Online Algorithms for Market Clearing

AVRIM BLUM AND TUOMAS SANDHOLM

Carnegie Mellon University, Pittsburgh, Pennsylvania

AND

MARTIN ZINKEVICH

University of Alberta, Edmonton, Alberta, Canada

Abstract. In this article, we study the problem of online market clearing where there is one commodity in the market being bought and sold by multiple buyers and sellers whose bids arrive and expire at different times. The auctioneer is faced with an online clearing problem of deciding which buy and sell bids to match without knowing what bids will arrive in the future. For maximizing *profit*, we present a (randomized) online algorithm with a competitive ratio of $\ln(p_{\max} - p_{\min}) + 1$, when bids are in a range $[p_{\min}, p_{\max}]$, which we show is the best possible. A simpler algorithm has a ratio twice this, and can be used even if expiration times are not known. For maximizing the *number* of trades, we present a simple greedy algorithm that achieves a factor of 2 competitive ratio if no money-losing trades are allowed. We also show that if the online algorithm is allowed to *subsidize* matches—match money-losing pairs if it has already collected enough money from previous pairs to pay for them—then it can actually be 1-competitive with respect to the optimal offline algorithm that is not allowed subsidy. That is, for maximizing the number of trades, the ability to subsidize is at least as valuable as knowing the future. We also consider objectives of maximizing buy or sell volume and social welfare. We present all of these results as corollaries of theorems on online matching in an incomplete interval graph.

We also consider the issue of incentive compatibility, and develop a nearly optimal incentive-compatible algorithm for maximizing social welfare. For maximizing *profit*, we show that no incentive-compatible algorithm can achieve a sublinear competitive ratio, even if only one buy bid and one sell bid are alive at a time. However, we provide an algorithm that, under certain mild assumptions on the bids, performs nearly as well as the best fixed pair of buy and sell prices, a weaker

An extended abstract of this article appeared in *Proceedings of the 13th Annual Symposium on Discrete Algorithms (SODA)*, ACM, New York, 2002, pp. 971–980. This article contains additional results not present in the conference version. This material is based upon work supported under National Science Foundation (NSF) grants CCR-9732705, CCR-0085982, CAREER Award IRI-9703122, IIS-9800994, ITR IIS-0081246, ITR IIS-0121678, and NSF Graduate Research Fellowship. T. Sandholm was also funded by a Sloan Fellowship. Any opinion, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Sloan Foundation.

Authors' addresses: A. Blum and T. Sandholm, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3891, e-mail: {avrim; sandholm}@cs.cmu.edu; M. Zinkevich, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, T6G 2E8, e-mail: maz@cs.ualberta.ca.

but still natural performance measure. This latter result uses online learning methods, and we also show how such methods can be used to improve our "optimal" algorithms to a broader notion of optimality. Finally, we show how some of our results can be generalized to settings in which the buyers and sellers themselves have online bidding strategies, rather than just each having individual bids.

Categories and Subject Descriptors: F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms: Algorithms, Economics, Theory

Additional Key Words and Phrases: Competitive analysis, double auctions, exchanges, online algorithms

1. *Introduction*

Exchanges (aka *double auctions*)—that is, centralized markets with potentially multiple buyers and multiple sellers—are perhaps the most important institutions for conducting commerce. Well-known applications include markets for stocks, bonds, commodities, and derivatives. Several independent business-to-business exchanges have also been founded (e.g., ChemConnect). In addition, large established companies have started to form *buying consortia* such as Covisint, Exostar, and Trade-Ranger, and large-scale sourcing of transportation services has been collaboratively conducted by manufacturers and their retailer customers [Sandholm et al. 2006, page 60]. These sourcing trends mean that instead of one company buying from multiple suppliers, we have *multiple* companies buying from multiple suppliers. In other words, part of sourcing is moving toward an exchange format. Exchanges have also started to play roles in the allocation of bandwidth, as well as in resource allocation in operating systems and computational grids.

These trends have led to an increasing need for fast market clearing algorithms for exchanges. Such algorithms have been studied in the offline (batch) context [Sandholm and Suri 2002, 2003; Sandholm et al. 2002; Wurman et al. 1998a; Sandholm and Suri 2006]. Also, recent electronic commerce server prototypes such as *eMediator* [Sandholm 2002] and *AuctionBot* [Wurman et al. 1998b] have demonstrated a wide variety of new market designs, leading to the need for new clearing algorithms.

In this article, we study the ubiquitous setting where there is a market for one commodity, for example, the stock of a single company, oil, electricity, memory chips, or CPU time, and clearing decisions must be made online. For simplicity, we assume that each buy bid and each sell bid is for a single unit of the commodity (to buy or sell multiple units, a bidder could submit multiple bids[1]). In these settings, the auctioneer has to clear the market (match buy and sell bids) without knowing what the future buy/sell bids will be. The auctioneer faces the tradeoff of clearing all possible matches as they arise versus waiting for additional buy/sell

[1] This works correctly if the buyers have diminishing marginal valuations for the units (the first unit is at least as valuable as the second, etc.) and the sellers have increasing marginal valuations for the units. Otherwise, the auctioneer could allocate the bidder's $k + 1$st unit to the bidder without allocating the bidder's kth unit to the bidder, thus violating the intention of the bid by allocating a certain number of units to the bidder at a different price than the bidder intended. These restrictions seem natural—at least in the limit—as buyers get saturated and sellers run out of inventory and capacity to produce.

bids before matching. Waiting can lead to a better matching, but can also hurt because some of the existing bids might expire or get retracted as the bidders get tired of waiting.

We formalize the problem faced by the auctioneer as an online problem in which buy and sell bids arrive over time. When a bid is introduced, the auctioneer learns the bid price and expiration time (though some of the simpler algorithms will not need to know the expiration times). At any point in time, the auctioneer can match a live buy bid with a live sell bid, removing the pair from the system. We will consider the problem both from the perspective of pure online algorithms—in which our goal is to design algorithms for the auctioneer whose performance is comparable to the best achievable in hindsight for the bid sequence observed—and from the perspective of incentive-compatible mechanism design, in which we also are concerned that it be in the bidders' interests to actually bid their true valuations.

Note that while the Securities Exchange Commission imposes relatively strict rules on the matching process in securities markets like NYSE and NASDAQ [Domowitz 1993], there are large numbers of other markets on which items other than securities are being exchanged (e.g., commodities markets and business-to-business trading). In those markets, the exchange (AKA auctioneer) has significant flexibility in deciding which buy bids and sell bids to accept. In this article, we will study how well the auctioneer can do in those settings, and with what algorithms.

1.1. Basic Definitions. We will assume that all bids and valuations are integer-valued (e.g., that money cannot be split more finely than pennies) and lie in some range $[p_{\min}, p_{\max}]$.

Definition 1.1. A *temporal clearing model* consists of a set B of buy bids and a set S of sell bids. Each bid $v \in B \cup S$ has a positive price $p(v)$, is introduced at a time $t_i(v)$, and is removed at a time $t_f(v)$. A bid v is said to be alive in the interval $[t_i(v), t_f(v)]$. Two bids $v, v' \in B \cup S$ are said to be concurrent if there is some time when both are alive simultaneously.

Definition 1.2. A *legal matching* is a collection of pairs $\{(b_1, s_1), (b_2, s_2), \ldots\}$ of buy and sell bids such that b_i and s_i are concurrent.

An *offline algorithm* receives knowledge of all buy and sell bids up front. An *online algorithm* only learns of bids when they are introduced. Both types of algorithms have to produce a legal matching: that is, a buy bid and sell bid can only be matched if they are concurrent. In our basic model, if the algorithm matches buy bid b with sell bid s (buying from the seller and selling to the buyer), it receives a profit of $p(b) - p(s)$. We will use competitive analysis [Borodin and El-Yaniv 1998] to measure the quality of an online algorithm, comparing its performance on a bid sequence (in expectation, if the algorithm is randomized) to that of the optimal offline solution for the same sequence. Note that in this measure we do not worry that had we acted differently, we might have seen a better sequence of bids: the comparison is only to the optimum on the sequence observed. Formally, the competitive ratio of an algorithm is the worst-case, over all possible bid sequences, of the ratio of its performance to the optimal offline solution on that sequence. We will consider several measures of "performance", including total profit, number of trades, and social welfare (see Section 1.3).

In Section 5.2, we consider a generalization of the temporal clearing model in which bidders can change their bids over time. In this generalization, each bid can be a function of time, with only the values so far revealed to the online algorithm, and the goal of the algorithm is to be competitive with the optimal offline solution for the actual functions.

The basic competitive-analysis model implicitly treats bidders as truthful in the sense that it takes the bids as they come and compares to the offline optimum on the same bid sequence. This is a reasonable model in settings where the gaps between buy and sell bids are fairly small compared to the prices themselves, and where true valuations are difficult to quantify (e.g., say in a stock market). However, we also consider in Section 6 the more general model of incentive-compatible mechanism design, in which we assume each bidder has a private *valuation* for the commodity, and our goal is to compete with respect to the optimal offline solution for the actual *valuations* of all the bidders. In this context, if we want the bidders to bid their true valuations, the mechanism needs to make it be in their interest to do so. We assume, though we relax this for some of our results, that the start and end times for bids are not manipulable: the only control agents have is over their bid values.[2] Even this, however, provides a sharp contrast between the setting of exchanges, which we consider here, and the simpler case of auctions.

1.2. Auctions versus Exchanges.

Competitive analysis of auctions (multiple buyers submitting bids to one seller) has been conducted by a number of authors [Bagchi et al. 2001; Bar-Yossef et al. 2002; Blum et al. 2004; Goldberg et al. 2000, 2001; Lavi and Nisan 2000] in both online and offline settings. In this article, we present the first competitive analysis of *online exchanges* (where there can be multiple buyers and multiple sellers submitting bids over time). Competitive analysis of 1-shot exchanges (all bids concurrent) has been performed by Deshmukh et al. [2002]. Exchanges are a generalization of auctions—one could view an exchange as a number of overlapping auctions—and they give rise to additional issues. For example, if a seller does not accept a buy bid, some other seller might.

There are two primary differences between auctions and exchanges with respect to the design of algorithms. The first is that in addition to the basic question of what profit threshold to require per sale, for exchanges there is the additional complication of deciding which bids meeting that criterion one should in fact match (e.g., maximizing the *number* of profitable trades is trivial for an auction). So achieving an optimal competitive ratio will require performing both tasks optimally. A second more subtle distinction, however, which impacts issues of incentive-compatibility, concerns the relation between prices and profit. In the context of auctions, there must exist a fixed sales price that achieves profit within a logarithmic factor of the optimal offline solution (in fact, the algorithms of Bagchi et al. [2001], Borodin and El-Yaniv [1998], and El-Yaniv et al. [1992] work by selecting a sales price from a specific distribution, so clearly the best fixed sales price must be at least as good). However, for exchanges, there need not exist a fixed pair of buy and sell prices that achieves a logarithmic, or even sublinear competitive ratio with respect to the optimum profit. For example, if day i contains a buyer at price i and

[2] In subsequent work, Bredin and Parkes [2005] provide a characterization of a wide class of mechanisms that are incentive-compatible with respect to start and end times. However, they do not consider the competitive ratios of such methods.

seller at price $i + 1$, the optimal offline profit in n days is n, but any fixed pair of prices will produce only one match and so can make at most 1.[3] Instead, what we show must exist in hindsight is a good *profit margin per trade*, that when combined with an appropriate matching algorithm, yields a large profit.

The net result of this distinction (between profit margins and a fixed pair of prices) is a difference between what can be achieved in the pure online model and what can be achieved by incentive-compatible algorithms when optimizing for profit. Specifically, on the positive side we give an optimal, $\ln(p_{\max} - p_{\min}) + 1$-competitive, online algorithm for maximizing profit, but which is not incentive-compatible. On the negative side, we show that in fact no individually rational (it is in bidders' interests to participate) incentive-compatible direct mechanism can achieve a competitive ratio for profit that is sublinear in $p_{\max} - p_{\min}$, even in the case of just one buyer and one seller. However, on the positive side again we present an incentive-compatible algorithm using online learning methods that performs nearly as well as the best fixed pair of buy and sell prices, so long as traffic is not too bursty.

(In contrast to the linear lower bound on incentive compatible profit maximization, on the positive side, we show an incentive compatible $2\ln(p_{\max}/p_{\min})$-competitive algorithm for maximizing *social welfare*.)

1.3. OBJECTIVES CONSIDERED AND RESULTS. In this article, we consider a number of objectives. Our goal, for each objective function we consider, will be to produce online algorithms with optimal competitive ratios. Specifically, we consider the following goals:

—*Maximize Profit*. Each pair of buy and sell bids that are matched produces a profit, which in our basic model is the difference between the buy bid and the sell bid. The total profit is the sum of these differences, over all matched pairs. Offline, the matching that optimizes profit can be found via weighted bipartite matching. We present a (randomized) online algorithm with competitive ratio $\ln(p_{\max} - p_{\min}) + 1$, which we show is the best possible. A simpler algorithm has ratio twice this, and can be used even if expiration times are not known. These algorithms build on analysis of El-Yaniv et al. [1992] for the one-way-trading problem. We also show how online learning results [Blum and Burch 2000; Freund and Schapire 1996; Littlestone and Warmuth 1994] can be used to produce algorithms with even stronger guarantees—a competitive ratio approaching 1 with respect to the optimal fixed threshold on profit per trade (see Section 5.1)—if bids are not too bursty. This idea has been applied by Bar-Yossef et al. [2002] and Blum et al. [2004] to the simpler context of online auctions.

