High Frequency Fairness*

Guillaume Haeringer
Baruch College

Hayden Melton
Refinitiv

October 20, 2020

Abstract

The emergence of high frequency trading has resulted in 'bursts' of orders arriving at an exchange (nearly) *simultaneously*, yet most electronic financial exchanges implement the continuous limit order book which requires processing of orders *serially*. Contrary to an assumption that appears throughout the economics literature, the technology that performs serialization provides only *constrained* random serial dictatorship (RSD) in the sense that not all priority orderings of agents are possible. We provide necessary and sufficient conditions for fairness under different market conditions on orders for constrained RSD mechanisms. Our results show that exchanges relying on the current serialization technology cannot ensure fairness, including exchanges using 'speed bumps.' We find that specific forms of constrained RSD ensure fairness under certain assumptions about the content of those orders but that the general case nevertheless requires unconstrained RSD. Our results have implications for the design of trading exchanges.

Keywords: Electronic trading, limit order book, fairness, random serial dictatorship.

JEL classification: D71, G100, D47.

*We are grateful to Dhruva Bhaskar, Josh Mollner and Larry Ryan for their extensive feeback, comments, and suggestions. Views expressed herein do not necessarily reflect those of Refinitiv, and do not constitute legal or investment advice.

1 Introduction

In recent years there has been no shortage of controversy about the fairness of electronic financial markets in the presence of high frequency trading (HFT). At the root of the problem are market designs such as the continuous limit order book trading (CLOB) that operate on a first-come, first-served (FCFS) basis.[1] Since FCFS rewards speed, fast traders are advantaged over slower market participants such as long-term investors, and this is sufficient for authors like Lewis (2014) to conclude that electronic financial markets are not fair.

While controversy remains about whether fairness requires that no group of market participants be able to obtain speed advantages over another (see e.g., Angel and McCabe (2013)), what is entirely uncontroversial is that under the CLOB equally fast traders ought to be treated equally. What this means is that when equally fast traders compete for a resource in the CLOB they all ought to have the same probability of being allocated it. The way that this uncontroversial form of fairness has been conveyed in the economics literature is that absent any consideration about the content of traders' orders, the CLOB is akin to a random serial dictatorship (RSD) mechanism, that is, concurrent orders are randomly ordered before being processed. Budish et al. (2015) for instance, having astutely observed that computers operate in discrete time, assume that under the CLOB what happens is "random serial processing of orders that reach the exchange at the exact same discrete time."[2]

While the fairness provided by RSD—owing to the fact it ensures all possible permutations occur with equal probability—makes it a desirable *assumption* for tie-breaking among orders received simultaneously by an exchange implementing a CLOB, the *reality* of tie-breaking on real exchanges is a different story. When an exchange is designed to process orders as quickly as possible (as most are) its implementation of the CLOB is better described as *constrained* RSD because only a small subset of all possible permutations of orders can be generated.[3]

[1] A very short primer on CLOB trading and its terminology is offered in Section 2.1.

[2] Li et al. (2019) too suggest the assumption of RSD: "when multiple HFT order messages (limit orders, market orders, or cancellations) reach the exchange at the same time, they are processed serially in a random order." See also for Baldauf and Mollner (2020) for a similar assumption.

[3] As to the motivation for exchanges seeking to process orders as quickly as possible: Angel et al. (2011), and MacKenzie and Pablo Pardo-Guerra (2014) have noted that historically exchanges were able to increase their market share by processing orders more quickly than their competitors. More recently both Menkveld and Zoican (2017), and Wang (2018) note that exchanges continue to compete on the basis of how fast they can process participants' orders. Further supporting this, improvements exchanges have made in their order

It is the dichotomy between the CLOB, which specifies the processing of orders *serially*, and the large inflows of (nearly) *simultaneous* orders submitted by fast traders that motivates our work. On exchanges that have been designed to be 'fast' (i.e., to minimize the time taken to process participants' orders) it is a general purpose computer networking device known as a *network switch* that is responsible for imposing an ordering on its output port on messages it receives simultaneously on its input ports (i.e., traders' orders).[4] If the exchange is designed to be as fast as possible the priority ordering imposed by the switch is that in which the participants' orders are presented to the CLOB (Melton, 2018; Lohr and Neusüß, 2019). It follows then that the only possible orderings are those the switch is capable of providing, and those orderings are surprisingly constrained. As Sivaraman et al. (2016) have noted, "*today's fastest switches, also known as line-rate switches, provide a small menu of scheduling algorithms: typically, a combination of Deficit Round Robin, strict priority scheduling, and traffic shaping. A network operator can change parameters in these algorithms, but cannot change the core logic in an existing algorithm, or program a new one, without building new switch hardware.*" In practice, this means for a switch that has n ports there are only n distinct possible orderings, and not $n!$ as in RSD. Although switches used by the industry have deterministic algorithms—given one ordering we can unambiguously determine the next—the ordering produced by the switch is independent of the signals that cause traders to submit orders to the exchange (and the content of those orders), so we may reasonably characterize it as random, and ultimately as a *constrained* RSD mechanism.

The first message of this paper is that when an exchange is designed to process orders as quickly as possible its implementation of the CLOB cannot provide fairness. To see this, consider the simple example of an exchange whose switch has only four ports, and there are four traders who are submitting simultaneously an order. Trader i is assigned to port $\sharp i$, $i = 1, \ldots, 4$. In this case the switch can only generate four different queues of those traders, depicted in Table 1. Fairness breaks down because among those four queues trader i_1 is ranked before i_2 three times, while i_2 is above i_1 only once. In other words, whenever traders' orders arrive simultaneously there is a systematic bias towards some traders, i.e., serialization of orders cannot be fair.

processing speeds feature prominently in their recent marketing materials (CME Group, 2020; Eurex, 2016; CBOE, 2020; NYSE, 2020).

[4]On a switch, *ports* the sockets into which network cables are plugged, onto which messages are sent and received.

q_1	q_2	q_3	q_4
i_1	i_2	i_3	i_4
i_2	i_3	i_4	i_1
i_3	i_4	i_1	i_2
i_4	i_1	i_2	i_3

Table 1: Possible queues for a switch with four ports

The main contribution of this paper is to characterize fairness in constrained RSD mechanisms. As a by-product, we show that under current technology when an exchange is designed to process orders as quickly as possible its implementation of the CLOB cannot provide fairness.

Our results are obtained by considering a model built upon the standard setup for object allocation using RSD, which consists of a set of agents and a set of objects where each agent is endowed with a preference relation over the objects. In the RSD mechanism a queue of agents is randomly generated (among all possible orderings), and each agent is asked, one at a time, to pick an object among the objects not chosen by the agents ranked before him in the queue. Our model departs from this standard description in two aspects. First, unlike RSD where all possible queues are equally likely we assume that only queues from a set Q, called a *technology*, can be realized. It is in this sense that our mechanism is a *constrained* RSD. Second, instead of having agents' preferences over objects we assume right away that agents derive a payoff from each realized queues in the allocation mechanism. This modeling approach does not affect our results but it permits us to directly relate fairness to the queuing technology.

Fairness in the social choice or mechanism design literature is often interpreted as a synonym of *equal treatment of equals* regarding agents' *outcomes* (Moulin, 2004). In this paper we follow this principle but consider two different ways to define agents' outcomes. One of the rationales for doing so is that electronic trading can be seen as the concatenation of two mechanisms: a queuing mechanism (the network switch), and a market mechanism (where traders' orders are executed). Here the queuing mechanism is meant to allocate *access* to the market. We thus consider two notions of fairness: *access fairness*, which focuses exclusively on the queuing mechanism, and *outcome fairness*, which applies to the complete queuing+market mechanism.

As its name suggests, access fairness, in line with most current regulations, is based on the desire that among equally fast traders no trader should be advantaged over any other in the allocation of resource he or she receives, with the resource to be understood as participating to the market mechanism. That is, equally fast traders should have the same probability to be ranked first, second, third, ...[5] Outcome fairness on the other hand is the standard notion of equal treatment of equals, which simply requires that two equally fast traders submitting *identical* orders should obtain the same expected *outcomes*. In a classic object allocation problem an agent's outcome is the object he receives, and in a trading context the outcome is the price and quantity at which the trader sold or bought the asset (or whether his order has been successfully canceled).

We show in this paper that *access fairness* and *outcome fairness* differ substantially in terms of their market design implications. A brief summary of our results is the following. We show that a technology Q guarantees access fairness if, and only if, the set of possible queues satisfies a condition we call *strong symmetry* (Proposition 2). This condition is a generalization of the symmetry in found for instance in the Condorcet cycle, where for any rank k in a queue, each trader is ranked k-th as many time as any other trader is also ranked k-th across all possible queues. Strong symmetry goes further by requiring that this symmetry condition also holds when the set of queues is restricted to any subset of traders. A crucial assumption behind this result is that the same queuing mechanism (i.e., the same switch) is used to rank orders for different instruments or assets, a standard feature in electronic markets. In an object allocation problem this translates as having the (constrained) RSD mechanism to generate a unique queue for independent allocation problems (i.e., with different sets of agents and objects).

For our characterization of outcome fairness we do not require our previous assumption that agents trading different instruments are ordered together by the same technology. Our model thus fits the 'standard' object allocation. Also, we highlight a crucial difference

[5] A prominent and recent case of market access (un)fairness appears in a disciplinary order by the Securities and Exchanges Board of India (2019). In this case participants who were otherwise equally fast were not treated as such by an exchange—some were able to obtain 'head-starts' when racing against others by connecting to faster market data distributors, and similarly some were able to 'race shorter tracks' by connecting to faster order gateways at the exchange than others. The review of this and many other cases of market access unfairness provided by Mavroudis and Melton (2019) would seem to indicate that the principle of fair access is ubiquitous across many jurisdictions and asset classes, and that it is of increasing concern to operators of financial exchanges.

between object allocation and trading problems when a running a (constrained) RSD mechanism. In object allocation problems, for any given queue, the outcome of an agent only depends on the preferences of the agents ranked before him. Once an agent is allocated an object the mechanism ends for him and what happens after that has no impact on his allocation. That is not the case in trading, for the success of a trader's order may depend not only on the orders of the traders ranked after him in the queue but also on how those agents are ranked. Accordingly, we characterize outcome fairness for both the object allocation and the trading environments. For the former, outcome fairness is guaranteed if, and only if the queuing technology is *strongly balanced* (Proposition 3), a condition that roughly requires that for any two traders the collection of subqueues of traders ranked above them are identical, up to a permutation of these two traders. This condition is stronger that strong symmetry. For trading problems we show that outcome fairness is guaranteed if, and only if the technology is *fully balanced*, which is a generalization of the strong balancedness condition considering now the entire queues. We later show in the paper that the only strongly balanced technology is the one made by all possible queues, like in unconstrained RSD (Theorem 2), and the same holds for fully balanced technologies. In other words, as long outcome fairness is a concern, RSD cannot be constrained.

Regarding outcome fairness we also investigate the special (and frequent) case when concurrent orders are made only by liquidity providers.[6] This case differ from the general case (characterized with strongly balancedness) in that traders' outcomes depend on which traders are queue before (and thus after) them but not how those traders are ranked. Outcome fairness is guaranteed in this case if, and only if the technology is weakly balanced. This property is similar in spirit to balancedness but only requires that for any two traders the set of traders ranked above them are identical across all possible queues. We show that in fact weak balancedness is equivalent to strong symmetry (Theorem 1).

The rest of the paper is organized as follows. In Section 2 we offer a quick description of the continuous limit order book protocol and an outline of the typical network architecture of electronic financial markets. Our model is introduced in Section 3. In section 4 we explain

[6]This case is a bit unusual and perhaps less interesting for the object allocation literature. It roughly consist of the following situation. The set of agents and objects can both be partitioned into different subsets, $\{N_1, N_2, \ldots, N_k\}$ and $\{X_1, X_2, \ldots, X_k\}$, respectively (with both partitions having the same number of elements). For each h, agents in X_h have the same preferences over the objects in X_h, and all objects in $X_{h'}$, $h' \neq h$ are deemed unacceptable.

how trading fits the allocation model presented in Section 3, discussing the assumptions that traders' order are simultaneous, the switch technology and traders' payoffs (whether liquidity providers or takers). Access fairness is characterized in Section 5 and Outcome fairness in Section 6. We conclude this paper In Section 8 by discussing several alternative market design that aim at improving fairness with CLOB trading protocols.

