How Brokers Can Optimally Abuse Traders

Manuel Lafond ✉ 🆔
University of Sherbrooke, Canada

─── Abstract ───

Traders buy and sell financial instruments in hopes of making profit, and brokers are responsible for the transaction. There are several hypotheses and conspiracy theories arguing that in some situations, brokers want their traders to lose money. For instance, a broker may want to protect the positions of a privileged customer. Another example is that some brokers take positions opposite to their traders', in which case they make money whenever their traders lose money. These are reasons for which brokers might manipulate prices in order to maximize the losses of their traders.

In this paper, our goal is to perform this shady task optimally – or at least to check whether this can actually be done algorithmically. Assuming total control over the price of an asset (ignoring the usual aspects of finance such as market conditions, external influence or stochasticity), we show how in quadratic time, given a set of trades specified by a stop-loss and a take-profit price, a broker can find a maximum loss price movement. We also look at an online trade model where broker and trader exchange turns, each trying to make a profit. We show in which condition either side can make a profit, and that the best option for the trader is to never trade.

1 Introduction

Trading is the practice of buying or selling financial assets with the aim of making a profit. A trader can buy an instrument at some price p and sell it at price p', making a profit of $p' - p$ (which might be negative). The trader can also sell an unowned instrument at some price p with the obligation to buy it back someday, say at a time where the new price is p', making a profit of $p - p'$ (this is called *shorting*). A *broker* is usually responsible for the execution of a trade, taking care of the technical aspects of the transaction.

This power over trade execution has given rise to several conspiracy theories. This has been especially prevalent in the year 2021 where unprecedented financial events occurred. One of them was the rise and fall of the GameStop (GME) stock that started in January. In short, large institutions had significantly shorted the stock and, in a concerted counteract effort, retail traders massively bought it and made its price go up 30-fold. At this point, several traders were unable to buy more GME stocks, the transaction being blocked by their broker (see [3]). This may have been caused by technical issues, but of course the Internet accused brokers of manipulating prices to protect the short positions of their large clients. The stock went back near its original price, allowing the shorts to limit their losses, and then stabilized a bit higher in the months after. The year 2021 has also seen cryptocurrencies gain monstrous gains in value. They are under less regulations than stocks and have been suspected of shady price manipulation techniques, for instance pumps and dumps, since their rise in popularity [1]. As a final example, the foreign exchange currency market is largely managed by brokers called *market-makers*. These place trades in the opposite direction of their traders – if a trader wants to buy an asset, the market-maker will sell it, and if the

11th International Conference on Fun with Algorithms (FUN 2022).
Editors: Pierre Fraigniaud and Yushi Uno; Article No. 18; pp. 18:1–18:19

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

trader wants to sell it, they will buy it. After all, there has to be two parties involved in a transaction, and market-makers assume one of the roles. This gives them an incentive for their clients to perform poorly. See [2] for a gentle introduction on market makers and [14] for a thesis on the topic. Price manipulation theories are based on the arguments that large brokerage firms have access to enough funds to control prices to some extent, along with statistics showing that a majority of traders lose money (see [4]).

Now, taking such accusations seriously is likely to make economists jump out of their seats, since there are too many factors driving asset prices for a single entity to control it. But here, we rather take an algorithmic perspective on these theories. To take it to the extreme, suppose that brokers have total control over the prices. Armed with the knowledge of every trade that is currently open, their goal is to make people lose as much money as possible. The question is: even with the ultimate power of price manipulation, *can they*? Is this optimization problem easy? In this paper, we wear the hat of (would-be) mischievous brokers and devise price manipulation algorithms to do our evil bidding. Note that this question has another interpretation: as a trader, what is the worst that could happen with a set of opened trades[1]? There seems to be no algorithmic answer in the literature for this simple question. Algorithms exist for the seemingly related notions of *value at risk* [10] and *maximum loss* [13], but these measures are based on stochastic prices, operate on multiple assets and are subject to various market conditions, unlike here where we assume full control.