In Section 6, we consider the issue of incentive-compatible mechanism design, where we prove both a negative result and a positive result. On the negative side, we show no direct, individually rational, incentive-compatible mechanism can

[3] In Deshmukh et al. [2002], which studies competitive analysis of exchanges in a 1-shot setting, the term "competitive" is used to refer to the ratio with respect to the best *fixed pair of buy/sell prices* (technically, the best pair that produces at least two trades). That work gives a reduction converting algorithms for 1-shot auctions that are competitive with respect to the best fixed sales price into algorithms for 1-shot exchanges that are competitive with respect to the best fixed pair of prices. In contrast, the issue here is that the relationship between the best fixed price and the overall optimum solution, while only logarithmic in the context of auctions, becomes linear in the context of exchanges.

achieve a competitive ratio that is sublinear in $p_{\max} - p_{\min}$. On the positive side, however, we give an incentive-compatible algorithm that performs well with respect to the best fixed pair of buy and sell prices (low additive regret so long as bids are not too bursty) using online learning techniques.

—*Maximize Liquidity.* Liquidity maximization is important for a marketplace for several reasons. The success and reputation of an electronic marketplace is often measured in terms of liquidity, and this affects the (acquisition) value of the party that runs the marketplace. Also, liquidity attracts buyers and sellers to the marketplace; after all, they want to be able to buy and sell.

We analyze three common measures of liquidity: (1) number of trades, (2) sum of the prices of the cleared buy bids (*buy volume*), and (3) sum of the prices of the cleared sell bids (*sell volume*). Under criterion 1, the goal is to maximize the number of trades made, rather than the profit, subject to not losing money. We show that a simple greedy algorithm achieves a factor of 2 competitive ratio, if no money-losing trades are allowed. This can be viewed as a variant on the online bipartite matching problem [Karp et al. 1990]. Interestingly, we show that if the online algorithm is allowed to *subsidize* matches—match money-losing pairs if it has already collected enough money from previous pairs to pay for them—then it can be 1-competitive with respect to the optimal offline algorithm that is not allowed subsidy. That is, the ability to subsidize is at least as valuable as knowing the future. (If the offline algorithm can also subsidize, then no finite competitive ratio is possible).

For the problems of maximizing buy or sell volume, we present algorithms that achieve a competitive ratio of $2(\ln(p_{\max}/p_{\min}) + 1)$ without subsidization. We also present algorithms that achieve a competitive ratio of $\ln(p_{\max}/p_{\min}) + 1$ with subsidization with respect to the optimal offline algorithm that cannot use subsidies. This is the best possible competitive ratio for this setting.

—*Maximize Social Welfare.* This objective corresponds to maximizing the good of the buyers and sellers in aggregate. Specifically, the objective is to have the items end up in the hands of the agents who value them the most. We obtain an optimal competitive ratio, which is the fixed point of the equation $r = \ln \frac{p_{\max} - p_{\min}}{(r-1)p_{\min}}$. Using an incentive-compatible algorithm, we can achieve a competitive ratio at most twice this.

We develop all of our algorithms in a more general setting we call the *incomplete interval-graph matching problem*. In this problem, we have a number of intervals (bids), some of which overlap in time, but only *some* of those may actually be matched (because we can only match a buy to a sell, because the prices must be in the correct order, etc.). By addressing this more general setting, we are able to then produce our algorithmic results as corollaries.

1.4. SUBSEQUENT RELATED WORK. There has been substantial work on online auctions and exchanges subsequent to the initial conference publication of this work; we mention here just a few closely related results. In the case of digital-good auctions (a seller with unlimited supply) and buyers who do not linger ($t_i(v) = t_f(v)$), Blum and Hartline [2005] use online learning methods to achieve profit quickly approaching that of the best series of k sales prices. Hajiaghayi et al. [2004] develop algorithms for limited-supply auction problems that are incentive-compatible with respect to both price and timing information, and achieve good performance when bidders' valuations are drawn independently from a distribution. Hajiaghayi et al. [2005] consider the case of an auction of *reusable* goods and characterize the class of incentive-compatible online allocation rules for this setting, as well as provide auctions with good competitive ratio bounds. Bredin and Parkes [2005] build on the work of Hajiaghayi et al. [2005], providing a characterization of a wide class of mechanisms for online exchanges that are incentive-compatible with respect to both price and timing information. However, they do not consider the competitive ratios of such methods. Work has also been done on more complex matching problems with multiple related commodities [Gong 2002].

2. *An Abstraction: Online Incomplete Interval Graphs*

In this section, we introduce an abstraction of the temporal bidding problem that will be useful for producing and analyzing optimal algorithms, and may be useful for analyzing other online problems as well.

Definition 2.1. An *incomplete interval graph* is a graph $G = (V, E)$, together with two functions t_i and t_f from V to $[0, \infty)$ such that:

(1) For all $v \in V$, $t_i(v) < t_f(v)$.
(2) If $(v, v') \in E$, then $t_i(v) \leq t_f(v')$ and $t_i(v') \leq t_f(v)$.

We call $t_i(v)$ the start time of v, and $t_f(v)$ the expiration time of v. For simplicity, we assume that for all $v \neq v' \in V$, $t_i(v) \neq t_i(v')$ and $t_f(v) \neq t_f(v')$.[4]

An incomplete interval graph can be thought of as an abstraction of the temporal bidding problem where we ignore the fact that bids come in two types (buy and sell) and have prices attached to them, and instead we just imagine a black box "E" that given two bids v, v' that overlap in time, outputs an edge if they are allowed to be matched. By developing algorithms for this generalization first, we will be able to more easily solve the true problems of interest.

We now consider two problems on incomplete interval graphs: the online edge-selection problem and the online vertex-selection problem. In the online *edge-selection* problem, the online algorithm maintains a matching M. The algorithm sees a vertex v at the time it is introduced (i.e., at time $t_i(v)$). At this time, the algorithm is also told of all edges from v to other vertices which have already been introduced. The algorithm can select an edge only when both endpoints of the edge are alive. Once an edge has been selected, it can never be removed. The objective is to maximize the number of edges in the final matching, $|M|$.

In the online *vertex-selection* problem, the online algorithm maintains a set of vertices W, with the requirement that there must *exist* some perfect matching on W. At any point in time, the algorithm can choose two live vertices v and v' and add them into W so long as there exists a perfect matching on $W \cup \{v, v'\}$. Note that there need not exist an edge between v and v'. The objective is to maximize

[4] All our results can be extended to settings without this restriction, and where $t_i(v)$ may equal $t_f(v)$. This is accomplished by imposing an artificial total order on simultaneous events. Among the events that occur at any given time, bid introduction events should precede bid expiration events. The introduction events can be ordered, for example, in the order they were received, and so can the expiration events.

the size of W. So, the vertex-selection problem can be thought of as a less stringent version of the edge-selection problem in that the algorithm only needs to commit to the *endpoints* of the edges in its matching, but not the edges themselves.

It is easy to see that no deterministic online algorithm can achieve a competitive ratio less than 2 for the edge-selection problem.[5] A simple greedy algorithm achieves this ratio:

ALGORITHM 1 (GREEDY). *When a vertex is introduced, if it can be matched, match it (to any one of the vertices to which it can be matched).*

The fact that Greedy achieves a competitive ratio of 2 is essentially folklore (e.g., this fact is mentioned in passing for a related model in Karp et al. [1990]). For completeness, we give the proof here.

THEOREM 2.2. *The Greedy algorithm achieves a competitive ratio of* 2 *for the edge-selection problem.*

PROOF. Consider an edge (v, v') in the optimal matching M^*. Define v to be the vertex which is introduced first, and v' to be the vertex which is introduced second. Then the algorithm will match either v or v'. In particular, if v is not matched before v' is introduced, then we are guaranteed v' *will* be matched (either to v or some other vertex). Therefore, the number of *vertices* in the online matching M is at least the number of *edges* in M^*, which means $|M| \geq |M^*|/2$. □

For the vertex-selection problem, we show the following algorithm achieves a competitive ratio of 1. That is, it is guaranteed to find (the endpoints of) a maximum matching in G.

ALGORITHM 2. *Let W be the set of vertices selected so far by the algorithm. When a vertex v is about to expire, consider all the live unmatched vertices v', sorted by expiration time from earliest to latest. Add the first pair $\{v, v'\}$ to W such that there exists a perfect matching on $W \cup \{v, v'\}$. Otherwise, if no unmatched vertex v' has this property, allow v to expire unmatched.*

THEOREM 2.3 (MAIN THEOREM). *Algorithm* 2 *produces a set of nodes having a perfect matching M which is a maximum matching in G.*

The proof of Theorem 2.3 appears in Appendices A and B. The core of the proof is to show that if W is the set of selected vertices, then at all times the following invariants hold:

- H_1: For any expired, unmatched vertex w, there does not exist any untaken vertex w' such that there is a perfect matching on $W \cup \{w, w'\}$. (An untaken vertex is a vertex that has been introduced but not matched.)
- H_2: For any matched vertex w, there does not exist an untaken vertex w' such that there is a perfect matching on $W \cup \{w'\} - \{w\}$ and $t_f(w) > t_f(w')$.
- H_3: For any two unexpired vertices $w, w' \in W$, there exists no perfect matching on $W - \{w, w'\}$.

[5] Consider the following scenario: vertex u expires first and has edges to v_1 and v_2. Then, right after u expires, a new vertex w arrives with an edge to whichever of v_1 or v_2 the algorithm matched to u.

The first invariant says that the algorithm is complete: it lets no matchable vertex expire, and expired vertices do not later become matchable. The second and third invariants say that the algorithm is cautious. The second invariant says that the algorithm favors vertices which expire earlier over those which expire later. The third invariant states that the algorithm only matches vertices that it has to: no subset of the set of vertices chosen has a perfect matching and contains all of the expired vertices.[6]

Since an untaken vertex is a vertex which has been introduced but not matched, no untaken vertices exist at the start of the algorithm. Also, W is empty. Therefore, these invariants vacuously hold at the start.

The proof then considers the three types of events that can occur: a vertex can be introduced, a vertex can expire without being matched, or a vertex can expire and be added with some other vertex to W. We establish that if all the invariants hold before any of these events, then they hold afterwards as well. If the first invariant holds at the termination of the algorithm, then no pair could be added to the set selected. The augmenting path theorem [Berge 1957] establishes that the selected set is therefore optimal. See Appendices A and B for a full proof.

3. *Competitive Algorithms for Profit, Liquidity, and Welfare*

We now show how the results of Section 2 can be used to design optimal algorithms for the temporal bidding problem for the objectives of maximizing profit, liquidity, and social welfare. In each case, we construct a graph in which vertices are bids and edges are placed between a buy bid and a sell bid if they overlap *and meet some additional criteria* (such as having a price gap of at least some threshold θ) that will depend on the specific problem. Note that because bids are intervals, if Algorithm 2 chooses to add some pair $\{v, v'\}$ to its matching, the corresponding bids *will* overlap in time, even though there may not be a direct edge between them in the graph.

3.1. Profit Maximization.

We now return to the temporal bidding problem and show how the above results can be used to achieve an optimal competitive ratio for maximizing profit.

We can convert the profit maximization problem to an incomplete interval graph problem by choosing some θ to be the minimum profit that we will accept to match a pair. So, when translating from the temporal bidding problem to the incomplete interval matching problem, we insert an edge between a concurrent buy bid b and a sell bid s if and only if $p(b) \geq p(s) + \theta$.

The Greedy algorithm (Algorithm 1) then corresponds to the strategy: "whenever there exists a pair of bids in the system that would produce a profit at least θ, match them immediately." Algorithm 2 attempts to be more sophisticated: first of all, it waits until a bid is about to expire, and then considers the possible bids to match to in order of their expiration times. (So, unlike Greedy, this algorithm needs to know what the expiration times are.) Second, it can choose to match a pair with profit

[6] The paraphrasing of this last point is a bit more extreme than the others, but it turns out that if there exists a perfect matching on W and a perfect matching on $W' \subset W$, then there exists two vertices $v, v' \in W - W'$ such that there is a perfect matching on $W - \{v, v'\}$.

less than θ if the actual *sets* of matched buy and sell bids could have been paired differently in hindsight so as to produce a matching in which each pair yields a profit of at least θ. This is not *too* bizarre since the sum of surpluses is just the sum of buy prices minus the sum of sell prices, and so does not depend on which was matched to which.

Define $M^*(G_\theta)$ to be the maximum matching in the incomplete interval graph G_θ produced in the above manner. Then, from Theorem 2.2, the greedy edge-selection algorithm achieves a profit of at least $\frac{1}{2}\theta \cdot |M^*(G_\theta)|$. Applying Algorithm 2 achieves surplus of at least $\theta \cdot |M^*(G_\theta)|$.

The final issue is how we choose θ. If we set θ to 1, then the number of matched pairs will be large, but each one may produce little surplus. If we set θ deterministically any higher than 1, it is possible the algorithms will miss every pair, and have no surplus even when the optimal matching has surplus. Instead, one approach is to use Classify-and-Randomly-Select [Lipton and Tomkins 1994; Awerbuch et al. 1994]: choose $\theta = 2^i$ for i chosen at random from $\{0, 1, \ldots, \ell\}$ for $\ell = \lfloor \lg(p_{\max} - p_{\min}) \rfloor$. In this case, in expectation a $1/(\ell+1)$ fraction of OPT's profit is from matches made between bids whose prices differ by an amount in the range $[\theta, 2\theta]$, and therefore the expected value of $\theta \cdot |M^*(G_\theta)|$ is at least $\frac{1}{2}\mathrm{OPT}/(\ell + 1)$. We can achieve a tighter bound by choosing an exponential distribution as in El-Yaniv et al. [1992] (see also Goldman et al. [2000]). Specifically, for all $x \in [1, p_{\max} - p_{\min}]$, let

$$\Pr[\theta \leq x] = \frac{\ln(x) + 1}{\ln(p_{\max} - p_{\min}) + 1},$$

where $Pr[\theta = 1] = \frac{1}{\ln(p_{\max} - p_{\min}) + 1}$. Observe that this is a valid probability distribution. Let OPT be the surplus achieved by the optimal offline algorithm.