2 Electronic trading

2.1 Continuous limit order book

We offer here a quick overview of the continuous time limit order book trading protocol. Readers familiar with it and the accompanying vocabulary can skip this section.

The CLOB protocol is simply a double auction that runs in continuous time. Buyers and sellers submit buying and selling orders, respectively, which are simple demand and supply functions consisting of a limit price and a maximum quantity. Prices submitted by buyers and sellers are traditionally called **bids** and **asks**, respectively. That is, a bid is the maximum price the buyer is willing to pay, and similarly an ask is the lowest price a seller will accept to sell the asset. The quantity bought or sold by a trader cannot exceed the quantity set in his order. There are many types of orders that can be submitted by traders, but the two most common ones are **limit orders** and **market orders**. Market orders only consist of a quantity, the buyer or seller submitting the price will accept any price that is given by the market. So a market order submitted by a buyer is equivalent to a limit order with an infinite price (and a price equal to $-\infty$ in the case of a seller).

Orders submitted by traders are not necessarily filled, i.e., the trader may not be able to buy or sell all the quantity specified in his order. For a buyer, an order is filled if the buyer's bid is at least as high as the lowest ask submitted by the sellers, and similarly for the sellers. Orders that cannot be filled (or that are only partially filled) are stored in the **book**.[7]

The standard design for CLOB uses a price-time priority. An incoming buying limit order that can be filled will be processed first using the lowest ask in the book. If all the orders corresponding to that lowest ask do not sum up to the quantity asked by the buyer additional transactions will be made using the next lowest ask, and so on until either

[7] In certain markets, particularly equities, at the opening of the market books may already contain some orders, comprising unfilled orders from the pre-opening auction.

the buyer has bought all the unit he asked or the next lowest ask is higher than his bid. In this latter case the order is only partially filled and a new order is placed in the book corresponding to the buyer's bid and the quantity is his initial submitted quantity minus the quantity he purchased. Sellers' orders are processed similarly starting with the highest bid. The difference between the lowest ask and the highest bid in the book is called the **bid-ask spread**.

Traditionally traders are not distinguished between buyers and sellers but between **providers** and **takers**.[8] A provider is any trader whose order ends up in the book (in which case we refer to his order as a **active** order). A taker is any trader whose order 'crosses the spread,' that is, a buyer whose bid is above or equal to the lowest ask in the book, or a seller whose ask is lower or equal to the highest bid in the book.[9]

2.2 Exchange architecture

For reasons of risk, cost and interoperability electronic financial exchanges—like most distributed computer systems designed to perform business-related functions—incorporate various commercial off-the-shelf (COTS) components into their implementations.[10] While inclusion of these COTS components is a practical necessity is also not without drawbacks. Since the computer industry competes largely on the speed at which components operate, vendors tend to prioritize the pursuit of speed as a design goal over other goals that might inhibit speed. In the case of fast, 'line-rate' *network switches*—those that are required to handle the volume and rate at which orders are received on a modern financial exchange—it is a deliberate design decision made by vendors along exactly these lines to provide only a handful of simple but fast *scheduling algorithms* for serializing messages the switch simultaneously receives; more sophisticated scheduling algorithms would inhibit the switch's speed (Shreedhar and Varghese, 1995; McKeown, 1997; Sivaraman et al., 2016).

To understand the centrality of a network switch to our work in this paper on fairness

[8] Providers are also called *market makers*. See Gould et al. (2013) for a nomenclature.

[9] A taker whose order is only partially filled is then first a taker and then a provider.

[10] To elaborate: the inclusion of these COTS components: (i) reduces risk because they have been proven in the field through their wide-usage in many other distributed systems, (ii) reduces costs because although the specific components used in integration tend to serve a generic purpose they also tend to be difficult and expensive to design, build and test, and (iii) improves interoperability because standardization of the interfaces they implement enable straightforward integration with other COTS components, including those of the market participant's computer systems that must be able to achieve connectivity to the exchange.

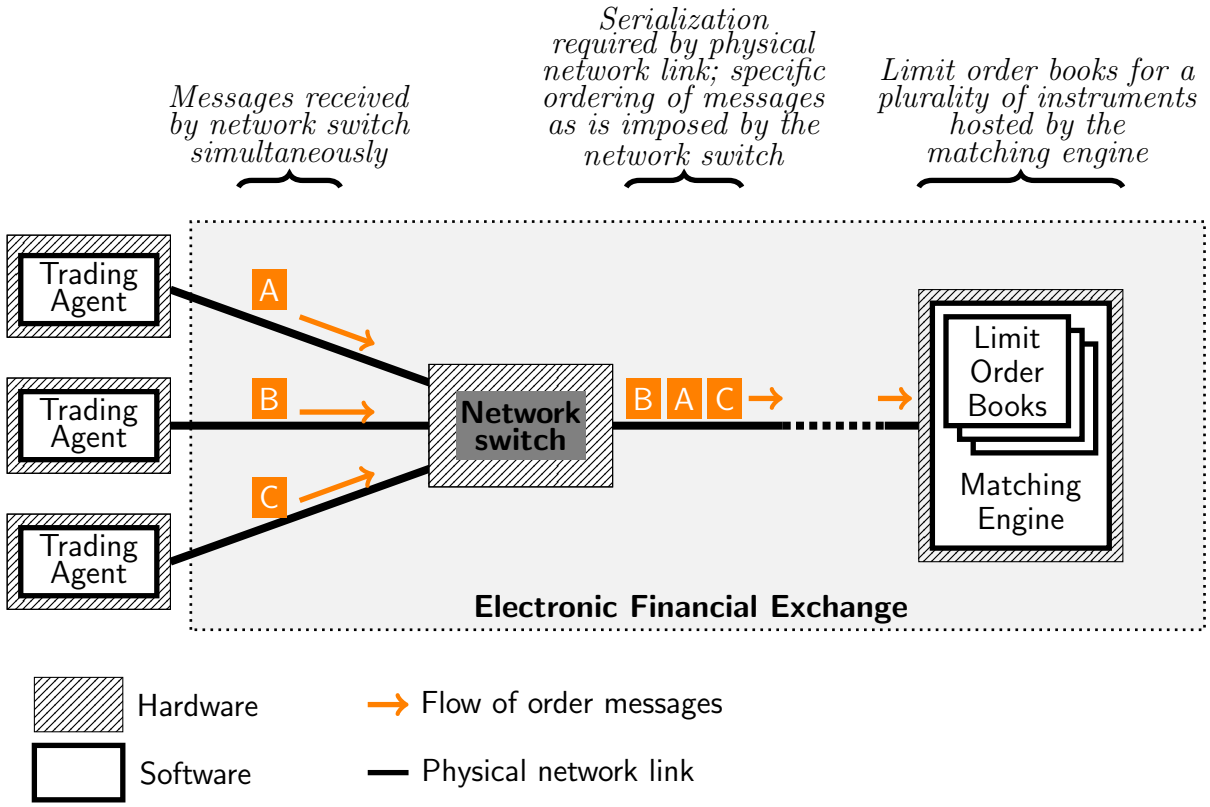

Figure 1: Architectural view of a modern electronic financial exchange

consider the architectural view in Figure 1 of a modern financial exchange.[11] In this figure market participants are simultaneously sending messages to the network switch component of the exchange, and that switch is imposing an ordering on those messages as it forwards them to the matching engine component which hosts limit order books for the instruments that trade on the exchange.[12] Abstracting from *jitter*, the ordering generated by the switch is that in which messages are processed in the CLOB.[13]

[11] From a software engineering perspective the architectural view provided in this figure is, at an appropriate level of abstraction, consistent with the published architectures of many real exchanges (CME Group, 2014; Eurex, 2016; Kluber, 2017; Melton, 2018)

[12] Switch designers use the term *output port contention* to describe the situation shown in the figure where several messages received simultaneously must all be forwarded to the same physical link (Rojas-Cessa, 2016); it is the switch's *scheduling algorithm* that performs this ordering function.

[13] Melton (2020) provides a discussion of the meaning, nature and causes of *jitter* in electronic financial exchanges, but to summarize: jitter in a computer system is the variability in the time taken to perform a given operation. It is caused by features deeply embedded in both hardware and software that exploit the spatial and temporal locality in a computer program's data and instructions that has long been observed to

3 The model

The usual playground for the random serial dictatorship mechanism (RSD) is the classic assignment model consisting of a set of objects and a set N of agents (both finite), with each individual being endowed with a preference relation over the objects. In this mechanism a queue of agents is randomly generated and each agent chooses, one at a time, an object among the objects that haven't been choosen yet.

We make two departures from this description. The first and main departure is that we assume that not all queues have a positive probability. We assume that there is a set Q of possible queues, called a **technology**. Like for RSD, all queues in Q have the same probability to be realized. We denote by $|Q|$ the number of queues in Q.

Given two queues $q = (t_1, t_2, \ldots, t_k)$ and $q' = (t'_1, t'_2, \ldots, t'_\ell)$, we say that q is a subqueue of q' if $k \leq \ell$ and $t_i = t'_i$ for $i = 1, \ldots, k$. The rank of a trader t in a queue q is denoted $r_t(q)$. For a queue q, the truncation of a queue q at trader t, denoted $q|_t$. For example, if $q = t_1, t_2, \ldots, t_{h-1}, t_h, t_{h+1}, \ldots, t_m$ then $q|_{t_h} = t_1, t_2, \ldots, t_{h-1}$.

The second departure we make is that we abstract from the description of set of objects (and agents' preferences over objects) by assuming instead that agents have preferences over queues. This is without loss of generality because each realized queue yields a unique assignment and agents' preferences are independent of the assignment of the other agents.

To sum up, a problem is given by a set N of agents, a technology Q, and for each agent $i \in N$, a payoff function $u_i : Q \to \mathbb{R}$. Given a technology and payoff profile $(u_i)_{i \in N}$, the expected outcome of an agent $i \in N$ is

$$\mathbb{E}_Q(u_i) = \frac{1}{|Q|} \sum_{q \in Q} u_i(q). \tag{1}$$

Most of our results essentially rely on a simple identification argument. We will thus make the assumption that the possible payoff profiles are sufficiently *rich* so that for any trader there exists a payoff profile such that for any two distinct queues the trader obtains different payoffs. We will make this assumption more precise depending on the case we consider.

exist in its execution so as to make it run more quickly. See also Baldauf and Mollner (2020) for an analysis of the impact of jitter on market performance.

4 A trading model

The model we outlined in the previous section is fairly general, with a description more adapted for the Social Choice literature. We discuss in this section how it can also fit a trading problem.

4.1 Remark on the simultaneity assumption

There is both empirical evidence and explanatory theory to suggest this phenomenon is widespread.[14] In terms of empirical evidence, Brolley and Zoican (2019) estimate that around 10% of the time a typical exchange is processing 'bursts' of orders received substantially simultaneously. On specific exchanges: We observe that on Refinitiv Matching, of all the millisecond timestamps in which orders were received in 2014-2015, approximately 7% of those timestamps contained a plurality of orders[15]; Lohr and Neusüß (2019) observe on Eurex that there are many millions of instances of a plurality of orders being received within a few nanoseconds of one another; Aquilina et al. (2020) observe on the London Stock Exchange that 20% of trading volume involves a plurality of orders received with 5-10 microseconds of one another; and Menkveld (2018) observes on NASDAQ that around 20% of trades involve a plurality of orders received within a sub-millisecond window.