Our contributions. We model trades as bounded by two closing prices - a winning and a losing price. This is typical in trading: traders want to limit their risk and often set up a *stop-loss*, a price at which the trade closes automatically when too much losses are incurred. This is usually accompanied with a *take-profit* price, which closes the trade when its profit is high enough. Brokers can use these two pieces of information to their advantage, leading to several problems. First, we study the *offline* problem where, given a set of trades, we ask for a price movement that maximizes trader losses. We show that this problem reduces to finding a maximum independent set on the *trade conflict graph*, which exhibits which trades cannot be won simultaneously. This graph is bipartite and thus our problem can be solved in polynomial time using maximum flow techniques [11, 12]. By digging a bit deeper, we develop a specialized $O(n^2)$ time algorithm by characterizing trade conflict graphs geometrically. That is, we define an equivalent graph class called *bicolored plane domination graphs*, which are shown to be chordal bipartite (see [5]), and on which dynamic programming can be performed for our purposes. This class is interesting in itself and leads to several open algorithmic and graph theoretical problems.

Second, we look at the *online* setting. That is, the trader can add a new trade or close a trade at any given time, and the broker still has to maximize losses. This leads to a two-player game where the trader and broker exchange turns. We show that there is essentially one strategy that the broker must use, which is to always move the price greedily in the direction of maximum *potential* profit. In particular, using the independent set algorithm from above in the online setting incurs infinite losses against an optimal trader. We conclude by showing that if the broker uses this strategy, the trader should simply not open any trade and get rich through other means.

---

[1] This suggests that traders could open trades in opposing directions. This appears to make little sense, but it has its advantages – in fact, this is a well-known risk management technique called *hedging*.
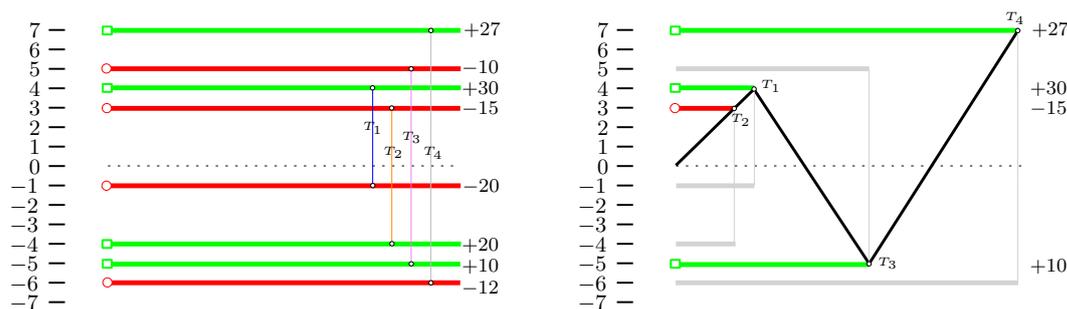
Figure 1 Left: an example input for the Maximum Trader Abuse problem with $|\mathcal{T}| = \{T_1, T_2, T_3, T_4\}$. Each trade has a winning price (green lines that start with a square) and a losing price (red lines that start with a circle). Each winning and losing price has a corresponding profit indicated on the right. For instance, $T_1 = (4, -1, f)$ where $f(4) = 30$ and $f(-1) = -20$. (b) An optimal price movement M for $\mathcal{T}$ that makes a total profit of $30 - 15 + 10 + 27 = 52$. The green/red (square/circle) lines now indicate the lifespan of each trade according to M. The profit points not realized by M are grayed-out. Only T_2 is lost since price 3 is reached before -4.