LEMMA 3.1. *If θ is chosen from the above distribution, then* $\mathbf{E}[\theta \cdot |M^*(G_\theta)|] \geq \frac{\mathrm{OPT}}{\ln(p_{\max} - p_{\min}) + 1}$.

COROLLARY 3.2. *The algorithm that chooses θ from the above distribution and then applies Greedy to the resulting graph achieves competitive ratio* $2(\ln(p_{\max} - p_{\min}) + 1)$. *Replacing Greedy with Algorithm* 2 *achieves competitive ratio* $\ln(p_{\max} - p_{\min}) + 1$.

In Section 4, we prove a corresponding lower bound of $\ln(p_{\max} - p_{\min}) + 1$ for this problem.

PROOF (OF LEMMA 3.1). Let us focus on a specific pair (b, s) matched by OPT. Let $R_\theta(b, s) = \theta$ if $p(b) - p(s) \geq \theta$ and $R_\theta(b, s) = 0$ otherwise. Observe that $\theta \cdot |M^*(G_\theta)| \geq \sum_{(b,s) \in \mathrm{OPT}} R_\theta(b, s)$ because the set of pairs of profit at least θ matched by OPT is a legal matching in the incomplete interval graph. So, it suffices to prove that $\mathbf{E}[R_\theta(b, s)] \geq (p(b) - p(s))/(\ln(p_{\max} - p_{\min}) + 1)$.

We do this as follows. First, for $x > 1$, $\frac{d}{dx}\Pr[\theta \leq x] = \frac{1}{x(\ln(p_{\max} - p_{\min}) + 1)}$. So,

$$\begin{aligned} \mathbf{E}[R_\theta(b, s)] &= \Pr[\theta = 1] + \int_1^{p(b)-p(s)} \frac{x\,dx}{x(\ln(p_{\max} - p_{\min}) + 1)} \\ &= \frac{p(b) - p(s)}{\ln(p_{\max} - p_{\min}) + 1}. \end{aligned}$$

□

One somewhat strange feature of Algorithm 2 is that it may recommend matching a pair of buy and sell bids that actually have negative profit. Since this cannot possibly improve total profit, we can always just ignore those recommendations (even though the algorithm will think that we matched them).

3.2. Liquidity Maximization.

In this section, we study the online maximization of the different notions of liquidity: number of trades, aggregate price of cleared sell bids, and aggregate price of cleared buy bids.

3.2.1. *Maximizing the Number of Trades.*

Suppose that instead of maximizing profit, our goal is to maximize the *number* of trades made, subject to the constraint that each matched pair have non-negative profit. This can directly be mapped into the incomplete interval graph edge-matching problem by including an edge for every pair of buy and sell bids that are allowed to be matched together. So, the greedy algorithm achieves competitive ratio of 2, which is optimal for a deterministic algorithm, as we prove in Section 4.3.

However, if the online algorithm can subsidize matches (match a pair of buy and sell bids of negative profit if it has already made enough money to pay for them) then we can use Algorithm 2, and do as well as the optimal solution in hindsight that is not allowed subsidization. Specifically, when Algorithm 2 adds a pair $\{b, s\}$ to W, we match b and s together, subsidizing if necessary. We know that we always have enough money to pay for the subsidized bids because of the property of Algorithm 2 that its set W always has a perfect matching. We are guaranteed to do as well as the best offline algorithm which is not allowed to subsidize, because the offline solution is a matching in the incomplete interval graph. Thus, the ability to subsidize is at least as powerful as knowing the future.[7]

3.2.2. *Maximizing Buy or Sell Volume.*

A different important notion of liquidity is the aggregate *size* of the trades.

Definition 3.3. Given a matching M the **buy-volume** is $\sum_{(b,s)\in M} p(b)$. The **sell-volume** is $\sum_{(b,s)\in M} p(s)$.

If we wish to maximize buy volume without subsidization, we can use an algorithm based on the greedy profit algorithm.

Algorithm 3. *Choose a buy price threshold θ at random. Specifically, for all $x \in [p_{\min}, p_{\max}]$, let*

$$\Pr[\theta \le x] = \frac{\ln(x) + 1}{\ln(p_{\max}/p_{\min}) + 1} \text{ and let}$$
$$\Pr[\theta = 1] = \frac{\ln(p_{\min}) + 1}{\ln(p_{\max}/p_{\min}) + 1}.$$

[7] If the offline algorithm is also allowed to subsidize money-losing matches, then no finite competitive ratio is possible. In particular, if the offline algorithm can use knowledge of *future* profitable matches to subsidize money-losing matches in the present, then it is clear no interesting competitive ratio can be achieved; but even if the offline algorithm can only use past profits to subsidize, then the logarithmic lower bound on profit of Section 4.1 immediately gives a corresponding lower bound for number of trades.

When a buy bid b is introduced, if $p(b) \geq \theta$, and there exists an untaken, unexpired sell bid that can be matched without subsidy, match them. When a sell bid s is introduced, if there exists an untaken, unexpired buy bid b such that $p(b) \geq \theta$ and the bids can be matched without subsidy, match them.

This algorithm achieves a competitive ratio of $2(\ln(p_{\max}/p_{\min}) + 1)$. The proof follows that of Lemma 3.1. If the online algorithm is allowed to use subsidization, then we can use Algorithm 2 as follows.

ALGORITHM 4. *Choose a buy price threshold θ at random according to the distribution in Algorithm* 3. *Convert the online problem into an incomplete interval graph as follows. For each bid b, insert a vertex with an interval* $[t_i(b), t_f(b)]$. *If a buy bid b and a sell bid s can be matched without subsidy, and $p(b) \geq \theta$, add an edge between their respective vertices.*

Run Algorithm 2 *on the constructed graph. If Algorithm* 2 *chooses a buy bid b and a sell bid s, match them. (If $p(b) < p(s)$, then this match involves a subsidy.)*

This achieves a competitive ratio of $\ln(p_{\max}/p_{\min}) + 1$ with respect to the offline algorithm which does not use subsidy. This is the best ratio that can be achieved (the proof is by threat-based analysis similar to that in Section 4.1).

Maximizing sell volume is analogous to maximizing buy volume. The best competitive ratio we know without using subsidy is $2(\ln(p_{\max}/p_{\min}) + 1)$. The best achievable with subsidy against an offline algorithm not allowed to use subsidy is $\ln(p_{\max}/p_{\min}) + 1$.

3.3. MAXIMIZING SOCIAL WELFARE. Maximizing social welfare means maximizing the sum of the valuations of the people who are left with an item, that is, matched buyers and unmatched sellers. If B' is the set of buy bids that were matched, and S' is the set of sell bids that were unmatched, then the term that we wish to maximize is:

$$\sum_{b \in B'} p(b) + \sum_{s \in S'} p(s).$$

Equivalently, if M is our matching, and S is the set of *all* sell bids, then what we wish to maximize is:

$$\sum_{(b,s) \in M} (p(b) - p(s)) + \sum_{s \in S} p(s).$$

Note that the second term cannot be affected by the algorithm (because we assume that the bids are predetermined). Furthermore, adding a constant to the offline and online algorithms' objective values can only improve the competitive ratio. Therefore, Corollary 3.2 immediately implies we can achieve a competitive ratio of $\ln(p_{\max} - p_{\min}) + 1$. However, we can in fact do quite a bit better, using the following algorithm.

ALGORITHM 5. *Let A be a given algorithm for matching on an incomplete interval graph with competitive ratio $1/\alpha$. Let r be the fixed point of the equation:*

$$r = \frac{1}{\alpha} \ln \frac{p_{\max} - p_{\min}}{(r-1)p_{\min}} \tag{1}$$

and let D be the (cumulative) distribution over $[rp_{\min}, p_{\max}]$*:*

$$D(x) = \frac{1}{r\alpha} \ln \frac{x - p_{\min}}{(r-1)p_{\min}}. \tag{2}$$

The algorithm is as follows: Draw a threshold T such that $\Pr[T \leq x] = D(x)$*, and construct an incomplete interval graph by adding an edge between any concurrent pair of buy and sell bids b and s such that* $p(b) \geq T$ *and* $p(s) \leq T$*. Then, use algorithm A to perform the matching.*

THEOREM 3.4. *Algorithm* 5 *has competitive ratio r from Equation* 1*. In particular, if Algorithm* 2 *is used as A, then the competitive ratio of Algorithm* 5 *is the fixed point of the equation:*

$$r = \ln \frac{p_{\max} - p_{\min}}{(r-1)p_{\min}} \tag{3}$$

Note that this competitive ratio is at most $\max(2, \ln(p_{\max}/p_{\min}))$.

THEOREM 3.5. *The fixed point of Equation* 3 *is the optimal competitive ratio.*

We prove Theorem 3.5 in Section 4.

PROOF (OF THEOREM 3.4). The technique is similar to Lemma 3.1. First, let us associate utilities from trades with specific sell bids. Define S to be the set of sell bids. Fix an optimal matching M^*. Let us define $R_{opt} : S \to [p_{\min}, p_{\max}]$ to be the valuation of the person who received the item from seller s in the optimal matching. If s trades with b, then $R_{opt}(s) = p(b)$. If s does not trade, then $R_{opt}(s) = p(s)$. Observe that the optimal welfare is $W_{opt}(s) = \sum_{s \in S} R_{opt}(s)$.

For an arbitrary matching M, let us use the notation $M_{b,\bar{s}}(T)$ to indicate the number of pairs in M where the buy bid is greater than or equal to T and the sell bid is less than T. Similarly, define $M_{b,s}(T)$ to be the number of pairs in M where the buy bid and sell bid are both greater than or equal to T. Define $M_s(T)$ to be the number of unmatched sell bids greater than or equal to T. Recall that α is the reciprocal of the competitive ratio of the given matching algorithm A, and so is a lower bound on the fraction of potential trades actually harnessed.

Define $N^*(T)$ to be the number of agents (buyers and sellers) that end up with an item in the optimal matching M^*, and have valuation greater than or equal to T. So,

$$N^*(T) = |\{R_{opt}(s) \geq T | s \in S\}| = M^*_{b,\bar{s}}(T) + M^*_{b,s}(T) + M^*_s(T).$$

Let M_{alg} be a matching obtained by an online algorithm. Define $N^{alg}(T)$ to be the number of agents (buyers and sellers) that end up with an item in the matching M_{alg}, and have valuation greater than or equal to T. Then,

$$N^{alg}(T) \geq \alpha M^*_{b,\bar{s}}(T) + M^*_{b,s}(T) + M^*_s(T) \geq \alpha\, N^*(T).$$

The social welfare achieved by the algorithm is at least

$$N^{alg}(T)\, T + (|S| - N^{alg}(T))p_{\min} = (N^{alg}(T))(T - p_{\min}) + |S|p_{\min}.$$

Because $T > p_{\min}$, this is an increasing function of $N^{alg}(T)$, so the algorithm achieves social welfare at least

$$(\alpha\, N^*(T))(T - p_{\min}) + |S|p_{\min}.$$

Arithmetic manipulation yields

$$N^*(T)[\alpha T + (1-\alpha)p_{\min}] + (|S| - N^*(T))p_{\min}.$$

By substitution, we get

$$|\{R_{opt}(s) \geq T | s \in S\}|[\alpha T + (1-\alpha)p_{\min}] + |\{R_{opt}(s) < T | s \in S\}|p_{\min}.$$

By defining $R_T(s) = \alpha T + (1-\alpha)p_{\min}$ when $R_{opt}(s) \geq T$ and $R_T(s) = p_{\min}$ otherwise, the algorithm is guaranteed to achieve social welfare of at least $\Sigma_{s \in S} R_T(s)$. We have divided the social welfare on a per sell bid basis. Therefore, if T is drawn according to $D(x)$, we want to prove that $\mathrm{E}[R_T(s)] \geq \frac{R_{opt}(s)}{r}$ for all s, and the problem is solved. If $R_{opt}(s) \leq rp_{\min}$, then $\mathrm{E}[R_T(s)] \geq p_{\min} \geq \frac{R_{opt}(s)}{r}$. Consider the case where $R_{opt}(s) > rp_{\min}$. If T is drawn according to $D(x)$, then since the density of T is $D'(x) = \frac{1}{r\alpha(x - p_{\min})}$ between $rp_{\min}$ and $p_{\max}$ and zero elsewhere:

$$\begin{aligned}
\mathrm{E}[R_T(s)] &= p_{\min}(1 - D(R_{opt}(s)) + \int_{x=rp_{\min}}^{R_{opt}(s)} (\alpha x + (1-\alpha)p_{\min})(D'(x))dx \\
&= p_{\min} + \int_{x=rp_{\min}}^{R_{opt}(s)} (\alpha x - \alpha p_{\min})D'(x)dx \\
&= p_{\min} + \int_{x=rp_{\min}}^{R_{opt}(s)} \frac{1}{r}dx \\
&= \frac{R_{opt}(s)}{r}.
\end{aligned}$$

Thus, for each sell bid s, $\mathrm{E}[R_T(s)] \geq R_{opt}(s)/r$, establishing the result. □

4. *Lower Bounds on Competitive Ratios*

In this section, we establish that our analysis is tight for these algorithms. Specifically, we show that no algorithm can achieve a competitive ratio lower than $\ln(p_{\max} - p_{\min}) + 1$ for the profit maximization problem, no algorithm can achieve competitive ratio lower than the fixed point of $r = \ln \frac{p_{\max} - p_{\min}}{(r-1)p_{\min}}$ for social welfare, and no *deterministic* algorithm can achieve a competitive ratio better than 2 for the trade volume maximization problem without subsidization. We also show that no randomized algorithm can achieve a competitive ratio better than 4/3 for the trade volume maximization problem without subsidization, though we believe this is a loose bound. Furthermore, on this problem it is impossible to achieve a competitive ratio better than 3/2 without taking into consideration the expiration times of the bids. Also, we prove that our greedy profit-maximizing algorithm does not achieve a competitive ratio better than $2(\ln(p_{\max} - p_{\min}) + 1)$ (i.e., our analysis of this algorithm is tight).