[14] In terms of the explanatory theory of why exchanges receive bursts of orders from fast traders, there are two that are relevant to our work: one involves the submission of orders that are competing for the same resource, and the other involves the submission of non-competing orders. Farmer and Skouras (2012) note that responsive to a publicly observable signal a plurality of fast participants may submit *competing* orders to (i) remove liquidity (e.g., because of mispriced bid or offer in the CLOB), or (ii) to obtain a favorable queue position in the CLOB when providing liquidity by submitting a bid or offer at the same price-level in it. R. Roth (2019) notes that a burst of orders may result from a liquidity provider updating their bids and offers on an option chain responsive to a change in the chain's underlying cash instrument—in this case the orders are not competing with one another, and the explanation naturally generalizes to any instruments with correlated pricing. The two explanatory theories, of course, are not incompatible and as is further noted by R. Roth (2019), a burst of orders may contain a mix of competing and non-competing orders from a plurality of participants.

[15] Source: author's own analysis of Refinitiv Matching in his capacity as a Refinitiv employee.

4.2 Queuing technologies

Since orders are processed serially, one at a time, orders arriving simultaneously at the exchange must be ordered into a queue.[16] Those queues are generated by a switch, a device containing several ports to which traders are assigned and that receive traders' orders, and one additional port through which orders will be sent to the matching engine.[17] In very simplified terms switches operate as follows. Upon their arrival to a switch's port orders are buffered at the port level until they are forwarded to the matching engine. Switches visit ports one at a time (with each visit consisting of flushing the corresponding buffer) following a sequence determined by the switch's design. Importantly, trading events and the switch status (i.e., which port is currently visited by the switch) are independent. So, from the perspective of traders' at any moment each port has equal probability to be the first one to be open at the time the traders' orders arrive at the exchange.

Example 1 (circular switch) Of the switch scheduling algorithms the only one that 'attempts' to provide equal treatment of messages received simultaneously is the *deficit Round-robin*.[18] Ignoring the *deficit* aspect of the algorithm its *Round-robin* aspect is simply a circular shift on the switch's ports.[19] What this means is that if we label those ports 1 through n the priority in which simultaneously received messages will be processed can be expressed in terms of the following n permutations of those port numbers: [1,2,3,...,n], [2,3,...,n,1], ..., [$n-1$, n,1,2, ..., $n-2$], and [n,1,2, ..., $n-2$, $n-1$].

As an illustration, asssume that there are as many ports as traders, with trader i assigned to port $\sharp i$, $i = 1, \ldots n$.[20] A circular switch will visit port $\sharp 1$ through port $\sharp n$, and then start

[16]The natural terminology would be to refer to *orderings* of traders' orders. To avoid any confusion with the financial terminology 'orders' will always refer to traders' buy/cancel messages and 'queues' to linear orderings of those orders manifested as messages.

[17]The matching engine in a an electronic trading exchange is the processor where traders' orders will be executed.

[18]To abstract away from the notion of a switch we subsequently refer to these scheduling algorithms as *queuing technologies*, and the orderings they produce as *queues*.

[19]For the interested reader, the *deficit* aspect of the algorithm seeks to rectify unfairness in network bandwidth allocation by the switch that would otherwise occur in the *round robin* aspect when the size of messages received by the switch vary by sender (Shreedhar and Varghese, 1995). Throughout this work it is our implicit and optimistic assumption that messages are of equal size and that each is an order sent to the exchange by a trader. Under these assumptions *deficit round robin* in the switch implies plain old *round robin* on orders in the CLOB.

[20]In practice many traders may be assigned to a same port, or the a single trader may occupy multiple

over at port ♯1. If, for instance, at the time traders' orders arrive the switch is currently visiting, say, port ♯k, then trader i_k's order will be the first to be forwarded to the matching engine, followed by trader i_{k+1}'s order (who arrived at port ♯$k+1$), ... trader i_n's order, trader i_1's order, ... and trader i_{k-1}'s order will be the last order received by the matching engine.

Hence, depending on which port is currently being visited at the time traders' orders arrive at the switch, we have then n possible queues, depicted in Table 2. Readers familiar with the Social Choice literature will recognize this collection of queues as a Condorcet cycle.

q_1	q_2	q_3	$\cdots$	q_{m-1}	q_m
i_1	i_2	i_3	$\cdots$	i_{m-1}	i_m
i_2	i_3	i_4	$\cdots$	i_m	i_1
i_3	i_4	i_5	$\cdots$	i_1	i_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
i_m	i_1	i_2	$\cdots$	i_{m-2}	i_{m-1}

Table 2: Queues in circular switch

Upon the arrival of traders' orders each port has equal probability to be the port currently visited by the switch, so each queue in the Condorcet cycle occurs with probability $\frac{1}{m}$.

Example 1 illustrates the difference between the 'standard' RSD, where there are $n!$ possible queues of traders, each with probability $\frac{1}{n!}$, and the RSD implemented by high speed switch, where the number of possible queues is vastly reduced.

4.3 Traders' outcomes

When considering trading the payoff function $u(\cdot)$ may take different forms, depending on whether t is a taker or a provider, although for both takers and providers it will depend on the book at the time their order is being processed, i.e., on how the book has been updated with the execution (or not) of the orders of the traders that were ranked before.

For clarity we outline now possible forms that the function $u_t(\cdot)$ may take, depending on whether the trader t is a taker or a provider. For simplicity we will consider the case of liquidity takers buying security X and providers canceling their active (selling) orders. We

ports. We leave aside such complications.

abstract from quantities in traders' orders, focusing only on the bids and asks.[21] To this end, denote by B the CLOB at the time traders submit their orders and a_0 the lowest ask in B. Since we focus on the case of traders sending buying orders or canceling selling orders a LOB is simply a distribution f_B over $[a_0, \infty)$ that indicates, for each price level, how many selling limit orders where submitted at that price. The execution of traders' orders will modify the book. For a sequence $q = (i_1, i_2, \ldots, i_k)$ of traders' orders we denote by $B(q)$ the book after the orders $i_1, i_2, \ldots, i_k$ have been processed. The lowest ask is $B(q)$ is denoted $a(B(q))$.

4.3.1 Providers' outcomes

Providers' do not necessarily have the same active orders. Hence, whether they manage to cancel their orders depend on how the value of the lowest ask in the book when their order is processed and the price they ask. To this end, let p be a provider and let a_p the ask in provider p's active order. Providers' payoffs are normalized to 1 if they managed to cancel their order and to 0 otherwise (i.e., a taker bought the instrument from them). So for any provider p, $u_p(q)$ is given by

$$u_p(q) = \begin{cases} 0 & \text{if } a(B(q|_p)) > a_t \\ 1 & \text{if } a(B(q|_p)) \leq a_t \end{cases} \tag{2}$$

Note that the payoff description in Eq. (2) takes some liberty with the standard CLOB trading protocol that uses a price-time priority rule. In the standard CLOB design if two providers posted the same ask it is the oldest provider who will be matched with the taker. To see this consider a queue $q = (t_1, t_2, \ldots, t_{k-1}, t_k, t)$ such that t_k is a taker and p is a provider and

$$a(B(t_1, t_2, \ldots, t_{k-1})) = a_t$$
$$a(B(t_1, t_2, \ldots, t_{k-1}, t_k)) = a_t.$$

This implies that there are at least two providers who asked the price a_t who haven't cancelled yet when t_k's order was processed. By setting $u_t(q) = 1$ we implicitly assume that provider t's order is necessarily the most recent among all the providers who ask for a price equal to a_t.

[21] Not taking into account quantities in traders' orders has no impact here. Also, note that in practice for most securities orders are usually of the same size.

4.3.2 Takers' outcomes

Each taker $t \in T$ submits a limit order with bid b_t. A taker arriving just after trader t_k will thus buy the instrument only if $a(B(q)) \leq b_t$. Note that for any two queues q, q' such that $q \subset q'$, we have $a(B(q)) \leq a(B(q'))$.

Similarly to providers, takers' outcomes depend on the the book at the time their order is processed, which in turns depends on the set of takers and providers that are ranked before them. Hence,

$$q|_t \text{ is a subqueue of } q'|_t \quad \Rightarrow \quad u_t(q) \geq u_t(q'). \tag{3}$$

This definition of takers' outcomes encompasses different metrics. A first natural metric is the price at which a trader will be served. For instance, if each taker only buys one unit of the instrument a taker's payoff takes the following form,

$$u_t(q) = \begin{cases} v(a(B(q|_t))) & \text{if } a(B(q|_t)) \leq b_t \\ C & \text{if } a(B(q|_t)) > b_t \end{cases} \tag{4}$$

where C is a constant with $C > a_t$ and v_t is strictly decreasing. If a taker t arrives too late, then his order will not be succesful. Setting that the price is get is equal to some large constant C is without loss of generality as long as we are interested in having takers obtaining, in expectation, the same price. Otherwise one needs for modify Eq. (1) by taking the expectation conditional on being served.

Another metric is simply looking at whether a taker is served (still assuming that the taker only buys one unit of the instrument),

$$u(q) = \mathbb{1}_{a(B(q|_t) \leq a_t} \tag{5}$$

Note that the specification given by Eq. (4) guarantees that if two traders obtain the same expected "price" then their expectation of being served is also the same. This would not be the case if we would look at the expected price conditional on being served.

5 Access fairness

The simplest form of fairness boils down to whether traders have equal chances to be ranked first, second, ... in the realized queue. This notion fairness can be generalized at no cost by

assuming that traders' payoffs only depend on their positions in the queue. Thus, we assume that for each trader there is a payoff $v_i : Q :\to \mathbb{R}$ such that for any two queues $q, q' \in Q$,

$$r_i(q) = r_i(q') \qquad \Rightarrow \qquad v_i(q) = v_i(q').$$

Since $v_i(q)$ only depends on $r_i(q)$ for all q and all traders $i \in N$, abusing notation we will sometime write $v_i(k)$ to denote $v_i(q)$ for $r_i(q) = k$.

Given a queue $q \in Q$, and two traders $i, j \in N$, let $q^{i\leftrightarrow j}$ denote the queue q where i and j's positions are interchanged, that is, $q^{i\leftrightarrow j}$ is such that

$$r_h(q^{i\leftrightarrow j}) = \begin{cases} r_i(q) & \text{if } h = j \\ r_j(q) & \text{if } h = i \\ r_h(q) & \text{if } h \neq i, j \end{cases} \tag{6}$$

and denote by $Q^{i\leftrightarrow j} = \cup_{q\in Q} q^{i\leftrightarrow j}$. We say that two traders $i, j \in N$ are **symmetric** if $v_i(q) = v_j(q^{i\leftrightarrow j})$ for all $q \in Q$.

For this section we make the following assumption. Among the n traders who simultaneously submit an order to an exchange of particular interest is the subset of the traders who submit orders for the same financial instrument, say, X. The other traders have orders for a different instrument (but that is traded at the same exchange). This assumption is in line with the current practice in electronic exchanges where various instruments are traded on the same matching engine (i.e., the same processor). Hence, traders buying and selling different instruments are ordered in the same queues by the same switch. We call traders trading instrument X **essential traders**. The presence of essential traders implies that their payoffs for a given queue q only depends on the subqueue q restricted to the essential traders.

Given a technology Q, where $Q = \{q_1, q_2, \ldots, q_k\}$ and a set $S \subseteq N$ of traders we denote by Q^S the collection of queues $\{q_1^S, q_2^S, \ldots, q_k^S\}$ such that for each q_h^S, $h = 1, \ldots, k$, q_h^S is a restriction of q_h to the set S, i.e., for each pair of traders $t, t' \in S$, $r_t(q_h^S) < r_{t'}(q_h^S)$ if, and only if $r_t(q_h) < r_{t'}(q_h)$.

Definition 1 A set $S \subseteq N$ of traders is **essential** if for any trader i and queue $q \in Q$, $v_i(q) = v_i(q^S)$.

Definition 2 A technology guarantess **access fairness** if for any payoff profile $(v_i)_{i\in N}$, and any two symmetric traders $i, j \in N$,

$$\mathbb{E}_Q(v_i) = \mathbb{E}_Q(v_j). \tag{7}$$

5.1 Trading considerations

From a trading perspective access fairness can also capture the situation where all the traders competing for the instruments are liquidity takers who submit the same order.[22] This is for instance the case analyzed by Budish et al. (2015), where takers are high frequency traders and provider traditional (slower) investors.

