Paper Series*, August 2012.
- [6] R. Litztenberger, J. Castura, and R. Gorelick, "The impacts of automation and high frequency trading on market quality," *Annual Review of Financial Economics*, vol. 4, no. 1, pp. 59–98, 2012.
- [7] E. Haug, *The Complete Guide to Option Pricing Formulas*, 2nd ed. New York, NY: McGraw-Hill, 2008.
- [8] M. Harding and P. Ma, "The impact of high frequency market makers upon market liquidity: Evidence from exchange outages," Stanford University, Tech. Rep., December 2010.
- [9] E. Wah and M. Wellman, "Latency Arbitrage, Market Fragmentation, and Efficiency: A Two-Market Model," June 2013, Proc. of the 14th ACM Conf. on Electronic Commerce.
- [10] D. Easley, M. M. L. de Prado, and M. O'Hara, "Flow toxicity and liquidity in a high-frequency world," *Review of Financial Studies*, vol. 25, no. 5, pp. 1457–1493, 2012.
- [11] Credit Event Binary Options. [Online]. Available: <http://www.cboe.com/Credit>
- [12] R. Cont, S. Stoikov, and R. Talreja, "A stochastic model for order book dynamics," *Oper. Res.*, vol. 58, no. 3, pp. 549–563, May 2010.
- [13] M. Gould, M. Porter, S. Williams, M. McDonald, D. Fenn, and S. Howison, "Limit order books," *ArXiv*, April 2012. [Online]. Available: <http://arxiv.org/pdf/1012.0349v3.pdf>
- [14] BATS Definition and Order Types: Order and Routing Instruction Descriptions. [Online]. Available: http://batstrading.com/resources/features/bats_exchange_definitions.pdf