4.1. A THREAT-BASED LOWER BOUND FOR PROFIT MAXIMIZATION.

In this analysis, we prove a lower bound for the competitive ratio of any online algorithm for profit maximization by looking at a specific set of temporal bidding problems. We prove that even if the algorithm knows that the problem is in this set, it cannot achieve a competitive ratio better than $\ln(p_{\max} - p_{\min}) + 1$. This is very similar to the analysis of the continuous version of the one-way trading problem in El-Yaniv et al. [1992].

For this analysis, assume $p_{\min} = 0$. Consider the situation where you have a single sell bid at price 0 that lasts forever. First, there is a buy bid at price a, and then a continuous stream of increasing buy bids with each new one being introduced after the previous one has expired. The last bid occurs at some value y. Define $D(x)$ to be the probability that the sell bid has not been matched before the bid of x dollars expires. Since it is possible there is only one buy bid, if one wanted to achieve a competitive ratio of r, then $D(a) \leq 1 - \frac{1}{r}$. Also, define $Y(y)$ to be the expected profit of the algorithm. The ratio r is achieved if for all $x \in [a, p_{\max}]$, $Y(x) \geq x/r$. Observe that $Y'(x) = -D'(x)x$, because $-D'(x)$ is the probability density at x.

Observe that one wants to use no more probability mass on a bid than absolutely necessary to achieve the competitive ratio of r, because that probability mass is better used on later bids if they exist. Therefore, for an optimal algorithm, $D(a) = 1-\frac{1}{r}$, and $Y(x) = x/r$. Taking the derivative of the latter, $Y'(x) = 1/r$. Substituting, $1/r = -D'(x)x$. Manipulating, $D'(x) = -\frac{1}{rx}$. Integrating:

$$\begin{aligned} D(y) &= D(a) + \int_a^y D'(x)dx \\ &= 1 - \frac{1}{r} - \frac{1}{r}\ln\left|\frac{y}{a}\right| \end{aligned}$$

For the optimal case, we want to just run out of probability mass as y approaches $p_{\max}$. Therefore:

$$\begin{aligned} D(p_{\max}) &= 1 - \frac{1}{r} - \frac{1}{r}\ln\left|\frac{p_{\max}}{a}\right| = 0 \\ r &= \ln\frac{p_{\max}}{a} + 1 \end{aligned}$$

Thus, setting a to 1, and shifting back by $p_{\min}$, one gets a lower bound of $\ln(p_{\max} - p_{\min}) + 1$.

4.2. Greedy Profit Maximization. The following scenario shows that our analysis of the greedy profit algorithm is tight. Imagine that a buy bid for \$2 is introduced at 1:00 (and is good forever), and a sell bid for \$1 is introduced at 1:01 (that is also good forever). At 2:00, another buy bid for \$2 is introduced, which expires at 3:00. At 3:01, another sell-bid for \$1 is introduced. In this scenario, the optimal offline algorithm achieves a profit of \$2 (matching the first buy bid to the last sell bid, and the last buy bid to the first sell bid).

With a probability of $1 - \frac{1}{\ln(p_{\max}-p_{\min})+1}$, $\theta > 1$, and the greedy algorithm ignores all of the bids. Otherwise ($\theta = 1$), the greedy algorithm matches the first two bids for a profit of 1 and then cannot match the second two. Therefore, the expected reward is $\frac{1}{\ln(p_{\max}-p_{\min})+1}$ compared to an optimal of 2.

4.3. Lower Bound for Maximizing the Number of Trades. Here we establish that no deterministic algorithm can achieve a competitive ratio lower than 2 for maximizing the number of trades without subsidization. Also, no randomized algorithm can achieve a competitive ratio lower than 4/3. Furthermore, without observing the expiration times, it is impossible to achieve a competitive ratio less than 3/2.

Imagine a sell bid s^* for \$1 is introduced at 1:00 and will expire at 2:00. At 1:01, a buy bid b is introduced for \$3, and will expire at 3:00. At 1:02, a buy bid b' is introduced for \$2, and will expire at 4:00.

Now imagine there are two possible sell bids that can be introduced: either a sell bid s for \$2.50 at 2:30, or a sell bid s' for \$1.50 at 3:30. Observe that b can match s, and b' can match s'. So if s is to appear, s^* should match b', and if s' is to appear, s^* should match b. But when s^* expires, the online algorithm does not know which one of s and s' will appear. So while a randomized algorithm can guess the correct match to make with a probability of $1/2$, the deterministic algorithm must make a decision of which to take before the adversary chooses the example, and so it will choose the wrong match.

Without observing the expiration times, it is impossible achieve a competitive ratio below 3/2. Imagine that a sell bid is introduced at 9:00 AM for \$1, and a buy bid is introduced at 9:00 AM for \$2. The algorithm must then decide whether these bids should be matched. Suppose an algorithm A matches them with probability p by 10:00 AM. If $p \leq 2/3$, then they expire at 10:00 AM and no other bids are seen and the expected reward is less than 2/3 when it could have been 1. If $p > 2/3$, then the bids last all day. Moreover, there is another buy bid from 1:00 PM to 2:00 PM for \$2 and another sell bid from 3:00 PM to 4:00 PM for \$1. If the first two bids were unmatched, then one could achieve a profit of 2 whereas the expected reward of the algorithm is $(1)p + 2(1 - p) \leq 4/3$. Therefore, no algorithm has a competitive ratio better than $3/2$ on these two examples.

4.4. LOWER BOUND FOR MAXIMIZING SOCIAL WELFARE. Maximizing social welfare is a generalization of the continuous one-way trading problem [El-Yaniv et al. 1992]. In particular, imagine that one seller has value $p_{\min}$ for the item, and is alive from time 0 onward. If at time t the exchange rate is x, then there is a buy bid that is alive only at time t and has a price $p(b) = x$. Making a trade is the same as converting the currency. Leaving the seller with the good is the same as trading on the last day at the lowest possible price. Therefore, since the competitive ratio in El-Yaniv et al. [1992] is equal to the competitive ratio in Theorem 3.4, it is the case that Thereom 3.5 holds.

5. *Extensions*

In this section, we discuss two natural extensions to the basic model.

5.1. PROFIT MAXIMIZATION REVISITED: PERFORMING NEARLY AS WELL AS THE BEST THRESHOLD IN HINDSIGHT. The algorithm of Corollary 3.2 begins by picking a threshold θ from some distribution. It turns out that under certain reasonable assumptions described below, we can use online learning results to perform within a $(1 + \epsilon)$ factor of the *best* value of θ picked in hindsight, minus an additive term linear in $1/\epsilon$. This does not violate the optimality of the original algorithm: it could be that *all* thresholds perform a logarithmic factor worse than OPT. However, one can imagine that in certain natural settings, the best strategy *would* be to pick some fixed threshold, and in these cases, the modified strategy would be within a $(1 + \epsilon)$ factor of optimal.

The basic idea of this approach is to probabilistically combine all the fixed-threshold strategies using the Randomized Weighted Majority algorithm (also called Exponential-Weighting, or Hedge) of Littlestone and Warmuth [1994] and Freund and Schapire [1996], as adapted by Blum and Burch [2000] for the case of experts with internal state. In particular, at any point in time, for each threshold θ, we can calculate how well we *would* have done had we used this threshold since the beginning of time as a pair $(\mathsf{profit}_\theta, \mathsf{state}_\theta)$, where profit_θ is the profit achieved so far, and state_θ is the set of its current outstanding bids. For example, we might find that had we used a threshold of 5, we would have made \$10 and currently have unmatched live bids $\{b_1, b_2, s_1\}$. On the other hand, had we used a threshold of 1, we would have made \$14 but currently have no live unmatched bids.

In order to apply the approach of Blum and Burch [2000], we need to be able to view the states as points in a metric space of some bounded diameter D. That is, we need to imagine our algorithm can move from $\mathsf{state}_{\theta_1}$ to $\mathsf{state}_{\theta_2}$ at some cost $d(\mathsf{state}_{\theta_1}, \mathsf{state}_{\theta_2}) \leq D$. To do this, we make the following assumption:

ASSUMPTION 1. *There is some a priori upper bound B on the number of bids alive at any one time.*

We now claim that under this assumption we can view the states as belonging to a metric space of diameter $D \leq B(p_{\max} - p_{\min})$. Specifically, suppose we are currently following some base-algorithm of threshold θ. Let us define a potential function equal to the number of bids in the state state_θ of the algorithm we are currently following, but that are not actually in our state (we do not actually have available to us as live bids) because we already matched them at some point in the past. Note that by Assumption 1, this potential function is bounded between 0 and B. Now, suppose the algorithm we are following tells us to match a given pair of bids. If we do not have both of them in our current state, then we simply do not perform a match: this causes us to possibly make $p_{\max} - p_{\min}$ less profit than that algorithm, but decreases our potential by at least 1. Since we never match a pair of bids unless told to by our current base-algorithm, this potential cannot increase while we are following it (to be clear, even if our current state has two bids that differ in price by more than θ, we do not match them unless our base-algorithm tells us to). Now, if the overall "master" algorithm tells us to switch from threshold θ to some other threshold θ', the potential can increase by at most B. Therefore, our overall total profit is at most $B(p_{\max} - p_{\min})$ less per switch than what our profit should be according to the master algorithm.

We can now plug in the Randomized Weighted-Majority (Hedge) algorithm, using the fact that there are at most $N = p_{\max} - p_{\min}$ possible different thresholds to use, to get the following theorem.

THEOREM 5.1. *Under the assumption above, for any $\epsilon > 0$ (ϵ is given to the algorithm), we can achieve an expected gain at least*

$$\max_\theta \left[(1-\epsilon)\mathsf{profit}_\theta - \frac{2B(p_{\max} - p_{\min})}{\epsilon} \ln(p_{\max} - p_{\min}) \right],$$

where profit_θ is the profit of the algorithm of Corollary 3.2 *using threshold θ.*

The proof follows the exact same lines as Blum and Burch [2000] (which builds on Littlestone and Warmuth [1994] and Freund and Schapire [1996]). Technically, the analysis of Blum and Burch [2000] is given in terms of losses rather than gains (the goal is to have an expected loss only slightly worse than the loss of the best expert). For completeness, we give the proof of Theorem 5.1 from first principles in Appendix C.

5.2. More General Bidding Languages. So far, in this article, we have required bidders to submit bids in a somewhat restrictive form: a fixed price for some interval of time. We now consider a model in which each bid can be an arbitrary *function* from time to price. For instance, a buyer might want to put in a bid for \$100 at time t_1, and then if that bid has not been matched by time t_2 to raise it to \$120, or perhaps to slowly raise it according to some schedule until time t_3 at which point if not yet matched the buyer gives up (setting his price to \$0). We assume the *online algorithm* only gets to observe the past history of the bid function, and that if a bidder is matched then his bid function is truncated and no longer observable. Our goal is to compete against the offline optimal for the entire bid functions. For instance, in the above example, an online algorithm might match the buyer at price \$100, and the offline optimal might involve waiting until the price reached \$120, even though the online algorithm never got to see the \$120 bid price. Thus, in this setting, even the online algorithm cannot necessarily compute what would have been the offline optimal in hindsight.

We show that a competitive ratio of $2(\ln(p_{\max} - p_{\min}) + 1)$ for profit maximization can still be achieved in this setting by using the Greedy algorithm described earlier in this article.

The nature of the online graph is different in this model. We can still represent the bids as vertices and the matches as edges. If there is a single agent interested in a single unit, only one vertex will be used to represent the agent's bidding function. This will guarantee that the agent does not buy or sell more than one item. If there is, at some time, a buy bid at price $b(t)$ and a sell bid at price $s(t)$ for $b(t) \geq s(t) + \theta$, then an edge from b to s is inserted into the graph. Note that two vertices may enter without having an edge, and at a later time an edge may appear between them.

The algorithm remains the same: we choose a random profit threshold θ according to the same distribution as before and only include edges representing trades that achieve profit level at least θ. The algorithm selects from the edges greedily as they appear. This is still guaranteed to get a matching at least half as large as optimal. To see this, observe that at least one vertex from every edge in the optimal matching is selected by the algorithm. When an edge arrives, either one of the vertices has already been taken, or both are free. If both are free, they will be matched to each other, or at least one of them will be matched to another edge that happened to be introduced simultaneously. Thus, after an edge is introduced, at least one of its vertices has been selected.

This technique also works for maximizing liquidity. However, maximizing social welfare in this setting is a bit trickier. One problem is that it is not totally clear how social welfare should be defined when the bids of the participants change with time. If only the *buyers'* bids change with time, then one natural definition of social welfare for a matching M is

$$\sum_{b \in B'} p_M(b) + \sum_{s \in S'} p(s),$$

where B' is the set of matched buy bids, S' is the set of unmatched sell bids, and $p_M(b)$ is the value of b's bid at the time the match was made. In that case, the Greedy algorithm still achieves its near-optimal competitive ratio. This is perhaps surprising since the online algorithm is at an extra disadvantage, in that even if it magically knew which buyers and sellers to match, it still would need to decide exactly *when* to make those matches in order to maximize social welfare.