The next lemmata formalize this. They are immediate consequences from the fact that takers are all symmetric, so their proofs are omitted.

Lemma 1 Let Q be a technology, and assume that $T \neq \emptyset$ and $P = \emptyset$. For any $t \in T$ such that $r_t(q) = r_t(q')$ for $q, q' \in Q^T$, then for any book B and traders' orders $a(q|_t) = a(q'|_t)$.

Lemma 2 Let Q be a technology, and assume that $T \neq \emptyset$ and $P = \emptyset$. For any $t, t' \in T$ and $q, q' \in Q^T$ such that $r_t(q) = r_{t'}(q')$, $r_t(q') > r_{t'}(q')$ and $r_{t'}(q) > r_t(q)$, we have, for any book B, $\bar{a}$, $a(q|_t) = a(q'|_{t'})$.

More generally, in the case where fast takers are 'sniping', i.e., competing against similarly fast providers who are seeking to cancel their active bids or offers, one can view a provider seeking to cancel her own order to be equivalent to that provider sending a marketable order to match against her own bid or offer. While much of the literature makes a distinction between a provider's cancellation and takers' marketable orders (see e.g., Budish et al. (2015) and Baldauf and Mollner (2019)) both have the same effect of removing liquidity from the book, that is, causing a previously active order to become inactive —as in Wuyts (2012) or Mounjid et al. (2019). Crucially though, as both Farmer and Skouras (2012) and Li et al. (2019) have independently observed, the economic consequences for a provider are equivalent: if one views the provider's cancel as just that their loss is zero; if one views the

[22]This case is more interesting than the providers-only case because traders' outcomes worsen as their position in the queue gets lower; if there were only providers canceling their orders any technology that consists of always ranking all providers would be trivially fair.We will consider the case where there are only providers (with non-cancelling orders) in Section 6.3.

provider's cancel instead as a taker order then their loss is also zero because their profit on the taker order is exactly equal to their loss on their provider order.[23]

5.2 Characterizing fair access technologies

Access fairness intuitively requires that all traders have the same rank distribution over the queues in Q, that is, there are as many queues ranking a trader t in position k as there are queues ranking any other trader t' in position k, for $k = 1, \ldots, n$. The next example shows that this is not enough. Fairness breaks down in a constrained serial dictatorship mechanism as soon as the set of essential traders is a strict subset of the set of traders.

Example 2 There are four traders, i_1, i_2, i_3 and i_4. Traders i_1 and i_2 are essential traders, while i_3 and i_4 trade another instrument (so i_3 and i_4's orders do not affect the book for instrument X and thus i_1 and i_2's payoff).

The technology we consider is the Condorcet cycle, which corresponds to the Round-robin switch we described in Example 1. There are four ports, g_1, g_2, g_3 and g_4, and traders i_1, i_2, i_3 and i_4 are assigned to ports g_1, g_2, g_3 and g_4, respectively. The technology Q is made of the four queues depicted in Table 3.

q_1	q_2	q_3	q_4
$\boxed{i_1}$	$\boxed{i_2}$	i_3	i_4
$\boxed{i_2}$	i_3	i_4	$\boxed{i_1}$
i_3	i_4	$\boxed{i_1}$	$\boxed{i_2}$
i_4	$\boxed{i_1}$	$\boxed{i_2}$	i_3

Table 3: Circular switch with four traders

Note that trader i_1 is ranked first among the essential traders three times out of four.

[23] One might argue that the consequences are *not* equivalent because in one case there is a transfer of profit from one group of participants (providers) to another (takers) when a cancel does not succeed. In reality however, the roles of provider and taker are not fixed and a participant alternate roles dynamically. Empirically Dahlström (2019) finds that high-speed 'makers' are actually *net beneficiaries* of latency arbitrage because besides canceling their own quotes, they also simultaneously act as takers by sending marketable orders on others' stale quotes.

For instance $v_{i_1}(q_4) = v_{i_1}(q_4^{\{i_1,i_2\}}) = v_{i_1}(1)$. So traders i_1 and i_2's expected outcomes are

$$\mathbb{E}_Q(u_{i_1}) = \frac{1}{4}\left(3v(1) + v(2)\right) \qquad \text{and} \qquad \mathbb{E}_Q(u_{i_2}) = \frac{1}{4}\left(v(1) + 3v(2)\right),$$

which implies that this technology cannot be fair.

The technology of Example 2 fails to be fair for a lack of symmetry in the queues regarding traders i_1 and i_2. To see this, notice that the case in the example is equivalent to the one where there are only takers t_1 and t_2, but with the technology $Q' = (q'_1, q'_2, q'_3, q'_4)$,

q'_1	q'_2	q'_3	q'_4
i_1	i_2	i_1	i_1
i_2	i_1	i_2	i_2

While Q treats each trader equally (they have equal probability to be ranked at any position), it fails to do so when we consider only traders i_1 and i_2. Since the technology must work for any set of takers one needs a symmetry property that holds for any set S of essential traders, which we introduce now.

Definition 3 A technology Q is **symmetric** if for any pair of traders $i, j \in N$

$$|\{q \in Q \ : \ r_i(q) = k\}| = |\{q \in Q \ : \ r_j(q) = k\}| \qquad \text{for all } k = 1, \ldots, n, \tag{8}$$

and a technology Q is **strongly symmetric** if for any set $S \subseteq N$ of traders the technology Q^S is symmetric.

Our definition of symmetry compares traders, for each rank, but does not compare how many times traders are ranked for different ranks. It turns out that this if a technology Q is symmetric then each trader is ranked the same number of times for each rank.

Proposition 1 If a technology Q is symmetric then for any traders $i \in N$,

$$|\{q \in Q \ : \ r_i(q) = k\}| = |\{q \in Q \ : \ r_i(q) = k'\}| \qquad \text{for all } k, k' = 1, \ldots, n. \tag{9}$$

Proof. There is necessarily a queue $q \in Q$ and a trader $i \in N$ such that $r_i(q) = 1$. Let $z = |\{q \in Q \ : \ r_i(q) = 1\}|$. Since Q is symmetric then $|\{q \in Q \ : \ r_j(q) = k\}| = z$, for all $j \in N$, and thus there are $z|N|$ queues in Q. Without loss of generality suppose that for some rank h and trader $i \in S$ we have $|\{q \in Q \ : \ r_i(q) = h\}| = z' < z$. Since Q is

symmetric $|\{q \in Q \ : \ r_j(q) = h\}| = z'$, for all $j \in N$. So there are $z'|N| < z|N|$ queues in Q, a contradiction. The case if $z' > z$ is similar. ■

Our richness assumption regarding access fairness is the following.

Assumption 1 (rich payoffs) For any trader $i \in N$, there exists a payoff profile $(v_j)_{j\in N}$ such that for any two queues $q, q' \in Q$ with $r_i(q) \neq r_i(q')$, $v_i(q) \neq v_i(q')$.

We are now ready to characterize fair market access technologies.

Proposition 2 A technology Q guarantees access fairness if, and only if Q is strongly symmetric.

Proof. That a strongly symmetric technology guarantees access fairness is straightforward. We only show the converse. Let $S \subseteq N$ be a set of essential traders, and let Q be a technology that guaranteess access fairness. So, for any two symmetric traders $i, j \in S$,

$$\frac{1}{|Q|}\sum_{q\in Q} v_i(q) = \frac{1}{|Q|}\sum_{q\in Q} v_j(q) \tag{10}$$

$$\Leftrightarrow \qquad \sum_{k=1}^{k=n}\sum_{\substack{q\in Q\\ r_i(q)=k}} v_i(q) = \sum_{k=1}^{k=n}\sum_{\substack{q\in Q\\ r_j(q)=k}} v_j(q)\,. \tag{11}$$

Since i and j are symmetric we have $v_i(k) = v_j(k)$ for all k. Also, S being a set of essential traders i and j's outcomes only depend on the queues restricted to S. So Eq. (11) is equivalent to

$$\sum_{k=1}^{k=n}\sum_{\substack{q\in Q^S\\ r_i(q)=k}} v(k) = \sum_{k=1}^{k=n}\sum_{\substack{q\in Q^S\\ r_j(q)=k}} v(k) \tag{12}$$

$$\Leftrightarrow \qquad \sum_{k=1}^{k=n}\pi(i,k)v(k) = \sum_{k=1}^{k=n}\pi(j,k)v(k)\,, \tag{13}$$

where

$$\pi(i,k) = \left|\{q \in Q^S \ : \ r_i(q) = k\}\right| .$$

Since Eq. (13) must holds for any profile $(v_i)_{i\in N}$, by identification we obtain

$$\pi(t,k) = \pi(t',k), \quad \text{for } k = 1,\ldots,n.$$

That is, Q^S must be symmetric and thus Q is strongly symmetric. ■

6 Outcome fairness

When considering agents' outcome there is a crucial difference between the standard object allocation and trading. In the context of object allocation with (constrained) RSD as the allocation mechanism, for any realized queue an agent's outcome only depends on the subqueue up to that agent. The relative ranking of the agents that are ranked below him has no impact on his payoff.[24] This property does not hold for electronic trading, however, for a trader's outcome may also depend on the relative ranking of traders that are ranked below him. The following example illustrates this point.

Example 3 There are two sellers, s_1 and s_2 and two buyers, b_1 and b_2. All buyers and sellers submit an order for the same quantity. The prices proposed by s_1, s_2, b_1 and b_2 are \$14, \$11, \$12, and \$13, respectively. Consider the following two queues,

$$q = s_1, b_1, s_2, b_2 \qquad \text{and} \qquad q' = s_1, b_1, b_2, s_2\,.$$

For b_1 the subqueues obtained from q and q' by only considering the traders above him are identical. So in an object allocation problem b_1 should receive the same outcome for both q and q'. That is not the case in a trading context. Under both q and q' trader b_1 is a provider, i.e., his order cannot be executed and is thus stored in the book. Under q trader s_2 is a liquidity taker. His order will cross the spread thus b_1 will buy the instrument (for a price equal to his bid, \$12). Under q' trader b_2 is a liquity provider and thus his order is stored in the book. Then comes s_2 who will sell to b_2 at a price of \$13. Hence, b_1 will not manage to buy the instrument under q'.

In object allocation problems two agents are said to be symmetric if they have the same preferences over the objects. In the context of trading two traders i and j are symmetric if they submitted the same order.[25] Since we are working here with payoffs depending on queues this translates as two traders having the same payoff function over queues. Note that since a queue is an ordering of *all* agents, for any queue two supposedly symmetric agents will have different position on that queue, and thus their outcomes may not be equal. The following defines agents' symmetry when payoffs are defined over queues.

Definition 4 Two traders $i, j \in N$ are **symmetric** if $u_i(q) = u_j(q^{i \leftrightarrow j})$ for all $q \in Q$.

[24] see for instance Abdulkadiroğlu and Sönmez (1998).

[25] So a taker and a provider cannot be identical in our setup.

Note that $u_j(q^{i\leftrightarrow j})$ denotes the payoff that j would obtain if he had taken i's position in the queue q (and i took j's position). If i and j have the same preferences over objects or i and j are traders who submitted the same order then j would obtain under $q^{i\leftrightarrow j}$ the same outcome as i would get under q.

Definition 5 A technology Q guarantees **outcome fairness** if, for any payoff profile $(u_i)_{i\in N}$, and any two symmetric traders $i, j \in N$

$$\mathbb{E}_Q(u_i) = \mathbb{E}_Q(u_j). \tag{14}$$

6.1 Outcome fairness for object allocation

We consider here the standard setup for object allocation, which means that for any agent and any realized queue his outcome only depends on the subqueue up to him. Formally, we assume here that for any trader $i \in N$ and queues $q, q' \in Q$, if $q|_i = q'|_i$ then $u_i(q) = u_i(q'|_i)$. In this context our richness assumption is the following.