2   Preliminary notions

We use the notation $[a] = \{1, 2, \dots, a\}$ and $[a, b] = \{x \in \mathbb{Z} : a \le x \le b\}$. Given an integer $x \in \mathbb{Z}$, the sign of x is denoted $sgn(x)$, which is either 1 or -1. All graphs in this paper are finite and simple. For a graph G and $X \subseteq V(G)$, $G[X]$ denotes the subgraph induced by X. A *weighted graph* is a pair (G, h) where G is a graph and $h : V(G) \to \mathbb{R}$ assigns a weight to each vertex. For $X \subseteq V(G)$, we write $h(X)$ for the sum of weights of vertices in X. Some of our routines will require to sort integers. We will write *integer sorting time* to refer to the time required to sort a list of n arbitrary integers, when n is clear from the context (this can range from $O(n \log \log n)$ to $O(n)$ depending on conditions that we prefer to leave out of consideration [9, 8]).

Regarding the financial notions used in this paper, a *price* is simply an integer, possibly negative. Using integral prices is justified by the fact that in trading, prices are usually handled to the fourth of fifth decimal and may easily be treated as integers. All trades are done on a single asset, and for our purposes no fee is required to open a trade (although the fees could be embedded in the profit functions described below). We assume that the current price of the asset is 0.

A *trade* $T = (w, \ell, f)$ consists of three parameters: w is the winning price from the broker's point of view, ℓ is the losing price from the broker's point of view, and $f : \{w, \ell\} \to \mathbb{R}$ is the profit made by the broker if the price is attained before the other (see Figure 1 and the description of price movements below). Note that profits can be negative. We require that either $\ell < 0 < w$, in which case T is an *up* trade, or $w < 0 < \ell$, in which case T is a *down* trade. We will assume that $f(w) \ge f(\ell)$ (otherwise, we can flip the roles of w and ℓ).

Note that in a typical real-life trade setting, we have $f(w) > 0$ since w represents a profit point and $f(\ell) < 0$ since ℓ it represents a loss point. Moreover, profits and losses are usually linearly related, i.e. there is some δ such that $f(w) = f(\ell) + \delta \cdot |w - \ell|$ (as is the case in Figure 1). However, such restrictions will not be needed for our algorithms until we discuss online trades, and we prefer to keep f generic. As another side note, in our terminology, an *up* trade corresponds to a *sell* from the trader since the mischievous broker makes profit when the trader loses, and likewise a *down* trade is a *buy* from the trader. Since we take the broker's viewpoint in this paper, trades are described with the broker's preferred direction.

A *price movement* $M = (m_1, \ldots, m_k)$ is a sequence where $m_i \in \{+1, -1\}$ for each $i \in [k]$. The current price after the j-th movement is $\sum_{i=1}^{j} m_i$. Consider a trade $T = (w, \ell, f)$ and a price movement M. If, under M, price w is reached before ℓ, then T is *won* and a profit of $f(w)$ is made, and if instead price ℓ is reached before w, then T is *lost* and a profit of $f(\ell)$ is incurred. The first price in $\{w, \ell\}$ reached by M is denoted $close_M(T)$ and is called the *closing price* of T under M. If M does not reach either w or ℓ, then $close_M(T)$ is undefined. If it is defined, then the broker's profit made from T under price movement M can be written as $f(close_M(T))$. Given a set of trades $\mathcal{T}$, we say that price movement M is *valid for* $\mathcal{T}$ if $close_M(T)$ is defined for every $T \in \mathcal{T}$. If M is valid, the *total profit* of M for $\mathcal{T}$ is

$$profit_{\mathcal{T}}(M) = \sum_{T=(w,\ell,f) \in \mathcal{T}} f(close_M(T))$$

The problem statement follows:

The **Maximum Trader Abuse** problem:
Given: a set of trades $\mathcal{T}$;
Find: a valid price movement M that maximizes $profit_{\mathcal{T}}(M)$.

We will denote $n = |\mathcal{T}|$ unless stated otherwise. We will say that $\mathcal{T}' \subseteq \mathcal{T}$ is an *optimal set of trades* if there exists a price movement M that maximizes $profit_{\mathcal{T}}(M)$ such that the set of trades won by M is precisely $\mathcal{T}'$. This definition implies that the problem is equivalent to finding an optimal set of trades.