6. *Strategic Agents*

Up until this point of the article, we have considered the standard competitive analysis model in which we compare performance of an online algorithm to the optimal offline solution for the same sequence of bids (or bidding functions). We now switch to a model in which each bidder instead has a private *valuation* for the item, and we are competing against the offline optimal for those *valuations* (i.e., the optimal had we known the true valuations and time-intervals in advance). In particular, this means that if we want bidders to actually bid their true valuations, then our mechanism must be (dominant-strategy) *incentive-compatible*. That is, no matter how other bidders behave, it should be in a bidder's best interest to report truthfully.

In this setting, we show the following results. For social welfare maximization, we present an incentive-compatible algorithm with competitive ratio at most $2\max(\ln(p_{\max}/p_{\min}), 2)$. For profit maximization we present a lower-bound, showing that no incentive-compatible, individually rational direct mechanism can achieve a competitive ratio sublinear in $p_{\max} - p_{\min}$, even when only two bids are alive at any point in time. (This lower bound actually holds even if incentive compatibility and individual rationality are only required in expectation over other agents' actions.) We then complement this with two algorithmic results: a very simple incentive-compatible algorithm with competitive ratio $p_{\max} - p_{\min}$, and a more sophisticated algorithm with a ratio of $1 + \epsilon$ (plus an additive loss term) with respect to the best *fixed* pair of buy and sell prices, when bids are not too bursty. For these latter results we assume that the start and end times of bids are not manipulable—the only control a bidder has is over his bid value.

6.1. AN INCENTIVE-COMPATIBLE MECHANISM FOR MAXIMIZING SOCIAL WELFARE. We design an algorithm that is incentive-compatible in dominant strategies for maximizing social welfare. That is, each agent's best strategy is to bid truthfully regardless of how others bid. Bidding truthfully means revealing one's true valuation.

We assume that each bidder wants to buy or sell exactly one item. We also assume that there is one window of time for each agent during which the agent's valuation is valid, and that inside the window this valuation is constant (outside this window, a seller's valuation is infinite, and a buyer's valuation is negative infinity). For example, the window for a seller might begin when the seller acquires the item, and ends when the seller no longer has storage space. A buyer's window might begin when the buyer acquires the space to store the item. Finally, we assume bidders are identifiable and therefore we can restrict each to making at most one bid.

Let us first consider a simpler setting with no temporal aspects. That is, everyone makes bids upfront to buy or sell the item. In our mechanism, we first decide a price T at which all trades will occur, if at all. We then examine the bids and only consider sell bids below T and buy bids above T. If there are more sell bids than buy bids, then for each buy bid, we select a random seller with a valid bid to match it to. We do the reverse if there are more buy bids than sell bids.

Now, there are three factors that affect the expected reward of an agent:

(1) The agent's valuation for the item. This is fixed before the mechanism begins, and cannot be modified by changing the strategy.
(2) The price of a possible exchange. This is set upfront, and cannot be modified by changing the strategy.
(3) The probability that the agent is involved in an exchange.

The last factor is the only one that the agent has control over. For a buyer, if its valuation is below the price of an exchange, then the buyer does not wish to exchange, and can submit one bid at its true valuation, and it will have zero probability of being matched.

The more interesting case is when the agent's valuation is above the trading price. Now, the agent wishes to maximize the probability of getting to trade. If the other agents submit at least one fewer buy bid above T than the number of sell bids below T, then the agent is guaranteed a trade. On the other hand, if there are as many or more buy bids than sell bids, then the agent would not be *guaranteed* to purchase an item. Regardless of what bid the buyer submits above T, it will only have as much of a chance as any other buyer. So, the buyer loses nothing by submitting a bid at its true valuation.

A similar argument applies to sellers. Therefore, all of the agents are motivated to bid truthfully.[8] An important aspect of this is that the trading price does not depend on the bids.

Now, let us apply this technique to the online problem.

ALGORITHM 6. *Define A to be a greedy algorithm that matches incoming bids uniformly at random to existing bids. Run Algorithm* 5 *with algorithm A.*

Under this algorithm, each agent is motivated to bid its true price. Furthermore, no agent can benefit from only bidding for a fraction of its window.

This algorithm is an instance of our Greedy algorithm, because it always matches two bids when it can. For each pair in the optimal matching (where the buy bid is above the price threshold and sell bid is below the threshold), the above algorithm will select at least one bid. If the first bid introduced is not there when the second bid arrives, then it was matched. If the first bid still remains when the second bid arrives, then (since either all buy bids or all sell bids are selected), one bid or the other bid is selected from the pair. This implies that our competitive ratio for the Greedy social welfare maximizing algorithm applies.

6.2. INCENTIVE-COMPATIBILITY FOR PROFIT MAXIMIZATION.

As mentioned in the introduction (Section 1.2), the landscape of what can be achieved via incentive-compatible mechanisms for profit maximization is more complicated for exchanges

[8] This scheme is vulnerable to coalitions: if for a buyer b and a seller s, $p(b) > p(s) > T$, then b has a motivation to pay s to decrease his offered price to below T, such that b can receive the item. If we assume there are no side payments, such problems do not occur.

than for 1-sided auctions. The key issue is perhaps best understood by considering the following three classes of offline competitor algorithms, from strongest to weakest:

(1) The optimal algorithm that knows all bids in advance and can produce any legal matching $\{b_1, s_1), \ldots, (b_n, s_n)\}$, extracting full profit from them $\sum_i (b_i - s_i)$. This is what we have meant throughout this article by the *offline optimal* solution OPT, and our algorithm of Section 3.1 achieved an optimal $\ln(p_{\max} - p_{\min}) + 1$ competitive ratio with respect to this "gold standard".

(2) The optimal *fixed profit threshold*. This is the optimal algorithm that picks some profit threshold ("markup") θ and can then produce any legal matching $\{(b_1, s_1), \ldots, (b_n, s_n)\}$ such that $b_i - s_i \geq \theta$, making θ profit per trade. The analysis of Section 3.1 implies that the optimal fixed profit threshold is at most a $\ln(p_{\max} - p_{\min}) + 1$ factor worse than OPT, and our algorithm of Section 5.1 performs within a $1 + \epsilon$ factor of the optimal fixed profit threshold under Assumption 1 (bids are not too bursty).

(3) The optimal *fixed pair of prices*. This is the optimal algorithm that picks two thresholds θ_b and θ_s, and can produce any legal matching $\{(b_1, s_1), \ldots, (b_n, s_n)\}$ such that $b_i \geq \theta_b$ and $s_i \leq \theta_s$ for all i, making $\theta_b - \theta_s$ profit per trade. As pointed out by the example in Section 1.2, the optimal fixed pair of prices can be much worse, by a $p_{\max} - p_{\min}$ factor, than OPT or even the optimal fixed profit threshold.

In the case of 1-sided auctions, the second two benchmarks above collapse, and for that reason it is possible to be incentive-compatible and have a logarithmic competitive ratio. In contrast, we show for exchanges that no individually-rational (it is in bidders' interests to participate) incentive-compatible direct mechanism for exchanges can achieve a competitive ratio sublinear in $p_{\max} - p_{\min}$. It is easy to achieve a linear ratio by choosing a pair of buy and sell prices from an appropriate distribution (see Section 6.2.1 below), so this is tight. In addition, we give an algorithm using online learning methods that performs nearly as well as the *best fixed pair of buy and sell prices* (benchmark (3) above), so long as traffic is not too bursty. We begin with the positive algorithmic results, and then give the lower bound, followed by a discussion of some related issues.

In this section, we assume a somewhat more restrictive model than in Section 6.1. In particular, we assume that bidders' start and end times are not manipulable, and the only control a bidder has is over his bid value.

6.2.1. *Positive Results for Profit I: Achieving a Linear Ratio.* We begin with a very simple incentive-compatible algorithm that achieves a linear competitive ratio for maximizing profit.

THEOREM 6.1. *Suppose we create an incomplete interval graph by using thresholds* $\theta_s = j$ *and* $\theta_b = j + 1$ *(placing an edge between b and s if* $p(b) \geq \theta_b$ *and* $p(s) \leq \theta_s$*) for j chosen uniformly at random from* $\{p_{\min}, p_{\min} + 1, \ldots, p_{\max} - 1\}$. *We then run Algorithm* 2, *making* \$1 *per match by buying from the seller at* θ_s *and selling to the buyer at* θ_b. *Then, this algorithm achieves an expected profit at least* $\mathrm{OPT}/(p_{\max} - p_{\min})$.

PROOF. Consider the offline optimal matching $\{(b_1, s_1), \ldots, (b_n, s_n)\}$, with its profit $\mathrm{OPT} = \sum_i (b_i - s_i)$. We can rewrite this sum (pictorially, considering it "horizontally" instead of "vertically") as

$$\text{OPT} = \sum_{j=p_{\min}}^{p_{\max}-1} |\{i : s_i \leq j \text{ and } b_i \geq j+1\}|.$$

Therefore, if we pick j at random from $\{p_{\min}, p_{\min}+1, \ldots, p_{\max}-1\}$ and choose $\theta_s = j$, and $\theta_b = j+1$, the expected profit of this matching using those posted prices is $\text{OPT}/(p_{\max} - p_{\min})$. Therefore, the expected profit of Algorithm 2 using these posted prices is at least as large. □

6.2.2. *Positive Results for Profit II: Competing with the Best Pair of Prices.* We now show how online learning methods can be used to develop an incentive-compatible algorithm for profit maximization that performs nearly as well as the best fixed pair of buy/sell prices in hindsight, when the sequence of bids is not too bursty. In particular, we make the same assumption used in Section 5.1 (Assumption 1) that at most B bids are alive at any one time, and in addition we assume that all events happen at integral time steps. Our conclusion, however, will be weaker in that our additive term will grow, sublinearly, with time (though we show how to remove this with the additional assumption that all bids have at most constant length). Specifically, if we define $\text{OPT}_{\textit{fixedpair}}$ to be the optimal profit achievable by posting some fixed pair (θ_b, θ_s) of buy and sell prices, and matching bids that come in on their respective sides of these prices, making $\theta_b - \theta_s$ profit per trade, then we achieve a profit of $\text{OPT}_{\textit{fixedpair}} - O(BT^{2/3}(p_{\max} - p_{\min}) \log^{1/3}(p_{\max} - p_{\min}))$. Thus, for our conclusion to be non-vacuous, we need that $\text{OPT}_{\textit{fixedpair}}$ grows at least at the rate of $T^{2/3}$. For example, if the optimal fixed-price profit grows linearly with time (a natural assumption in any reasonably-stationary market), then our competitive ratio with respect to the best fixed pair of prices approaches 1. In game-theoretic terminology, our algorithm has sublinear additive regret with respect to this comparison class.

The algorithm is similar to that used in Section 5.1, in that we will switch between base-algorithms using a probability distribution given by an online learning method. However, there are two primary differences:

(1) We will need to use base-algorithms (experts) that themselves are incentive-compatible. In particular, the base-algorithms we combine will each be of the same type as $\text{OPT}_{\textit{fixedpair}}$: each is defined by a pair (θ_b, θ_s) of posted buy and sell prices, and will match pairs (b, s) such that $p(b) \geq \theta_b$ and $p(s) \leq \theta_s$ using Algorithm 2, making $\theta_b - \theta_s$ profit per trade (buying from the seller at θ_s and selling to the buyer at θ_b). Since our model allows bidders control only over their prices (and not start/end times), these algorithms are incentive-compatible. In particular, Algorithm 2 does not use the actual bid values to determine which matches to make, only the start and end times.

(2) We need to ensure that our method of combining base-algorithms retains incentive compatibility. Note that it is not enough to ensure that only expired bids influence our decision to switch between base-algorithms, since a bidder may decide to bid more aggressively than his true valuation in the hope that some *other* bidders will cause variation in our prices. In fact we will address this issue through a somewhat draconian method: we define in advance a sequence

of points in time $t_0, t_1, t_2, \ldots$ such that switches can occur only at those times, and throw out *any* bid whose time-interval contains any of those t_i.

Note that one nice feature of the draconian method used in (2) above is that, after accounting for lost profit due to bids thrown out, there are no longer any costs for switching between base-algorithms, as we had in Section 5.1. However, the penalty for this is an additive term that grows with time due to the presence of the t_i: for example, if *every* bid overlapped some t_i, then our algorithm would make no profit at all (on the other hand, the t_i will be spaced out farther and farther apart, so this can only happen if the rate of increase in $\mathrm{OPT}_{\mathit{fixedpair}}$ with time drops to zero.

THEOREM 6.2. *There is an incentive-compatible algorithm for profit maximization such that in T time steps,*

$$\mathbf{E}[\mathit{profit}] \;\geq\; \mathrm{OPT}_{\mathit{fixedpair}} - O\left(B(p_{\max} - p_{\min})T^{2/3}\log^{1/3}(p_{\max} - p_{\min})\right)$$

under the assumption that at most B bids can be alive at any point in time and all bid intervals have integral endpoints.