Assumption 2 (rich payoffs) For any trader $i \in N$, there exists a payoff profile $(u_j)_{j\in N}$ such that for any two queues $q, q' \in Q$ with $q|_i \neq q'|_i$, $u_i(q|_i) \neq u_i(q'|_i)$.

For any two traders i and j, we denote by $q|_j^{i\leftrightarrow j}$ the queue q that is first truncated at j and then i and j are swapped. That is, $q|_i^{i\leftrightarrow j} = q|_i$ if $j \notin q|_i$, and otherwise $q|_i^{i\leftrightarrow j}$ is such that $j \notin q|_i^{i\leftrightarrow j}$ and

$$r_h(q|_i^{i\leftrightarrow j}) = \begin{cases} r_h(q|_i) & \text{if } h \neq i\,, \\ r_j(q|_i) & \text{if } h = i\,.^{26} \end{cases}$$

Definition 6 A technology Q is **balanced** if for any pair i, j of traders, $\{q|_i\}_{q\in Q} = \{q|_j^{i\leftrightarrow j}\}_{q\in Q}$.

In other words, a technology is balanced if any queue q truncated at i can be mapped uniquely to a queue truncated at j that is identical, up to a permutation of i and j.

We are now ready to characterize outcome fairness.

Proposition 3 A technology Q guarantees outcome fairness if, and only if Q is balanced.

Proof. Showing that a balanced technology guarantees outcome fairness is straightforward and is left to the reader. We prove the converse. Let Q be a technology that guarantees

outcome fairness. So, for any two symmetric traders $i, j \in N$,

$$\frac{1}{n}\sum_{q\in Q} u_i(q) = \frac{1}{n}\sum_{q\in Q} u_j(q) \tag{15}$$

$$\Leftrightarrow \qquad \sum_{\substack{q\in Q\\ j\notin q|_i}} u_i(q|_i) + \sum_{\substack{q\in Q\\ j\in q|_i}} u_i(q|_i) = \sum_{\substack{q\in Q\\ i\notin q|_j}} u_j(q|_j) + \sum_{\substack{q\in Q\\ i\in q|_j}} u_j(q|_j) \tag{16}$$

Note that for $q \in Q$, if $i \notin q|_j$ then $q|_j = q|_j^{i\leftrightarrow j}$, and since i and j are symmetric, $u_j(q|_j) = u_i(q|_j^{i\leftrightarrow j})$. Similarly, if $i \in q|_j$ then i and j's symmetry implies $u_j(q|_j) = u_i(q|_j^{i\leftrightarrow j})$. Hence, Eq. (16) is equivalent to

$$\sum_{\substack{q\in Q\\ j\notin q|_i}} u_i(q|_i) + \sum_{\substack{q\in Q\\ j\in q|_i}} u_i(q|_i) = \sum_{\substack{q\in Q\\ i\notin q|_j}} u_i(q|_j^{i\leftrightarrow j}) + \sum_{\substack{q\in Q\\ i\in q|_j}} u_i(q|_j^{i\leftrightarrow j})\,. \tag{17}$$

Note that for the first sums of the left-hand side and right-hand side i's payoff is calculated for similar types of subqueues, i.e., subqueues of the form $h_1, h_2, \ldots, h_k$, with $j \neq h_k$ for $\ell = 1, \ldots, k$.[27] Similarly, the second sums of both sides are over subqueues of the form $h_1, h_2, \ldots, h_k$, with $j = h_\ell$ for some $\ell = 1, \ldots, k$. Hence, Assumption 2 implies using an identification argument that $\{q|_i\}_{q\in Q}$ —from the left-hand side of Eq. (17)— is the same as $\{q|_j^{i\leftrightarrow j}\}$. That is, Q is balanced. ■

Proposition 3 is relatively intuitive. If i and j are symmetric traders, then for any $q \in Q$ we must have $u_i(q|_i) = u_j(q|_i^{i\leftrightarrow j})$. Symmetry is thus guaranteed if for any subqueue $q|_i$ in Q, the subqueue $q|_i^{i\leftrightarrow j}$ is also in Q, which is ensured by balancedness.

6.2 Outcome fairness for trading

We now consider the more general case when a trader's payoffs may not only depend on the relative ordering of the traders ranked above him but also on the relative ordering of the agents ranked below him. The richness assumption then becomes the following.

Assumption 3 (rich payoffs) For any trader $i \in N$, there exists a payoff profile $(u_j)_{j\in N}$ is such that for any two distinct queues $q, q' \in Q$, $u_i(q) \neq u_i(q')$.

[27] The first sum of the left-hand side is for queues $q \in Q$ such that i is ranked below j, i.e., $i \notin q|_j$. For those queues the sum calculate $u_i(q|_j^{i\leftrightarrow j})$, but it is easy to see that $j \notin q|_j^{i\leftrightarrow j}$.

The concept of balancedness can easily be extended to this case, except that now we consider the entire queues in Q.

Definition 7 A technology Q is **fully balanced** if for any $i, j \in N$ and $q \in Q$, $\{q\}_{q\in Q} = \{q^{i\leftrightarrow j}\}_{q\in Q}$.

We then easily obtain an analog of Proposition 3.

Proposition 4 A technology Q guarantees outcome fairness if, and only if Q is fully balanced.

The proof is similar to that of Proposition 3. The main difference is relative to the trader's payoffs: If i and j are symmetric then we have

$$\frac{1}{n}\sum_{q\in Q} u_i(q) = \frac{1}{n}\sum_{q\in Q} u_j(q) \tag{18}$$

$$\Leftrightarrow \quad \sum_{\substack{q\in Q \\ j\notin q|_i}} u_i(q) + \sum_{\substack{q\in Q \\ j\in q|_i}} u_i(q) = \sum_{\substack{q\in Q \\ i\notin q|_j}} u_i(q^{i\leftrightarrow j}) + \sum_{\substack{q\in Q \\ i\in q|_j}} u_i(q^{i\leftrightarrow j})\,. \tag{19}$$

Then from Assumption 3 and by identification we obtain that Q must be fully balanced.

In appearance fully balancedness looks like being more demanding than balancedness. We show in Section 7.2 that these two concepts are actually equivalent.

6.3 Competing providers

A special case of the situation analyzed in the previous section is when the burst of orders received by the exchange consist of non-competing orders, that is, when none of the orders submitted by the traders cross the spread and thus they are all providers (who are not necessarily canceling their orders).

This case is not as hypothetical as it may seem; it is in fact relatively common for an exchange to receive burst of non-crossing orders (i.e., a batch of orders with a least one bid and one ask such that the ask is at least as high as the bid). For instance, for 2014 and 2015 there are 41,981,349 and 66,163,815 instances where Refinitiv received burst of orders within a millisecond timestamp. Out of those, only 44,938 and 52,687 (for 2014 and 2015, respectively) contained crossing orders, i.e., less than 0.1% of the time.

We generalize the problem by assuming that traders' orders may differ and orders are not necessarily from the same side of the market, i.e., some may be buying orders and others

selling orders. In this case providers' outcomes depart from the definition outlined in Section 4.3.1 because a provider's outcome ultimately depends on the orders received before and after him.[28] Note, however, that the relative ranking of the orders above his, as well as the relative ranking of the offers ranked below his, do not have any impact on the providers' outcome. Hence, a sufficient statistic to capture a provider's outcome is the set of orders ranked above his.

To this end, given a technology Q and a queue $q \in Q$, denote by $U_i(q)$ the upper-contour set of trader i in q, $U_i(q) = \{j \in N \ : \ r_j(q) < r_j(q)\}$, and let $\mathbf{U}_i(Q) = \cup_{q \in Q} U_i(q)$. When all traders are providers we have

$$U_i(q) = U_i(q') \qquad \Rightarrow \qquad u_i(q) = u_i(q') \ . \tag{20}$$

The richness assumption in the case where there are only providers is the following,

Assumption 4 (rich payoffs) For any trader $i \in N$, there exists a payoff profile $(u_j)_{j \in N}$ such that for any two queues q, q' such that $U_i(q) \neq U_i(q')$, $u_i(q) \neq u_i(q')$.

For a set $S \subseteq N$, let $S^{i \leftrightarrow j}$ be the set where i and j are swapped. That is,

$$S^{i \leftrightarrow j} = \begin{cases} S & \text{if } i, j \notin S \\ S \cup \{j\} \backslash \{i\} & \text{if } i \in S, j \notin S \\ S \cup \{i\} \backslash \{j\} & \text{if } i \notin S, j \in S \\ S & \text{if } i, j \in S \end{cases}$$

We denote by $\mathbf{U}_i^{i \leftrightarrow j}(Q) = \cup_{q \in Q} (U_i(Q))^{i \leftrightarrow j}$.

Definition 8 A technology Q is **weakly balanced** if for any $i, j \in N$,

$$\mathbf{U}_i(Q) = \mathbf{U}_j^{i \leftrightarrow j}(Q) \tag{21}$$

In short, in a weakly balanced technology the upper-contour sets across all queues of any two traders must coincide, up to a permutation of these two traders.

Proposition 5 Let the set of traders be only composed of liquidity providers. A technology Q guarantees outcome fairness if, and only if, Q is weakly balanced.

[28] A providers' outcome may depend from the orders received after a his if at a later date the provider seeks to cancel. The more orders with a lower ask (if it is a selling order, or a higher bid if it is a buying order) will give higher chances to the provider to cancel before being sniped by takers.

Proof. The *if* part is straightforward. We show the *only if* part. Let Q be a fair technology and let i, j be two symmetric providers. Since Q is fair we have

$$\frac{1}{n}\sum_{q\in Q} u_i(q) = \frac{1}{n}\sum_{q\in Q} u_j(q) \tag{22}$$

$$\Leftrightarrow \qquad \sum_{\substack{q\in Q\\ j\notin q|_i}} u_i(q) + \sum_{\substack{q\in Q\\ j\in q|_i}} u_i(q) = \sum_{\substack{q\in Q\\ i\notin q|_j}} u_j(q) + \sum_{\substack{q\in Q\\ i\in q|_j}} u_j(q)\,. \tag{23}$$

Define the function $w : 2^N \to \mathbb{R}$ as $w_i(U_i(q)) = u_i(q)$. So using Eq. (20), Eq. (23) is equivalent to,

$$\sum_{\substack{S\in \mathbf{U}_i(Q)\\ j\notin S}} w_p(S) + \sum_{\substack{S\in \mathbf{U}_i(Q)\\ j\in S}} w_p(S) = \sum_{\substack{S\in \mathbf{U}_j(Q)\\ i\notin S}} w_j(S) + \sum_{\substack{S\in \mathbf{U}_j(Q)\\ i\in S}} w_j(S)\,. \tag{24}$$

Note that for $q \in Q$ such that $i \notin U_j(q)$, since i and j are symmetric, $w_j(U_j(q)) = w_i(U_j(q))$. If q is such that $i \in U_j(q)$, then, again from i and j being symmetric, $w_j(U_j(q)) = w_p(U_j(q) \cup \{j\}\backslash\{i\})$. Hence, (24) is equivalent to

$$\sum_{\substack{S\in \mathbf{U}_i(Q)\\ j\notin S}} w_i(S) + \sum_{\substack{S\in \mathbf{U}_i(Q)\\ j\in S}} w_i(S) = \sum_{\substack{S\in \mathbf{U}_j(Q)\\ i\notin S}} w_i(S) + \sum_{\substack{S\in \mathbf{U}_j(Q)\\ i\in S}} w_i(S\cup\{j\}\backslash\{i\})\,. \tag{25}$$

Observe that the firs sums of the left-hand side and right-hand side are over similar sets, i.e., sets S such that $j \notin S$, and similarly for the second sums of each side (i.e., sets S such that $j \in S$). So from Assumption 4 and using an identification argument the first (resp. second) sums of both sides of Eq. (25) are over the same sets, which implies that Q is weakly balanced. ■

7 Comparing technologies

7.1 Strong symmetry and weak balancedness

Balancedness is obviously a more stringent requirement than weak balancedness. The following example shows that when both providers and takers are competing weak balancedness is not enough to ensure fairness.