PROOF. The algorithm is as follows: For simplicity, we assume we are given up front the number of time steps T that our exchange will be active, though this assumption can easily be removed by guessing a value and then doubling it when that guess is exceeded as in Cesa-Bianchi et al. [1997]. We now break up the T time steps into s time segments of length T/s each, so that $t_i = iT/s$. By the assumption that at most B bids can be alive at any one time, the maximum possible gain in a single time segment is $M = B(p_{\max} - p_{\min})T/s$. We now run the standard Randomized Weighted Majority algorithm with $N = (p_{\max} - p_{\min})^2$ experts, each as defined in item (1) above, having one expert for every possible pair of buy and sell thresholds. We view the entire interval $[t_{i-1}, t_i]$ as the ith trial, with each expert's gain per trial bounded in $[0, M]$. Note that due to ignoring bids that cross a time-interval, $\mathrm{OPT}_{\mathit{fixedpair}}$ may be larger than the gain of the best expert by up to $B(p_{\max} - p_{\min})s$. Putting this all together, for any $\epsilon > 0$ given as a parameter to the learning algorithm, by [Littlestone and Warmuth 1994; Freund and Schapire 1996] we can achieve a total expected gain at least:

$$(1-\epsilon)(\mathrm{OPT}_{\mathit{fixedpair}} - B(p_{\max} - p_{\min})s) - O\left(\frac{1}{\epsilon}\frac{T}{s}B(p_{\max} - p_{\min})\log(p_{\max} - p_{\min})\right)$$

$$\geq\; \mathrm{OPT}_{\mathit{fixedpair}} - O\left(\left(\epsilon T - s - \frac{1}{\epsilon}\frac{T}{s}\log(p_{\max} - p_{\min})\right)B(p_{\max} - p_{\min})\right).$$

Now, plugging in $s = T^{2/3}\log^{1/3}(p_{\max} - p_{\min})$ and $\epsilon = T^{-1/3}\log^{1/3}(p_{\max} - p_{\min})$, we get

$$\mathbf{E}[\text{online profit}] \;\geq\; \mathrm{OPT}_{\mathit{fixedpair}} - O\left(B(p_{\max} - p_{\min})T^{2/3}\log^{1/3}(p_{\max} - p_{\min})\right)$$

as desired. □

The above result requires an additive term that increases with time. If we make the additional assumption that all bid intervals have some maximum length ℓ, we can remove this dependence on time and replace it with a dependence on ℓ. Specifically, rather than have each time segment be a fixed length T/s as in the above algorithm, we can choose each segment to be a random length chosen uniformly from $[2\ell/\epsilon, 4\ell/\epsilon]$. Notice that for any pair of overlapping bids (of total length at most 2ℓ) there is at most an ϵ chance that either bid intersects an interval boundary. Thus, in expectation, at most an ϵ fraction of the profit of $\mathrm{OPT}_{\mathit{fixedpair}}$ is lost due to throwing out such bids, and so applying the same weighted-majority bounds used above we achieve an expected gain at least:

$$(1-\epsilon)^2\mathrm{OPT}_{\mathit{fixedpair}} - O\left(\frac{1}{\epsilon}\frac{\ell}{\epsilon}B(p_{\max}-p_{\min})\log(p_{\max}-p_{\min})\right).$$

Dividing ϵ by 2, we therefore have the following:

THEOREM 6.3. *There is an incentive-compatible algorithm for profit maximization such that for any given $\epsilon \in (0,1)$,*

$$\mathbf{E}[\mathit{profit}] \geq (1-\epsilon)\mathrm{OPT}_{\mathit{fixedpair}} - O\left(\frac{\ell B(p_{\max}-p_{\min})}{\epsilon^2}\log(p_{\max}-p_{\min})\right),$$

under the assumption that at most B bids can be alive at any point in time and all bid intervals have length at most ℓ.

6.2.3. *Negative Results for Maximizing Profit.* In this section, we prove that no incentive compatible, individually rational, direct mechanism can obtain a competitive ratio for profit that is sublinear in $p_{\max} - p_{\min}$. Furthermore, this holds even in the case where there is just one buy bid and one sell bid, both introduced at time 0 and both expiring at time 1. This lower-bound can then be extended to cases with arbitrarily-large values of OPT, by repeating it over time (with bids extending from time 2 to time 3, time 4 to time 5, etc.), so it is not just an issue of an additive constant.

Myerson and Satterthwaite [1983] find an optimal mechanism for maximizing the broker's revenue in a 1-buyer 1-seller setting when the valuation distributions are common knowledge. Many of the techniques they use are applicable here, but we will apply them for understanding how much utility a broker can obtain if the distributions are unknown.

Definition 6.4. In general, a mechanism is an incentive compatible, individually rational, direct mechanism if:

(1) *Direct*. The agents communicate their types (i.e., value of the item, introduction and expiration time) to the mechanism.

(2) *Incentive Compatible*. Each agent is motivated to reveal her true type: that is, she cannot increase her expected utility by stating any other type, assuming the other agents reveal their true types. (This form of incentive compatibility, aka *Bayes-Nash incentive compatibility*, is a weaker requirement than the dominant-strategy incentive compatibility used elsewhere in this article. This makes the impossibility result below stronger.)

(3) *Individually Rational*. Each agent is motivated to participate in the mechanism: that is, averaged over the other agents' true types, her expected utility is non-negative. (This form of individually rationality is a weaker requirement than elsewhere in this article, where we required that an agent is best off participating for sure. This makes the impossibility result below stronger.)

THEOREM 6.5. *It is not possible to achieve a competitive ratio less than $\frac{\lfloor p_{\max} \rfloor + 1}{3}$ with a direct, incentive-compatible, individually rational mechanism for maximizing profit, even with a single buyer and single seller with public introduction and expiration times.*

Incentive compatibility and individual rationality are difficult to characterize in general in this domain. However, we will simplify matters by considering the case that there is a single buy bid and a single sell bid, both introduced at time 0 and expiring at time 1. Moreover, we will also focus on the case where the buyer and the seller know each other's value. We then compare ourselves to the broker who knows the value of the buyer and the seller.

Formally, we can now define a mechanism with three functions:

(1) $p(\theta_1, \theta_2)$, the probability of a trade if the seller reveals type θ_1 and the buyer reveals type θ_2,

(2) $x_1(\theta_1, \theta_2)$, the expected amount paid to the seller, and

(3) $x_2(\theta_1, \theta_2)$, the expected amount charged to the buyer.

Observe that $x_1(\theta_1, \theta_2)$ is *not* defined as the expected amount given to the seller conditional on a trade being made, but the expected amount conditional on the buyer being type θ_1 and the seller being type θ_2. Similarly, $x_2(\theta_1, \theta_2)$ is the expected amount charged the buyer given only θ_1 and θ_2.

We can now specify what it means to be incentive compatible and individually rational for one buyer and one seller given x_1, x_2, and p. To make the statements clearer, define

$$U_1(\theta_1, \theta_2) = x_1(\theta_1, \theta_2) - \theta_1 p(\theta_1, \theta_2), \quad U_2(\theta_1, \theta_2) = \theta_2 p(\theta_1, \theta_2) - x_2(\theta_1, \theta_2) \tag{4}$$

to be the expected utility of the seller and buyer, respectively, given that their respective types are θ_1 and θ_2 and that both agents reveal their types truthfully.

(1) The mechanism is incentive compatible if for all $\theta_1, \theta_2, \theta_1', \theta_2' \in [p_{\min}, p_{\max}]$:

$$U_1(\theta_1, \theta_2) \geq x_1(\theta_1', \theta_2) - \theta_1 p(\theta_1', \theta_2), \quad U_2(\theta_1, \theta_2) \geq \theta_2 p(\theta_1, \theta_2') - x_2(\theta_1, \theta_2'). \tag{5}$$

(2) The mechanism is individually rational if for all $\theta_1, \theta_2 \in [p_{\min}, p_{\max}]$:

$$U_1(\theta_1, \theta_2) \geq 0, \quad U_2(\theta_1, \theta_2) \geq 0. \tag{6}$$

Finally, the expected profit of the broker is:

$$\begin{aligned} U_0(\theta_1, \theta_2) &= x_2(\theta_1, \theta_2) - x_1(\theta_1, \theta_2) \\ &= (\theta_2 - \theta_1) p(\theta_1, \theta_2) - U_2(\theta_1, \theta_2) - U_1(\theta_1, \theta_2) \end{aligned} \tag{7}$$

THEOREM 6.6. *If an online direct mechanism is incentive compatible, then for all θ_1,θ_2, $p(\theta_1, \theta_2)$ is weakly decreasing in θ_1 and weakly increasing in θ_2, and:*

$$U_1(\theta_1, \theta_2) = U_1(p_{\max}, \theta_2) + \int_{\theta_1}^{p_{\max}} p(s_1, \theta_2) ds_1, \ U_2(\theta_1, \theta_2) = U_2(\theta_1, 0) + \int_{p_{\min}}^{\theta_2} p(\theta_1, s_2) ds_2.$$

The proof of Theorem 6.6 follows quite closely the beginning of Theorem 1 of Hagerty and Rogerson [1987] who prove a similar result for budget-balanced auctions. As is shown by Myerson and Satterthwaite [1983], extending such a result beyond the budget-balanced case is not hard. See Chapter 3 of Zinkevich [2004] for a full version of the proof.

Without loss of generality, assume $p_{\min} = 0$. Define $n = \lfloor p_{\max} \rfloor$ (or more generally, $\lfloor p_{\max} - p_{\min} \rfloor$). Consider the n pairs of types for the seller and the buyer, $\{(0, 1), (1, 2), \ldots, (n-1, n)\}$. Observe that the optimal broker achieves a value of 1 in any of these scenarios. If p is the probability of trade for the online algorithm, define

$$p_1 = \min_{i \in \{1 \ldots n\}} p(i-1, i).$$

Notice that by Eq. (7), the competitive ratio of the online algorithm can be no more than $\frac{1}{p_1}$ if it satisfies individual rationality. We now show a recursive bound that gives an upper bound on p_1.

LEMMA 6.7. *Define p_Δ such that:*

$$p_\Delta = \min_{i \in \{\Delta \ldots n\}} p(i - \Delta, i).$$

If the mechanism has a finite competitive ratio then for all $\Delta > 1$, $p_\Delta \geq \frac{\Delta+1}{3} p_1$.

PROOF. Define

$$p^*_\Delta = \begin{cases} p_1 & \text{if } \Delta = 1 \\ \frac{\Delta+1}{3} p_1 & \text{otherwise.} \end{cases}$$

We now argue by induction. Assume that the theorem holds for all $\Delta' < \Delta$. Now, combining Eqs. (6) and (7), and Theorem 6.6 we have that for any integer $a \in \{\Delta, n\}$:

$$U_0(a - \Delta, a) \leq \Delta p(a - \Delta, a) - \int_{a-\Delta}^{p_{\max}} p(s_1, a) ds_1 - \int_0^a p(a - \Delta, s_2) ds_2$$

Since p is nonnegative, decreasing in its first argument, and increasing in its second argument, this can be bounded above by:

$$\begin{aligned} U_0(a - \Delta, a) &\leq \Delta p(a - \Delta, \Delta) - \int_{a-\Delta}^{a-1} p(s_1, a) ds_1 - \int_{1+a-\Delta}^{a} p(a - \Delta, s_2) ds_2 \\ &\leq \Delta p(a - \Delta, \Delta) - \sum_{s_1 = 1+a-\Delta}^{a-1} p(s_1, a) - \sum_{s_2 = 1+a-\Delta}^{a-1} p(a - \Delta, s_2). \end{aligned}$$

By the definition of $p_{\Delta'}$, this can be bounded above by:

$$U_0(a - \Delta, a) \leq \Delta p(a - \Delta, \Delta) - 2 \sum_{\Delta'=1}^{\Delta-1} p_{\Delta'}.$$

In order to obtain a finite competitive ratio, the expected utility of the broker must be nonnegative regardless of the types of the agents. Thus $U_0(a - \Delta, a) \geq 0$, and:

$$p(a - \Delta, \Delta) \geq \frac{2}{\Delta} \sum_{\Delta'=1}^{\Delta-1} p_{\Delta'}$$

By the inductive hypothesis

$$p(a-\Delta,\Delta) \geq \frac{2}{\Delta}\sum_{\Delta'=1}^{\Delta-1} p^*_{\Delta'}.$$

Since a is any integer in $\{\Delta,\ldots,n\}$:

$$\begin{aligned} p_\Delta \geq \frac{2}{\Delta}\sum_{\Delta'=1}^{\Delta-1} p^*_{\Delta'} &= \frac{2}{\Delta}\left(1+\sum_{\Delta'=2}^{\Delta-1}\frac{\Delta'+1}{3}\right)p_1 \\ &= \frac{2}{\Delta}\left(\frac{\Delta(\Delta+1)}{6}\right)p_1 = \frac{\Delta+1}{3}p_1. \qquad \square \end{aligned}$$

PROOF (OF THEOREM 6.5). From the above lemma, $p(0,n) \geq \frac{n+1}{3}p_1$. This implies that $p_1 \leq \frac{3}{n+1}$, and the competitive ratio is no less than $\frac{\lfloor p_{\max}-p_{\min}\rfloor+1}{3}$. □

Some mechanisms are not direct, so the theorem above does not apply. An indirect mechanism may motivate the agents to lie (about valuations and/or start and end times). Furthermore, the mechanism may not even ask the agents to communicate their types: the mechanism may instead have the agents signal about their types in a different action space, which may involve an agent taking multiple actions over time.

It turns out that we can obtain the same lower bound as above even for such indirect mechanisms. To achieve this, we require that each agent have a *dominant strategy* in the mechanism, that is, the strategy of a bidder must be optimal *regardless of the other agents' bids*. (We do not require that the dominant strategy be to tell the truth.) We also require that the mechanism be (ex-post) individually rational; that is, each bidder is no worse off by participating.[9] Under these requirements, we can use the revelation principle[10] and the above technique to get the same bound.

THEOREM 6.8. *No individually rational dominant-strategy mechanism can achieve a competitive ratio for profit lower than* $\frac{\lfloor p_{\max}-p_{\min}\rfloor+1}{3}$.

7. *Conclusions and Open Questions*

In this article, we have derived bounds on competitive ratios for a number of basic online market clearing problems. These include the objectives of maximizing profit, maximizing liquidity according to several natural measures, and maximizing social welfare. Using the abstraction of online incomplete interval graphs, we obtain algorithms that achieve best-possible competitive ratios for most of these problems, and for the rest we are within a factor of two of the best that is achievable online. For the objective of maximizing number of trades, we demonstrate that by allowing the online algorithm to subsidize matches with profits from previous trades, we can

[9] A direct mechanism that is incentive compatible and individually rational for any pair of valuations is an individually rational, dominant strategy mechanism. However, in a dominant strategy mechanism, there is no assumption that the mechanism can elicit the true types of the agents.