Example 4 There four traders, p_1, p_2, t_3 and t_4, each selling or buying one unit of instrument X. Providers p_1 and p_2 have an order in the book with an ask equal to \$1 and \$2, respectively. There are two additional asks in the book at \$3 and \$4. Takers t_3 and t_4 submit a limit order with a bid equal to \$4.

Table 4 represents a strongly symmetric technology with four traders. The last two rows give the price at which takers t_3 and t_4 will buy the instrument for each possible queue, respectively.

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}
	p_1	p_2	t_3	t_4	p_1	p_2	t_3	t_4	p_1	p_2	t_3	t_4
	t_3	p_1	p_1	p_1	t_4	t_4	p_2	p_2	p_2	t_3	t_4	t_3
	t_4	t_3	p_2	t_3	p_2	p_1	t_4	t_3	t_4	p_1	p_1	p_2
	p_2	t_4	t_4	p_2	t_3	t_3	p_1	p_1	t_3	t_4	p_2	p_1
t_3	2	3	1	2	3	3	1	3	4	1	1	2
t_4	3	4	3	1	2	1	3	1	3	3	2	1

Table 4: strongly symmetric technology for $n = 4$.

Taking the average we obtain that takers t_3 and t_4 will pay an expected price of $\frac{26}{12}$ and $\frac{27}{12}$, respectively. So Q is not fair.

The cases considered in this secction and Sections 5 differ in an important respect. In the fomer a trader's outcome only depend on the rank of his order in the queue, whereas in the latter it depends on which orders are ranked above. Surprisingly, the techologies that ensure fairness in both cases are identical.

Theorem 1 A technology Q is strongly symmetric if, and only if it is weakly balanced.

Proof. See the Appendix. ■

7.2 Constrained v. unconstrained RSD

A balanced technology must then consist of more queues than an strongly symmetric (or weakly balanced) technology. The question is then how many more? We address now this question.

Denote by Q^{RSD} the technology corresponding to the standard (i.e., unconstrained) Random Serial Dictatorship mechanism. That is, Q^{RSD} contains $n!$ pairwise different queues. For any technology Q, denote by $\overline{Q}$ the smallest technology (i.e., the technology with the fewest number of queues) such there is exists some integer ℓ such that

$$|\{q' \in Q \ : \ q' = q\}| = \ell \times |\{q' \in \overline{Q} \ : \ q' = q\}| \qquad \text{for all } q \in Q. \tag{26}$$

For instance, if $Q = \{(i_1, i_2), (i_1, i_2), (i_2, i_1), (i_2, i_1)\}$ then $\overline{Q} = \{(i_1, i_2), (i_2, i_1)\}$. We then have the following result.

Theorem 2 A technology Q is balanced if, and only if $\overline{Q} = \overline{Q}^{RSD}$.

Proof. The *only if* part trivial. To prove the *if* part, let Q be a balanced technology and for simplicity set $\overline{Q} = Q$. Let $q \in Q$ and assume without loss of generality that

$$q = i_1, i_2, \ldots, i_{n-2}, i_{n-1}, i_n.$$

Since Q is balanced, there exists $q' \in Q$ such that $q'|_{i_n} = q|_{i_{n-1}}$. So $q' = i_1, i_2, \ldots, i_{n-2}, i_n, i_{n-1}$. Assume now that there is some k such that for any $h > k$ it holds that for any queue $\widehat{q}$ such that $\widehat{q}|_{i_h} = q|_{i_h}$ then $\widehat{q} \in Q$. That is, any queue of the form

$$i_1, i_2, \ldots, i_{h-1}, i_h, j_1, j_2, \ldots, j_{n-h}, \qquad \text{with} \quad j_1, \ldots, j_{n-h} \in \{i_h, \ldots, i_n\}$$

belongs to Q. Consider now $q^{i_k \leftrightarrow i_{k+1}}|_{i_k} = i_1, i_2, \ldots, i_{k-1}, i_{k+1}$. So we have

$$q^{i_k \leftrightarrow i_{k+1}} = i_1, i_2, \ldots, i_{k-1}, i_{k+1}, i_k, j_1, \ldots, j_{n-h} \tag{27}$$

with $j_1, \ldots, j_{n-h} \in \{i_{k+2}, \ldots, i_n\}$. Note that so far we do not know how $i_{k+2}, \ldots, i_n$ are ranked in $q^{i_k \leftrightarrow i_{k+1}}$. However, since Q is balanced, there is $\widehat{q} \in Q$ such that $\widehat{q}|_{i_k} = q^{i_k \leftrightarrow i_{k+1}}|_{i_k}$.

So, from the induction hypothesis, for any $\ell \in \{k, k+2, \ldots, n\}$ if $\widehat{q}$ is such that $\widehat{q}|_{i_\ell} = q^{i_k \leftrightarrow i_{k+1}}|_{i_\ell}$, then $\widehat{q} \in Q$. Since Q is balanced, for any such queue $\widehat{q}$ and any $\ell \in \{k, k+2, \ldots, n\}$, there is $\widetilde{q} \in Q$ such that $\widetilde{q}|_{i_{k+1}} = \widehat{q}^{\,i_\ell \leftrightarrow i_{k+1}}|_{i\ell}$. That is, $\widetilde{q}$ is of the form

$$i_1, i_2, \ldots, i_{k-1}, i_\ell, j_1, j_2, \ldots, j_{n-k+1} \quad \text{with} \quad j_1, j_2, \ldots, j_{n-k+1} \in \{i_k, i_{k+2}, \ldots, i_n\} \backslash \{i_\ell\}.$$

From the induction hypothesis any such $\widetilde{q}$ (i.e., however traders in $\{i_k, i_{k+2}, \ldots, i_n\} \backslash \{i_\ell\}$ are ranked) belongs to Q. Hence, for any $\ell \in \{k, k+1, k+2, \ldots, n\}$ and any $\widehat{q}$ such that $q|_{i_k} = \widehat{q}|_{i_\ell}$ we have $\widehat{q} \in Q$, the desired result. ■

A direct corollary of Theorem 2 is the following.

Corollary 1 A technology Q is balanced if, and only if it is fully balanced.

Proof. Clearly, if Q is fully balanced then it is balanced. Conversely, if Q is balanced then $\bar{Q} = \bar{Q}^{RSD}$. Since $\bar{Q}^{RSD}$ is obviously fully balanced then so is Q. ■

Corollary 2 Outcome fairness implies access fairness, that is, if Q is (fully) balanced then Q is strongly symmetric.

Proof. Immediate from Theorem 2 because $\bar{Q}^{RSD}$ is obviously strongly symmetric. ■

8 Market design implications

Our results provide an unambiguous message regarding the standard design of centralized financial exchanges: hardware constraints imposed by the switch technology cannot guarantee fairness.

Regarding access fairness our result crucially depends on the assumption that agents trading different securities are queued together by the same switch. One may thus deduce that access fairness could be restored if each instrument or asset would have its own switch (and thus its own matching engine). This solution would make sense from a theoretical perspective, but not in practice: it would amount to have one exchange per instrument.

In recent years several modifications to the CLOB have been proposed and/or implemented. A first proposal is the random-delay scheme by Harris (2013). In this scheme, upon its receipt by the exchange, a random delay of (nominally) 0-10 milliseconds is added to each order before it is presented to the CLOB. This random delay of course has the effect of reordering messages received by the exchange. Since the draws from the random delay distribution are independent, the ordering produced by the scheme is equivalent to RSD. What this means, notwithstanding strategic behavior where a trader sends the same order in duplicate, is that if the interval from which the delays are drawn is large enough relative to the systemic differences in jitter among participants then the scheme can restore a notion of equal treatment of equally fast participants on an exchange.[29]

[29]To illustrate what we mean by a buffer 'long enough' relative to systemic differences in an exchange's jitter across participants consider the following example adapted from Melton (2020). Imagine a scenario where there are two traders that are otherwise equally fast but where one is subject to a persistent additional

Consider next the essence of the designs proposed by Tresser and Sturman (2002), and subsequently by Schwartz and Wu (2013) and Budish et al. (2015), which treat time not as a continuous variable but as one that constitutes a discrete, fixed size interval. Orders received in the same such interval are deemed by the exchange to have been received 'at the same time'. If these orders were to be processed serially against the CLOB then our results tell us that, absent any consideration of each order's content and the prevailing best bid and offer, RSD is necessary and sufficient. The same reasoning above about the length of the interval relative to the size of systemic differences in exchange jitter among otherwise equally fast participants applies here, too.

Consider finally the scheme proposed by Melton (2014b, 2017) implemented on Refinitiv Matching for the spot foreign exchange instruments that trade on it.[30] In this scheme queues are generated *only* over orders that are actually *competing*. There is an independent buffer and associated timer for each such group of competing orders that were received at substantially the same time. To illustrate: if the offer on an instrument is \$1.00 and two buy orders are received for \$1.01 those are both put into the 'buy as taker' buffer for that instrument; but if also substantially simultaneously three bids were received with limit prices of \$0.96 those would be put into a 'buy as maker at \$0.96' buffer for that instrument; if also simultaneously further bids were received with limit prices of \$0.98 those would go into the 'buy as maker at \$0.98' buffer for the instrument. After each buffer's timer has run for 3 milliseconds—the receipt of the first order in a buffer is what starts its timer—the orders that are in it are shuffled according to a particular procedure and drained from the buffer so as to be presented to the CLOB.

What our model tells us about queues of *only* competing orders is that strong symmetry is necessary and sufficient. In practice, to defend against an advantage that otherwise might be obtained by sending the same order in duplicate, the 'first' order each trader sent is subject to RSD, then if there exist duplicates the second such orders are subject to RSD and so on. What this means is that despite at the time the scheme was designed not having

delay by the exchange of 1 millisecond. Imagine then the interval from which the random delays are drawn from is 0-4 milliseconds. In the resultant orderings we want each to have an equal chance of being first and second but the persistent delay faced by one reduces his chance of being first to $\frac{3}{4} \times \frac{3}{4} \times \frac{1}{2} = 0.28125$. If however the random delays are drawn from a longer interval, say 0-100ms, then his chance of being first is $\frac{99}{100} \times \frac{99}{100} \times \frac{1}{2} = 0.49005 \approx 0.5$.

[30] Refinitiv Matching was formerly known as Thomson Reuters Matching, and prior to that as Reuters Matching. The name changes reflect ownership changes of the company.

a formal model of fairness like the one in this paper the queues produced by the mechanism deployed on Refinitiv Matching nevertheless exhibit fairness. As for the case in our model where providers are canceling simultaneously with takers 'sniping', a policy decision was made prior to the scheme's implementation not to buffer cancel messages, and instead to process them in real-time, i.e., forward them to the CLOB immediately upon their receipt. What this means is that the RSD requirement for cancels mixing with taker orders is not relevant to this particular scheme, though it is to the CLOB generally.

Many other 'speed bumps' deployed real financial exchanges, however, do *not* restore fairness because they continue to rely on the network switch to perform serialization of at least some simultaneously received orders. Many such schemes impose a fixed-length delay on taker orders (but not cancels) to reduce 'sniping' but they will not ensure fairness among equally fast taker and/or providers. Even those schemes that impose variable delays on taker orders will not restore fairness among providers. An inventory of these speed bumps and timeline showing their date of first introduction on real financial exchanges is provided by Osipovich (2019).

Finally, our results also tell us about necessary and sufficient properties for queues produced for market data distribution transmitted under a *unicast* scheme, i.e., when market data is specific to each trader.[31] Since market data updates, which are just a contemporaneous snapshot of the CLOB, are a trading signal or 'trigger' for races among market participants it is the queue on those updates that causes the queue at time of sending for the orders it triggers among equally fast market participants. The process of generating a market data update of the CLOB by first removing counterparties a participant cannot trade with is often called *credit-screening* or *credit-filtering* (Silverman and Hoffman, 1999; Melton, 2014a; Gould et al., 2017). Unicast is thus often used in markets that tend to operate on bilateral credit like the spot foreign exchange market. The difference between multicast and unicast transmission schemes is illustrated in Figure 2.

Market data messages being sent from the exchange to market participants as shown in Figure 2. These market data messages contain a point-in-time snapshot of the CLOB and many participants have trading strategies that send order messages responsive the content in these snapshots. Consequently, if one participants receives a snapshot before another it

[31]The alternative system is multicast, where the same market data is sent to all participants. This the standard system for markets that are centrally-cleared like equity markets, because everyone can trade with everyone else in the CLOB.

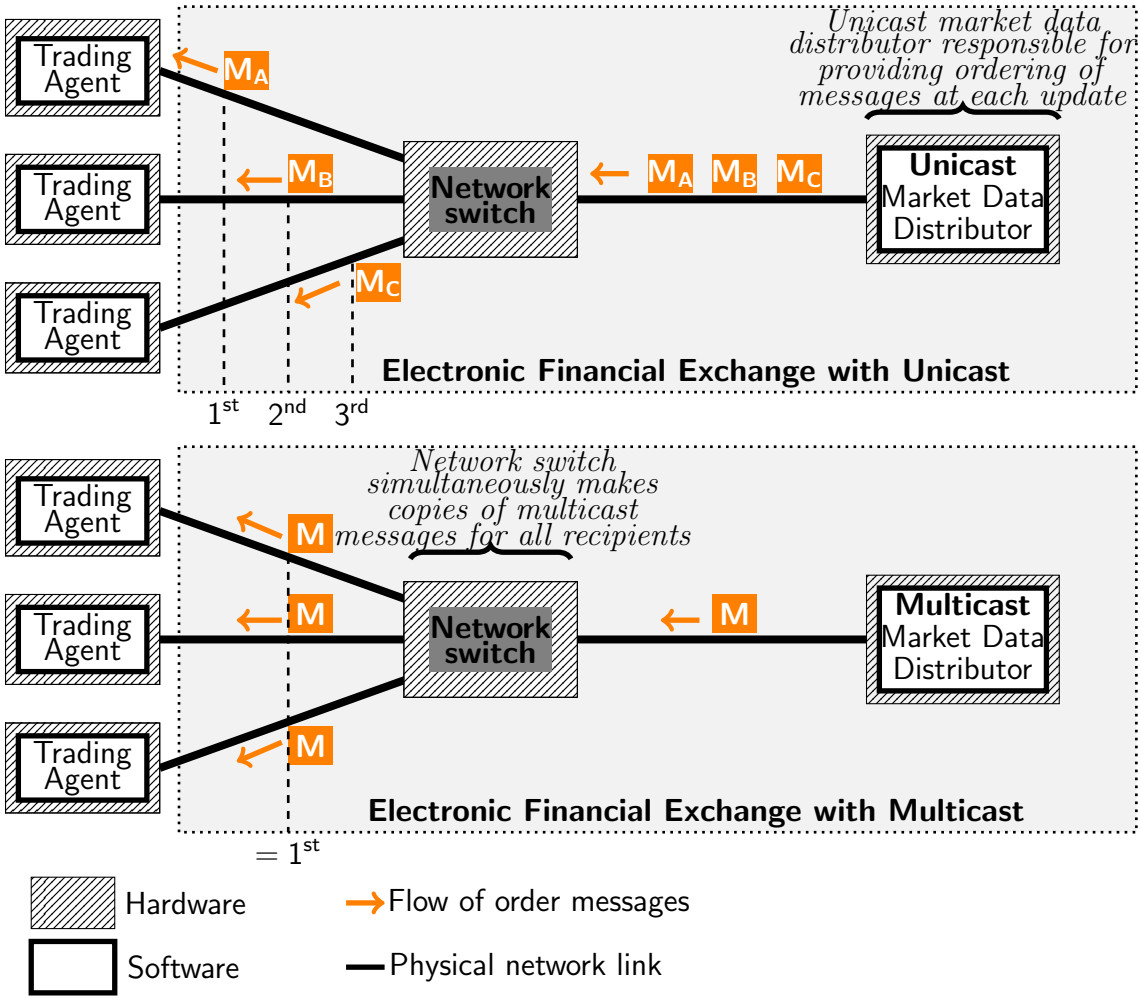

Figure 2: Unicast vs. multicast market data distribution on an electronic financial exchange.

is as if s/he has received a 'head-start' over the other even if the participants are otherwise equally fast. Since we cannot know what if any orders will be triggered by a given market data update our results tell us that a unicast market data update scheme, to result in equal-treatment-of-equally fast participants, must exhibit RSD (unless the differences from the first participant receiving an update to the last participant receiving it are 'small' relative to the response-time jitter of the participants, see Melton (2020)).

A Proof of Theorem 1

Proof of Theorem 1 Observe that if $|N| = 2$ then strong symmetry is trivially equivalent to balancedness.[32] So henceforth we assume $|N| \geq 3$.

(Only if) Let Q be a weakly balanced technology. So, for any $i, j \in N$ we have

$$|\{S \in \mathbf{U}_i(Q) \ : \ |S| = k\}| = |\{S \in \mathbf{U}_j(Q) \ : \ |S| = k\}|, \quad \text{for } k = 0, 1, \ldots n-1. \tag{28}$$

Since $|U_i(q)| = k$ implies $r_i(q) = k+1$, Eq. (28) is equivalent to

$$|\{q \in Q \ : \ r_i(q) = k+1\}| = |\{q \in Q \ : \ r_j(q) = k+1\}|, \quad \text{for } k = 0, 1, \ldots n-1. \tag{29}$$

So, Q is symmetric. We now show that Q^S is symmetric for any non-empty $S \subset N$. Note that for any $i \in S$, $\mathbf{U}_i(Q^S) = \{\widehat{S} \cap S\}_{\widehat{S} \in \mathbf{U}_i(Q)}$. Let i and j be any two traders in S, and let $\widehat{S} \in \mathbf{U}_i(Q)$. If $j \notin \widehat{S}$, then $\widehat{S} \in \mathbf{U}_j(Q)$ and thus $(\widehat{S} \cap S) \in \mathbf{U}_j(Q^S)$. If $j \in \widehat{S}$, then $(\widehat{S} \cup \{i\} \backslash \{j\}) \in \mathbf{U}_j(Q)$. Hence, $((\widehat{S} \cup \{i\} \backslash \{j\}) \cap S) \in \mathbf{U}_j(Q^S)$. Clearly, $|(\widehat{S} \cup \{i\} \backslash \{j\}) \cap S| = |\widehat{S} \cap S|$. Therefore,

$$\left|\{\widehat{S} \in \mathbf{U}_i(Q^S) \ : \ |\widehat{S}| = k\}\right| = \left|\{\widehat{S} \in \mathbf{U}_j(Q^S) \ : \ |\widehat{S}| = k\}\right|, \quad \text{for } k = 0, 1, \ldots n-1,$$

which implies that Q^S is symmetric. □

(If) Suppose now that Q is strongly symmetric. For any trader $i \in N$ and set $S \subseteq N \backslash \{i\}$, let

$$Q_{i,S} = \{q \in Q \ : U_i(q) = S\}. \tag{30}$$

Technology Q is weakly balanced if for any two traders $i, j \in N$, and any set $S \subseteq N$, $S \in \mathbf{U}_i(Q)$ implies $S \in \mathbf{U}_j(Q)$ whenever $j \notin S$, and $S \cup \{i\} \backslash \{j\} \in \mathbf{U}_j(Q)$ if $j \in S$. Hence, Q is weakly balanced if

$$|Q_{i,S}| = \begin{cases} |Q_{j,S}| & \text{if } j \notin S \\ |Q_{j,S \cup \{i\} \backslash \{j\}}| & \text{if } j \in S. \end{cases} \tag{31}$$

We show that Eq. (31) holds for any set S by induction.

To begin, note that for any $i \in N$, $Q_{i,\{\emptyset\}} = \{q \in Q \ : \ r_i(q) = 1\}$. From strong symmetry, $|\{q \in Q \ : \ r_j(q) = 1\}| = |\{q \in Q \ : \ r_i(q) = 1\}|$. So, $|Q_{i,\{\emptyset\}}| = |Q_{j,\{\emptyset\}}|$. It is convenient to distinguish between two cases, depending on whether $j \in S$.

[32] In this case we have for any $i \in N$ and $q \in Q$, $U_i(q) = \{\emptyset\}$ or $U_i(q) = N \backslash \{i\}$.

Case 1: $j \notin S$

For each subset $R \subseteq S$, $q \in Q_{i,R}$ implies $q^{N\setminus S} \in Q_{i,\emptyset}^{N\setminus S}$. Also, for $R \nsubseteq S$, $q \in Q_{i,R}$ implies $q^{N\setminus S} \notin Q_{i,\emptyset}^{N\setminus S}$. Hence,

$$|Q_{i,\emptyset}^{N\setminus S}| = \sum_{R\subseteq S} |Q_{i,R}|. \tag{32}$$

Let $S = \{k\}$, with $k \neq i, j$. We claim that $|Q_{i,\{k\}}| = |Q_{j,\{k\}}|$. From Eq. (32), $|Q_{i,\emptyset}^{N\setminus\{k\}}| = |Q_{i,\emptyset}| + |Q_{i,\{k\}}|$ and $|Q_{j,\emptyset}^{N\setminus\{k\}}| = |Q_{j,\emptyset}| + |Q_{j,\{k\}}|$. Since Q and $Q^{N\setminus\{k\}}$ are symmetric, $|Q_{i,\emptyset}^{N\setminus\{k\}}| = |Q_{j,\emptyset}^{N\setminus\{k\}}|$ and $|Q_{i,\emptyset}| = |Q_{j,\emptyset}|$. So $|Q_{i,\{k\}}| = |Q_{j,\{k\}}|$, which proves the claim.

Suppose now that for all sets $R \subset N$ such that $|R| < k$ and $i, j \notin R$, $|Q_{i,R}| = |Q_{j,R}|$. Let S such that $|S| = k$. Since $Q^{N\setminus S}$ is symmetric $|Q_{i,\emptyset}^{N\setminus S}| = |Q_{j,\emptyset}^{N\setminus S}|$ and thus from Eq. (32) we have

$$\sum_{R\subset S} |Q_{i,R}| + |Q_{i,S}| = \sum_{R\subset S} |Q_{j,R}| + |Q_{j,S}| \tag{33}$$

The induction hypothesis thus implies $|Q_{i,S}| = |Q_{j,S}|$, the desired result.

Case 2: $j \in S$

Suppose first that $S = \{j\}$. We need to show that $|Q_{j,\{i\}}| = |Q_{i,\{j\}}|$. Since Q is symmetric, $|Q_{h,\emptyset}| = |Q_{h',\emptyset}|$, for all $h, h' \in N$. Define $r = |Q_{h,\emptyset}|$.

Also, any $k \in N$, since $Q^{N\setminus\{k\}}$ is symmetric, $|Q_{h,\emptyset}^{N\setminus\{k\}}| = |Q_{h',\emptyset}^{N\setminus\{k\}}|$, for any $h, h' \in N\setminus\{k\}$. Eq. (32) thus implies $|Q_{h,\{k\}}| = |Q_{h',\{k\}}|$, for any $h, h' \in N\setminus\{k\}$. Note that $\sum_{h\neq k} |Q_{h,\{k\}}| = |Q_{k,\emptyset}| = r$. Hence,

$$|Q_{h,\{k\}}| = \frac{r}{n-1} \qquad \text{for any } h, k \in N. \tag{34}$$

Therefore, $|Q_{i,\{j\}}| = |Q_{j,\{i\}}|$.