[10] It is important to note that the revelation principle does not hold for Bayes–Nash mechanisms in the online case.

TABLE I. MAIN RESULTS OF THIS ARTICLE

Objective	Upper Bound	Lower Bound
Profit (non-IC)	$\ln(p_{\max} - p_{\min}) + 1$	$\ln(p_{\max} - p_{\min}) + 1$
Profit (non-IC)	$(1 + \epsilon)$ wrt fixed threshold with assumptions	
Profit (IC)	$p_{\max} - p_{\min}$	$\Omega(p_{\max} - p_{\min})$
Profit (IC)	$(1 + \epsilon)$ wrt fixed prices with assumptions	
# trades (non-IC)	2 (no subsidy) 1 (with subsidy)	4/3 (no subsidy)
Buy/sell volume (non-IC)	$2(\ln(p_{\max}/p_{\min}) + 1)$ (no subsidy) $\ln(p_{\max}/p_{\min}) + 1$ (with subsidy)	$\ln(p_{\max}/p_{\min}) + 1$
Social welfare (non-IC)	$r = \ln \frac{p_{\max} - p_{\min}}{(r-1)p_{\min}}$	$r = \ln \frac{p_{\max} - p_{\min}}{(r-1)p_{\min}}$
Social welfare (IC)	$2 \max(2, \ln(p_{\max}/p_{\min}))$	

perform as well as the optimal offline algorithm without subsidy. Thus, the ability to subsidize trades is at least as powerful as knowing the future, in this context.

It is a somewhat unfortunate fact about competitive analysis that no algorithm can guarantee better than a logarithmic ratio for maximizing profit. This lower bound occurs because we are comparing the performance of our online algorithm to the best performance of any arbitrary algorithm in hindsight. However, if we restrict our comparison class to a fixed, finite set of algorithms (such as the class of all fixed-threshold algorithms), and if we assume the data is not too "bursty", then we can use online learning results to achieve a $1 + \epsilon$ ratio, as we demonstrate in Section 5.1.

We also consider the issue of incentive compatibility, and show that our Greedy algorithm is incentive-compatible for the goal of maximizing social welfare. Thus, losing only a factor of 2 in competitive ratio we can achieve this desirable property. In fact, we show the Greedy algorithm is quite robust in that it maintains its competitive ratio even if bidders are allowed to modify their bids over time, making the task even harder on the online algorithm. For profit maximization, we show that no direct algorithm can achieve a sublinear competitive ratio, though we present an algorithm that has competitive ratio approaching 1 with respect to the best *fixed* pair of buy/sell prices when bids are not too bursty. We summarize our main results in Table I.

7.1. OPEN PROBLEMS. There are a number of open problems with respect to incentive-compatibility. For example in terms of competing with respect to the best fixed pair of buy and sell prices for profit maximization, our results require that bids not be too "bursty" (at most B alive at any one time). On the other hand, Deshmukh et al. [2002] provide algorithms for the offline problem where *all* bids are concurrent. Perhaps some combination can handle the general case.

There is also a gap between our best upper bound (2) and our best lower bound (3/2) for algorithms for the incomplete interval graph vertex-selection problem when *expiration times are unknown*. Algorithm 2 achieves an optimal matching but it needs to know the expiration times in advance. Related to this question, we do not know whether the Greedy algorithm can be improved upon for the setting of Section 5.2 in which a bidder may submit a bid whose value varies with time.

There has also been recent interest in offline clearing algorithms for *combinatorial* exchanges where a bid can concern multiple distinguishable items (possibly multiple units of each) [Sandholm 2002; Sandholm and Suri 2003; Sandholm et al. 2002]. A bid could state, for example, "I want to buy 20 IBM, buy 50 DELL, sell 700 HP, and get paid \$500". As we transformed the online clearing problem to an online matching problem in an incomplete interval graph, the online problem of maximizing profit or liquidity in a combinatorial exchange can be transformed to the problem of finding a matching on an incomplete interval *hypergraph* online. While we can show with a simple example that a maximum matching cannot be achieved online in the hypergraph problem, it might be possible to achieve a matching with an expected size no less than half of the maximum one.

One direction of future research is to extend these results to settings where there is a market maker who can carry inventory of the commodity (sell short and long) rather than simply deciding which bids to match.

Appendix A. Lemmas on Matchings

In this section, we prove some lemmas required for the proof of the main theorem in Appendix B. We will use some standard facts and definitions from graph theory; for a review, see Kozen [1991]. Paths will be represented as sets of edges, although sometimes it will be easier to write them as a sequence of vertices. $V(P)$ is the set of vertices of a path P. We will also use some well-known lemmas:

LEMMA A.1. *Given $W \subset V$, a perfect matching M on W, and $v, v' \in V - W$, if there exists some augmenting path P with respect to M from v to v', then $M \oplus P$ is a perfect matching on $W \cup \{v, v'\}$.*

LEMMA A.2. *The symmetric difference of two matchings M and M' consists of alternating paths and alternating cycles with respect to M.*

LEMMA A.3. *A vertex $v \in V$ is the endpoint of an alternating path in the symmetric difference of two matchings M and M' if and only if it is in $V(M) \oplus V(M')$.*

PROOF. Suppose a vertex is in $V(M) \oplus V(M')$. Then, it has an incident edge in either M or M', but not both. Hence, it can only have one incident edge in $M \oplus M'$, making it an endpoint of an alternating path.

Suppose a vertex is at the endpoint of a path in $M \oplus M'$. Then, it has one edge in $M \oplus M'$. Observe that, if it has an odd number of edges incident to it in $M \oplus M'$, the number of edges in M to it plus the number of edges in M' to it is also odd. But since the latter is bounded by zero and two, it is one. Therefore, the vertex cannot be in both $V(M)$ and $V(M')$. □

Augmenting paths capture some of the local properties of matchings. But they do not capture how to remove elements from a matching, or how to replace an element. Since these are important concepts in the online setting, we introduce some other types of paths. These paths have properties very similar to those of augmenting paths.

Definition A.4. An **abridging path** with respect to a matching M is an alternating path whose first and last edges are in M. A **replacement path** with respect to M is an alternating path whose first edge is in M, and whose last endpoint is not in $V(M)$.

LEMMA A.5. *Given $v, v' \in W$, and a perfect matching M on W, if there exists some abridging path P with respect to M from v to v', then $M \oplus P$ is a perfect matching on $W - \{v, v'\}$.*

The proof is similar to the proof of Lemma A.1.

LEMMA A.6. *Given $v \in W$ and $v' \in V - W$, and a perfect matching M on W, if there exists some replacement path P with respect to M from v to v', then $M \oplus P$ is a perfect matching on $W \cup \{v'\} - \{v\}$.*

The proof is similar to the proof of Lemma A.1.

LEMMA A.7. *Suppose W and W' are subsets of V, and there exists a perfect matching on W and a perfect matching on W', then there exists a partition of $W \oplus W'$ into sets of size two:*

$$\{\{a_1, b_1\}, \{a_2, b_2\}, \ldots, \{a_k, b_k\}\}$$

such that for all $1 \leq i \leq k$, there exists a perfect matching on $W \oplus \{a_i, b_i\}$.

PROOF. Define M to be a perfect matching on W and M' to be a perfect matching on W'. Then $M \oplus M'$ consists of a set of alternating paths and cycles. Here, we are only concerned with the paths. Since these paths only begin and end at points in $W \oplus W'$, we can partition $W \oplus W'$ where each set is the set of endpoints of a path in $M \oplus M'$. Consider a set in this partition $\{a, b\}$, where P is the path between the two elements of the set. There are three possibilities:

(1) Both vertices are in $W' - W$. Then P is an augmenting path, and $M \oplus P$ is a perfect matching on $W \cup \{a, b\} = W \oplus \{a, b\}$.
(2) One vertex is in $W - W'$ and one vertex in $W' - W$. In this case, the path P is a replacement path. Without loss of generality, assume $a \in W - W'$. Then $M \oplus P$ is a perfect matching on $W \cup \{b\} - \{a\} = W \oplus \{a, b\}$.
(3) Both vertices are in $W - W'$. The first and last edges are not in M', because a and b are not in W'. Also, the first and last edges are in $M \oplus M' \subseteq M \cup M'$. Therefore, the first and last edges are in M, and P is an abridging path. $M \oplus P$ is a perfect matching on $W - \{a, b\} = W \oplus \{a, b\}$. □

COROLLARY A.8. *If $W \subseteq V$ has a perfect matching, but it is not one of the largest sets that has a perfect matching, then there exists $v, v' \in V - W$ such that $W \cup \{v, v'\}$ has a perfect matching.*

Appendix B. *Proof of Theorem* 2.3 (*Main Theorem*)

Assume W is the set of selected vertices, then we define H_1,H_2, and H_3 as follows:

H_1: For any expired, unmatched vertex w, there does not exist any untaken vertex w' such that there is a perfect matching on $W \cup \{w, w'\}$.
H_2: For any matched vertex w, there does not exist an untaken vertex w' such that there is a perfect matching on $W \cup \{w'\} - \{w\}$ and $t_f(w) > t_f(w')$.
H_3: For any two unexpired vertices $w, w' \in W$, there exists no perfect matching on $W - \{w, w'\}$.

Let H denote the conjunction of the three invariants H_1, H_2, and H_3. We will prove that if these invariants of the algorithm hold before an event occurs, they will hold after the event. The possible events that can occur are a vertex being introduced, expiring without being matched, or expiring and being added.

LEMMA B.1. *If H holds before a vertex w' is introduced, H holds after the event.*

PROOF. We prove this for all three parts of H.

H_1: Consider w to be an expired, unmatched vertex. Here we need to prove that there exists no perfect matching on $W \cup \{w, w'\}$. We will prove this by contradiction. Suppose that M is a perfect matching on $W \cup \{w, w'\}$. Define v such that $(w', v) \in M$. Let us define $M' = M - \{(w', v)\}$. Then, M' is a perfect matching on $W - \{v\} + \{w\}$. Since w has expired and v has not expired, $t_f(w) < t_f(v)$. This contradicts the inductive hypothesis H_2.

H_2: Suppose that $w \in W$ and $t_f(w) > t_f(w')$. We need to prove that there does not exist a perfect matching for $W \cup \{w'\} - \{w\}$ by contradiction, assuming there exists a perfect matching M.

Define v such that $(w', v) \in M$. Observe that $t_f(v) > t_i(w')$, so v has not expired. Neither has w, since $t_f(w) > t_f(w') > t_i(w')$. However, $M - (w', v)$ is a perfect matching on $W - \{w, v\}$. This is a contradiction of H_3.

H_3: This cannot have any effect. □

LEMMA B.2. *If H holds before a vertex w expires without being matched, H holds after the event.*

PROOF. We prove this for all three parts of H.

H_1: Suppose w' is an untaken vertex. We need to prove that there does not exist a perfect matching on $W \cup \{w, w'\}$.

(a) Assume w' expired. By H_1, there there did not exist a perfect matching on $W \cup \{w, w'\}$.

(b) If there existed an unexpired, unmatched vertex w' such that $W \cup \{w, w'\}$ had a perfect matching, then w would have been matched.

H_2, H_3: This cannot have any effect. □

LEMMA B.3. *If H holds before a vertex v expires and is added with v', H holds after the event.*

PROOF. We prove this for all three parts of H.

H_1: Suppose w is an expired, unmatched vertex, and w' is an untaken vertex. We need to prove that there does not exist a perfect matching on $W \cup \{v, v', w, w'\}$. We can prove this by contradiction.

Observe that there exists a perfect matching on W. If there existed a perfect matching on $W \cup \{v, v', w, w'\}$, then, by Lemma A.7, one of the following conditions holds:

(a) There exists a perfect matching on $W \cup \{v', w\}$. This would contradict H_1.

(b) There exists a perfect matching on $W \cup \{v, w\}$. This would contradict H_1.

(c) There exists a perfect matching on $W \cup \{w, w'\}$. This would contradict H_1.

H_2: Consider an arbitrary vertex w in $W \cup \{v, v'\}$, and an arbitrary untaken vertex w'. Assume that $W \cup \{v, v', w'\} - \{w\}$ is a perfect matching. We must prove that $t_f(w') \geq t_f(w)$.

(a) $w = v$. Then $W \cup \{v, v', w'\} - \{w\} = W \cup \{v', w'\}$. So by H_1, w' has not expired. Then $t_f(w') \geq t_f(w)$, because w is just expiring.
(b) $w = v'$. Then $W \cup \{v, v', w'\} - \{w\} = W \cup \{v, w'\}$. So by H_1, w' has not expired. Thus if $t_f(w) > t_f(w')$, then w' would have been added instead of w.
(c) $w \in W$. Then by Lemma A.7 applied to W and $W \cup \{v, v', w'\} - \{w\}$, one of the following conditions holds:
  (i) $W \cup \{v\} - \{w\}$ and $W \cup \{v', w'\}$ have perfect matchings. Then by H_1, w' has not expired, and $t_f(w) < t_f(v)$, so w has expired.
  (ii) $W \cup \{v'\} - \{w\}$ and $W \cup \{v, w'\}$ have perfect matchings. Then by H_1, w' has not expired, so $t_f(w') \geq t_f(v')$. Also, $t_f(w) \leq t_f(v')$.
  (iii) $W \cup \{w'\} - \{w\}$ has a perfect matching. Then by H_2, $t_f(w') \geq t_f(w)$.

H_3: A vertex v expires and is added with v'. Suppose that $w, w' \in W \cup \{v, v'\}$. Assume that $W \cup \{v, v'\} - \{w, w'\}$ has a perfect matching. We must prove that w or w' has expired. Consider three cases:
(a) $w = v$. Then w has expired.
(b) $w = v'$. Then $W \cup \{v, v'\} - \{w, w'\} = W \cup \{v\} - \{w'\}$. Thus by H_2, $t_f(v) \geq t_f(w')$, so w' has already expired(a contradiction).
(c) $w \in W$. Then by Lemma A.7 applied to W and $W \cup \{v, v'\} - \{w, w'\}$, one of the following conditions holds:
  (i) $W \cup \{v\} - \{w\}$ has a perfect matching. By H_2, $t_f(v) \geq t_f(w)$, so w has expired.
  (ii) $W \cup \{v\} - \{w'\}$ has a perfect matching. By H_2, $t_f(v) \geq t_f(w')$, so w' has expired.
  (iii) $W - \{w, w'\}$ has a perfect matching. By H_3, either w or w' has expired. □

PROOF (OF THEOREM 2.3). We will prove by induction that H holds at the termination of the algorithm. First, observe that H_1, H_2, and H_3 are all properties of introduced vertices, so when there are no vertices introduced, they cannot be violated. This is the base case. Inductively, by Lemmas B.1, B.2, B.3, if these properties hold before an event, they hold afterward as well. Thus, H holds at the termination of the algorithm. Specifically, H_1 implies that there exists no $\{v, v'\} \subseteq V - W$ such that $W \cup \{v, v'\}$ has a perfect matching. By Corollary A.8, W is one of the largest sets with a perfect matching. □

Appendix C. Proof of Theorem 5.1 (*Performing Nearly as Well as the Best Threshold in Hindsight*)

The setting is we have N algorithms ("experts") and we can switch from one to another at cost $\leq D$. At each time step (introduction of a bid) the algorithms experience gains between 0 and $g_{\max}$. (We use $g_{\max}$ instead of $p_{\max}$ because we will be using "p" for probabilities). At any point in time, our overall "master" algorithm will be a probability distribution over the N base algorithms. Specifically, we use the standard multiplicative-weights algorithm [Littlestone and Warmuth 1994; Freund and Schapire 1996] as follows:

—We begin with each base algorithm having a weight $w_i = 1$. Let $W = \sum_i w_i$. Our probability distribution over base algorithms will be $p_i = w_i/W$.

—After each event, we update weights according to the rule:

$$w_i \leftarrow w_i(1+\hat{\epsilon})^{g_i/g_{\max}},$$

where g_i is the gain of the ith algorithm, and $\hat{\epsilon}$ is defined below. We then update our probability distribution p_i accordingly. Note: if $\vec{p}$ is the old distribution and $\vec{p}'$ is the new distribution, we can do this such that the expected movement cost is at most $\frac{D}{2}L_1(\vec{p}, \vec{p}')$ where $L_1(\vec{p}, \vec{p}')$ is the L_1 distance between the two distributions.

THEOREM C.1. *If we use $\hat{\epsilon} = \frac{\epsilon g_{\max}}{2D}$, and under the assumption that $D \geq g_{\max}$, the above algorithm produces an expected gain at least*

$$\mathrm{OPT}(1-\epsilon) - \frac{2D}{\epsilon}\ln N,$$

where OPT *is the gain of the best of the N base algorithms in hindsight.*

PROOF. Suppose the algorithm currently has weights $w_1, \ldots, w_n$, observes a gain vector $g_1, \ldots, g_n$ and then updates the weights to $w'_1, \ldots, w'_n$, where $w'_i = w_i(1+\hat{\epsilon})^{g_i/g_{\max}}$. Let $W = \sum_i w_i$ and let $W' = \sum_i w'_i$. Let $p_i = w_i/W$ and let $p'_i = w'_i/W'$.

The expected gain of the online algorithm from this event is $\sum_i p_i g_i$, which we will call E_t if this is event t. The expected cost due to moving from probability distribution $\vec{p}$ to distribution $\vec{p}'$ is at most

$$D\sum_{i:p'_i>p_i}(p'_i - p_i) \leq \frac{D}{W}\sum_{i:p'_i>p_i}(w'_i - w_i) \leq \frac{D}{W}(W'-W) = D\left(\frac{W'}{W}-1\right),$$

where the first inequality above uses the fact that $W' \geq W$, and the second inequality uses the fact that $w'_i \geq w_i$ for *all* i, not just the ones such that $p'_i > p_i$.

The next step in the analysis is to upper bound W' as a function of W. To do this, we use the fact that for $x \leq 1$, $(1+\alpha)^x \leq 1 + x\alpha$. So,

$$W' \leq \sum_i w_i\left(1+\frac{\hat{\epsilon}g_i}{g_{\max}}\right) = W\sum_i p_i\left(1+\frac{\hat{\epsilon}g_i}{g_{\max}}\right) = W\left(1+\frac{\hat{\epsilon}}{g_{\max}}E_t\right).$$

This upper bound on W' is useful in two ways. First, we can now analyze the expected movement cost: $D(W'/W - 1) \leq D(\frac{\hat{\epsilon}}{g_{\max}}E_t)$. So, our total expected gain for event t is at least $E_t(1 - \frac{D\hat{\epsilon}}{g_{\max}})$.

Second, we can use this to give an upper bound on the total weight W_{final} at the end of the process:

$$W_{\mathit{final}} \leq N\prod_t\left(1+\frac{\hat{\epsilon}}{g_{\max}}E_t\right).$$

This upper bound can be compared to a lower bound

$$W_{\mathit{final}} \geq (1+\hat{\epsilon})^{\mathrm{OPT}/g_{\max}}$$

which is simply the final weight of the best of the N algorithms. Taking logs on the upper and lower bounds to W_{final} and using the fact that $\ln(1+x) \in [x - x^2/2, x]$

for $x \in [0, 1]$, we have:

$$\frac{\mathrm{OPT}}{g_{\max}}(\hat{\epsilon}-\hat{\epsilon}^2/2) \leq \frac{\mathrm{OPT}}{g_{\max}}\ln(1+\hat{\epsilon}) \leq \ln N+\sum_t \ln(1+\frac{\hat{\epsilon}}{g_{\max}}E_t) \leq \ln N+\frac{\hat{\epsilon}}{g_{\max}}\sum_t E_t.$$

So,

$$\sum_t E_t \geq \mathrm{OPT}(1-\hat{\epsilon}/2) - \frac{g_{\max}}{\hat{\epsilon}}\ln N.$$

Finally, using the fact that the total expected gain G_{alg} (with movement costs included) is at least $(1-\frac{D\hat{\epsilon}}{g_{\max}})\sum_t E_t$, and using our definition $\hat{\epsilon} = \epsilon g_{\max}/(2D)$, we have:

$$G_{alg} \geq \mathrm{OPT}(1-\hat{\epsilon}/2)(1-\epsilon/2)-(1-\epsilon/2)\frac{2D}{\epsilon}\ln N \geq \mathrm{OPT}(1-\epsilon)-\frac{2D}{\epsilon}\ln N. \quad \Box$$

ACKNOWLEDGMENT. The authors would like to thank Ke Yang for a number of helpful suggestions.

REFERENCES

AWERBUCH, B., BARTAL, Y., FIAT, A., AND ROSÉN, A. 1994. Competitive non-preemptive call control. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. ACM, New York, 312–320.

BAGCHI, A., CHAUDHARY, A., GARG, R., GOODRICH, M., AND KUMAR, V. 2001. Seller-focused algorithms for online auctioning. In *Proceedings of the 7th International Workshop on Algorithms and Data Structures (WADS)*. 135–147.

BAR-YOSSEF, Z., HILDRUM, K., AND WU, F. 2002. Incentive-compatible online auctions for digital goods. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. ACM, New York.

BERGE, C. 1957. Two theorems in graph theory. In *Proceedings of the National Academy of Sciences. 43*, 842–844.

BLUM, A., AND BURCH, C. 2000. On-line learning and the metrical task system problem. *Mach. Learn. 39*, 1 (Apr.), 35–58.

BLUM, A., AND HARTLINE, J. 2005. Near-optimal online auctions. In *Procedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. ACM, New York, 1156–1163.

BLUM, A., KUMAR, V., RUDRA, A., AND WU, F. 2004. Online learning in online auctions. *Theoret. Comput. Sci. 324*, 2–3, 137–146. (Special issue in memory of Steven Seiden.)

BORODIN, A., AND EL-YANIV, R. 1998. *Online Computation and Competitive Analysis*. Cambridge University Press, Cambridge, UK.

BREDIN, J., AND PARKES, D. 2005. Models for truthful online double auctions. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*. 50–59.

CESA-BIANCHI, N., FREUND, Y., HELMBOLD, D., HAUSSLER, D., SCHAPIRE, R., AND WARMUTH, M. 1997. How to use expert advice. *J. ACM 44*, 3, 427–485.

DESHMUKH, K., GOLDBERG, A. V., HARTLINE, J. D., AND KARLIN, A. R. 2002. Truthful and competitive double auctions. In *Proceedings of the 10th Annual European Symposium on Algorithms (ESA)*. 361–373.

DOMOWITZ, I. 1993. Automating the continuous double auction in practice: Automated trade execution systems in financial markets. In *The Double Auction Market*, D. Friedman and J. Rust, Eds. Santa Fe Institute Studies in the Sciences of Complexity, vol. 14. Addison-Wesley, Reading, PA, 27–60.

EL-YANIV, R., FIAT, A., KARP, R. M., AND TURPIN, G. 1992. Competitive analysis of financial games. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society Press, Los Alamitos, CA, 327–333.

FREUND, Y., AND SCHAPIRE, R. 1996. Game theory, on-line prediction and boosting. In *Proceedings of the 9th Annual Conference on Computational Learning Theory (COLT)*. ACM, New York, 325–332.

GOLDBERG, A., HARTLINE, J., AND WRIGHT, A. 2000. Competitive auctions and multiple digital goods. Tech. rep., InterTrust 00–01.

GOLDBERG, A., HARTLINE, J., AND WRIGHT, A. 2001. Competitive auctions and digital goods. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (Washington, DC.), ACM, New York.

GOLDMAN, S., PARWATIKAR, J., AND SURI, S. 2000. On-line scheduling with hard deadlines. *J. Algorithms 34*, 370–389.

GONG, J. 2002. Exchanges for complex commodities: Search for optimal matches. M.S. dissertation, University of South Florida.

HAGERTY, K., AND ROGERSON, W. 1987. Robust trading mechanisms. *J. Econ. Theory 42*, 94–107.

HAJIAGHAYI, M. T., KLEINBERG, R., AND PARKES, D. C. 2004. Adaptive limited-supply online auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*. ACM, New York, 71–80.

HAJIAGHAYI, M. T., KLEINBERG, R. D., MAHDIAN, M., AND PARKES, D. C. 2005. Online auctions with re-usable goods. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC)*. ACM, New York, 165–174.

KARP, R. M., VAZIRANI, U. V., AND VAZIRANI, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)* (Baltimore, MA). ACM, New York, 352–358.

KOZEN, D. 1991. *The Design and Analysis of Algorithms*. Springer-Verlag, New York.

LAVI, R., AND NISAN, N. 2000. Competitive analysis of incentive compatible on-line auctions. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)* (Minneapolis, MN). ACM, New York, 233–241.

LIPTON, R. J., AND TOMKINS, A. 1994. Online interval scheduling. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. ACM, New York, 302–311.

LITTLESTONE, N., AND WARMUTH, M. K. 1994. The weighted majority algorithm. *Inf. Computation 108*, 212–261.

MYERSON, R., AND SATTERTHWAITE, M. 1983. Efficient mechanisms for bilateral trading. *J. Econ. Theory 28*, 265–281.

SANDHOLM, T. 2002. eMediator: A next generation electronic commerce server. *Computat. Intel. 18*, 4, 656–676. (Special issue on Agent Technology for Electronic Commerce.)

SANDHOLM, T., LEVINE, D., CONCORDIA, M., MARTYN, P., HUGHES, R., JACOBS, J., AND BEGG, D. 2006. Changing the game in strategic sourcing at Procter & Gamble: Expressive competition enabled by optimization. *Interfaces 36*, 1, 55–68.

SANDHOLM, T., AND SURI, S. 2002. Optimal clearing of supply/demand curves. In *Proceedings of the 13th Annual International Symposium on Algorithms and Computation (ISAAC)*. Vancouver, Canada. (Also appeared in the proceedings of the AAAI-02 workshop on Agent-Based Technologies for B2B Electronic Commerce (AAAI Technical Report WS-02-01), pp. 15–22, Edmonton, Ont., Canada.

SANDHOLM, T., AND SURI, S. 2003. BOB: Improved winner determination in combinatorial auctions and generalizations. *Artif. Intel. 145*, 33–58.

SANDHOLM, T., AND SURI, S. 2006. Side constraints and non-price attributes in markets. *Games and Economic Behavior 55*, 321–330. (Early version in IJCAI-01 Workshop on Distributed Constraint Reasoning, 55–61.)

SANDHOLM, T., SURI, S., GILPIN, A., AND LEVINE, D. 2002. Winner determination in combinatorial auction generalizations. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. Bologna, Italy, 69–76.

WURMAN, P., WALSH, W., AND WELLMAN, M. 1998a. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Syst. 24*, 1, 17–27.

WURMAN, P., WELLMAN, M., AND WALSH, W. 1998b. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In *Proceedings of the 2nd International Conference on Autonomous Agents (AGENTS)* (Minneapolis/St. Paul, MN). 301–308.

ZINKEVICH, M. 2004. Theoretical guarantees for algorithms in multiagent settings. Ph.D. dissertation. Carnegie Mellon University. CMU-CS-04-161.

RECEIVED JANUARY 2003; ACCEPTED JUNE 2006