Claim If $R, R' \subset N$ such that $|R| = |R'|$ then

$$|Q_{h,\emptyset}^{N\setminus R}| = |Q_{h',\emptyset}^{N\setminus R'}| \text{ for any } h \in N\setminus R \text{ and } h' \in N\setminus R'\,. \tag{35}$$

Proof of the Claim. Let $k_1, k_1' \in N$, and let $h \in N\setminus\{k_1\}$ and $h' \in N\setminus\{k_1'\}$. From Eq. (32),

$$|Q_{h,\emptyset}^{N\setminus\{k_1\}}| = |Q_{h,\emptyset}| + |Q_{h,\{k_1\}}| \qquad \text{and} \qquad |Q_{h',\emptyset}^{N\setminus\{k_1'\}}| = |Q_{h',\emptyset}| + |Q_{h',\{k_1'\}}|\,. \tag{36}$$

Since Q is symmetric, $|Q_{h,\emptyset}| = |Q_{h',\emptyset}|$. Also, from Eq. (34), $|Q_{h,\{k_1\}}| = \frac{r}{n-1} = |Q_{h',\{k_1'\}}|$. Hence, $|Q_{h,\emptyset}^{N\setminus\{k_1\}}| = |Q_{h',\emptyset}^{N\setminus\{k_1'\}}|$.

Define $N_1 = N\setminus\{k_1, k_1'\}$ and $Q' = Q^{N_1}$. Since Q is strongly symmetric, Q' is symmetric. Let $k_2, k_2' \in N$, and let $h \in N\setminus\{k_2\}$ and $h' \in N\setminus\{k_2'\}$. We can use the previous argument and

deduce that $|Q'^{N_1\setminus\{k_2\}}_{h,\emptyset}| = |Q'^{N_1\setminus\{k'_2\}}_{h',\emptyset}|$. Since $Q'^{N_1\setminus\{k_2\}} = Q^{N\setminus\{k_1,k_2\}}$ and $Q'^{N_1\setminus\{k'_2\}} = Q^{N\setminus\{k'_1,k'_2\}}$ we have $|Q'^{N\setminus\{k_1,k_2\}}_{h,\emptyset}| = |Q'^{N\setminus\{k'_1,k'_2\}}_{h',\emptyset}|$. Continuing this way with $\{k_3,k'_3\}$, $\{k_4,k'_4\}$, ...yields $|Q^{N\setminus\{S\}}_{h,\emptyset}| = |Q^{N\setminus S'}_{h',\emptyset}|$ for any S, S' such that $|S| = |S'|$, which proves the claim. □

Suppose now that $S = \{j,k\}$. Let $S' = S\cup\{i\}\setminus\{j\}$. Thus, from Eq. (32) and Eq. (35), we have

$$\underbrace{|Q_{i,\emptyset}|}_{=r} + \underbrace{|Q_{i,\{k\}}|}_{=\frac{r}{n-1}} + \underbrace{|Q_{i,\{j\}}|}_{=\frac{r}{n-1}} + |Q_{i,\{jk\}}| = \underbrace{|Q_{j,\emptyset}|}_{=r} + \underbrace{|Q_{j,\{k\}}|}_{=\frac{r}{n-1}} + \underbrace{|Q_{j,\{i\}}|}_{=\frac{r}{n-1}} + |Q_{j,\{ik\}}| \tag{37}$$

$$\Leftrightarrow \qquad |Q_{i,\{jk\}}| = |Q_{j,\{ik\}}|\,. \tag{38}$$

We can now show that $|Q_{i,S}| = |Q_{j,S\cup\{j\}\setminus\{i\}}$ for any set $S \subseteq N\setminus\{i\}$ such that $j \in S$. Assume that this equality holds for any set R such that $|R| < k$, and let S such that $|S| = k$. Let $S' = S\cup\{j\}\setminus\{i\}$. Since $|Q^{N\setminus S}_{i,\emptyset}| = |Q^{N\setminus S'}_{j,\emptyset}|$ we have from Eq. (32)

$$\sum_{\substack{R\subset S\\ j\notin R}} |Q_{i,R}| + \sum_{\substack{R\subset S\\ j\in R}} |Q_{i,R}| + |Q_{i,S}| = \sum_{\substack{R\subset S'\\ i\notin R}} |Q_{i,R}| + \sum_{\substack{R\subset S'\\ i\in R}} |Q_{i,R}| + |Q_{j,S'}|\,. \tag{39}$$

The induction hypothesis thus implies $|Q_{i,S}| = |Q_{j,S}|$, the desired result. From *Case 1* we have $\sum_{\substack{R\subset S\\ j\notin R}} |Q_{i,R}| = \sum_{\substack{R\subset S'\\ i\notin R}} |Q_{i,R}|$, and from the induction hypothesis we have $\sum_{\substack{R\subset S\\ j\in R}} |Q_{i,R}| = \sum_{\substack{R\subset S'\\ i\in R}} |Q_{i,R}|$. So we obtain $|Q_{i,S}| = |Q_{j,S'}|$, the desired result. ■

References

Abdulkadiroğlu, A. and Sönmez, T. (1998). Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–701.

Angel, J. J., Harris, L. E., and Spatt, C. S. (2011). Equity trading in the 21st century. *The Quarterly Journal of Finance*, 1(01):1–53.

Angel, J. J. and McCabe, D. (2013). Fairness in financial markets: The case of high frequency trading. *Journal of Business Ethics*, 112(4):585–595.

Aquilina, M., Budish, E., and O'Neill, P. (2020). Quantifying the high-frequency trading "arms race": A simple new methodology and estimates. *United Kingdom Financial Conduct Authority Occasional Paper*. `https://www.fca.org.uk/publication/occasional-papers/occasional-paper-50.pdf`.

Baldauf, M. and Mollner, J. (2019). Asymmetric speed bumps: A market design response to high-frequency trading. https://voxeu.org/article/asymmetric-speed-bumps-response-high-frequency-trading. In Vox CERP Policy Portal (online), accessed Apr 25 2020.

Baldauf, M. and Mollner, J. (2020). High-frequency trading and market performance. *Journal of Finance (forthcoming)*.

Brolley, M. and Zoican, M. (2019). Liquid speed: On-demand fast trading at distributed exchanges. *arXiv preprint arXiv:1907.10720*.

Budish, E., Cramton, P., and Shim, J. (2015). The high-frequency trading arms race: Frequent batch auctions as a market design response. *The Quarterly Journal of Economics*, 130(4):1547–1621.

CBOE (2020). CBOE system performance: World-class, sustained low latency. https://cdn.batstrading.com/resources/features/bats_exchange_Latency.pdf. Accessed May 5 2020.

CME Group (2014). Slides from new iLink architecture webinar (part i). https://web.archive.org/web/20141005062026/http://www.cmegroup.com/education/new-ilink-architecture-webinar.html.

CME Group (2020). The world's leading electronic platform: CME Globex. https://www.cmegroup.com/globex/files/globexbrochure.pdf. Accessed May 5 2020.

Dahlström, P. (2019). *New Insights on Computerized Trading: Implications of Frequently Revised Trading Decisions*. PhD thesis, Stockholm Business School, Stockholm University.

Eurex (2016). Insights into trading system dynamics: Eurex Exchange's T7. https://www.eurexchange.com/resource/blob/48918/ba7e2c5900f1069bc04f4785b15783eb/data/presentation_insights-into-trading-system-dynamics_en.pdf.

Farmer, D. and Skouras, S. (2012). Review of the benefits of a continuous market vs. randomised stop auctions and of alternative priority rules (policy options 7 and 12). *Manuscript, Foresight. Government Office for Science*.

Gould, M. D., Porter, M. A., and Howison, S. D. (2017). Quasi-centralized limit order books. *Quantitative Finance*, 17(6):831–853.

Gould, M. D., Porter, M. A., Williams, S., McDonald, M., Fenn, D. J., and Howison, S. D. (2013). Limit order books. *Quantitative Finance*, 13(11):1709–1742.

Harris, L. (2013). What to do about high-frequency trading. *Financial Analysts Journal*, 69(2).

Kluber, M. (2017). Information technology. In Francioni, R. and Schwartz, R., editors, *Equity Markets in Transition: The Value Chain, Price Discovery, Regulation, and Beyond*, chapter 7, pages 189–214. Springer International Publishing.

Lewis, M. (2014). *Flash Boys: A Wall Street Revolt*. A Wall Street Revolt. W. W. Norton.

Li, S., Wang, X., and Ye, M. (2019). Who provides liquidity, and when? Technical report, National Bureau of Economic Research.

Lohr, A. and Neusüß, S. (2019). Understanding an ultra-fast market through ultra-accurate time synchronization. `https://www.eurexchange.com/resource/blob/1567598/d42f3075b0ed50136f22b4dd5ba6283d/data/presentation_stac_summit_new_york.pdf`.

MacKenzie, D. and Pablo Pardo-Guerra, J. (2014). Insurgent capitalism: Island, bricolage and the re-making of finance. *Economy and Society*, 43(2):153–182.

Mavroudis, V. and Melton, H. (2019). Libra: Fair order-matching for electronic financial exchanges. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 156–168.

McKeown, N. (1997). A fast switched backplane for a gigabit switched router. *Business Communications Review*, 27(12):1–30.

Melton, H. (2014a). Fair credit screened market data distribution. US Patent App. 14/535,776.

Melton, H. (2014b). Ideal latency floor. US Pat. App. 14/533,543.

Melton, H. (2017). Market mechanism refinement on a continuous limit order book venue: a case study. *ACM SIGecom Exchanges*, 16(1):72–77.

Melton, H. (2018). On fairness in continuous electronic markets. In *Proceedings of the International Workshop on Software Fairness*, pages 29–31. ACM.

Melton, H. (2020). The paradoxical effects of jitter on fairness in financial exchanges: Engineering implications. In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 176–181. IEEE.

Menkveld, A. J. (2018). High-frequency trading as viewed through an electron microscope. *Financial Analysts Journal*, 74(2):24–31.

Menkveld, A. J. and Zoican, M. A. (2017). Need for speed? Exchange latency and liquidity. *The Review of Financial Studies*, 30(4):1188–1228.

Moulin, H. (2004). *Fair division and collective welfare*. MIT press.

Mounjid, O., Rosenbaum, M., and Saliba, P. (2019). From asymptotic properties of general point processes to the ranking of financial agents. *arXiv preprint arXiv:1906.05420*.

NYSE (2020). NYSE Pillar: Our integrated trading technology platform. `https://www.nyse.com/pillar#latency`. Accessed May 5 2020.

Osipovich, A. (2019). More exchanges add 'speed bumps,' defying high-frequency traders: Over a dozen financial markets are expected to have speed bumps or similar features by 2020. `https://www.wsj.com/articles/more-exchanges-add-speed-bumps-defying-high-frequency-traders-11564401611`. In *Wall Street Journal (online)*, accessed June 11 2020.

Rojas-Cessa, R. (2016). *Interconnections for Computer Communications and Packet Networks*. CRC Press.

Roth, R. (2019). Could slow be the better speed? A matter of opinion. `https://www.eurexchange.com/exchange-en/about-us/news/Could-slow-be-the-better-speed--1576216`.

Schwartz, R. A. and Wu, L. (2013). Equity trading in the fast lane: the staccato alternative. *Journal of Portfolio Management*, 39(3):3.

Securities and Exchanges Board of India (2019). Order in the matter of NSE colocation. https://www.sebi.gov.in/enforcement/orders/apr-2019/order-in-the-matter-of-nse-colocation_42880.html. Accessed May 16 2019.

Shreedhar, M. and Varghese, G. (1995). Efficient fair queueing using deficit round robin. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 231–242.

Silverman, D. L. and Hoffman, J. W. (1999). Distributed matching system for displaying a book of credit filtered bids and offers. US Patent 5,924,083.

Sivaraman, A., Subramanian, S., Alizadeh, M., Chole, S., Chuang, S.-T., Agrawal, A., Balakrishnan, H., Edsall, T., Katti, S., and McKeown, N. (2016). Programmable packet scheduling at line rate. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 44–57.

Tresser, C. and Sturman, D. (2002). Fair and scalable trading system and method. US Patent App. 09/864,015.

Wang, X. (2018). Why do stock exchanges compete on speed, and how? *Available at SSRN 3069529*.

Wuyts, G. (2012). The impact of aggressive orders in an order-driven market: a simulation approach. *The European Journal of Finance*, 18(10):1015–1038.