Observe that we could have removed the validity requirement on M in the problem definition. This would allow the broker to leave some trades open forever by never reaching one of their closing prices. This appears to be an important difference, and we leave it as an open question whether our techniques can handle this variant. On another note, let us briefly discuss the size of M. Consider a trade $T = (w, \ell, f) \in \mathcal{T}$, and notice that w or ℓ are not necessarily polynomial in n. Since M has to close T, representing M as a sequence of unit movements might yield an exponential-size output, which is not necessary. Instead, M can be described by the sequence of prices in which it changes direction, from which the +1/-1 sequence can easily be inferred. That is, we may represent the movement as a sequence of prices $M = (p_1, p_2, \ldots, p_k)$, where $p_1 = 0$ is the initial price. Then for each $i \in \{0, \ldots, k-1\}$, $|p_{i+1} - p_i|$ steps must be performed in the same direction (up if $p_{i+1} > p_i$, and down otherwise). In fact, each p_i can be assumed to be one of the values in $\bigcup_{(w,\ell,f) \in \mathcal{T}} \{w, \ell\}$ since only these determine profits, which guarantees that the output can be made of linear size.

3   Maximizing trader abuse in quadratic time

We first show that finding an optimal M reduces to finding a maximum weight independent set in a bipartite graph, which we call the *trade conflict graph*, that describes which pairs of trades cannot be won together. The latter problem can be solved in polynomial time using standard maximum flow techniques. We then study the trade conflict graph a bit more in-depth and show that its properties lead to a $O(n^2)$ time algorithm. This is achieved by representing the trade as a set of green and red points on the 2D plane where conflicts correspond to bicolored domination relationships.

We say that a set of trades $\mathcal{T}$ is *compatible* if there exists a price movement M such that for all $(w, \ell, f) \in \mathcal{T}$, we have $close_M(T) = w$. Otherwise $\mathcal{T}$ is *incompatible*. Two trades T_1, T_2 are compatible if the set $\{T_1, T_2\}$ is compatible (and otherwise incompatible). Pairs of incompatible trades are easy to characterize.

▶ **Lemma 1.** *Let $T_1 = (w_1, \ell_1, f_1)$ and $T_2 = (w_2, \ell_2, f_2)$ be two trades. Then T_1 and T_2 are incompatible if and only if $sgn(w_1) \neq sgn(w_2)$, $|\ell_1| \leq |w_2|$ and $|\ell_2| \leq |w_1|$.*

Proof. First assume that w_1 and w_2 have a different sign, and that $|\ell_1| \leq |w_2|$ and $|\ell_2| \leq |w_1|$ hold. As both trades are of opposite signs, it is impossible to reach price w_2 without reaching ℓ_1 first (or simultaneously if $w_2 = \ell_1$), and it is impossible to reach w_1 without reaching ℓ_2 first (or simultaneously). Thus, one of the loss prices must be hit, making the trades incompatible.

We prove the converse direction by contraposition. Assume that one of the three conditions does not hold. Suppose first that $sgn(w_1) = sgn(w_2)$. If $w_1, w_2 > 0$, then the price movement that always goes up wins both trades, and if $w_1, w_2 < 0$, we can make the price go down. In either case, T_1 and T_2 are compatible. We may thus assume that $sgn(w_1) \neq sgn(w_2)$. If $|\ell_1| > |w_2|$, we first win T_2 by moving the price to w_2. This does not reach ℓ_1 and so T_1 is still open at this point. It then suffices to go from w_2 to w_1, winning both trades. The same idea applies when $|\ell_2| > |w_1|$. ◀

For example, trades T_1 and T_2 in Figure 1 are incompatible.

Obviously, if a set of trades $\mathcal{T}$ has two trades T_1, T_2 that are incompatible, then they cannot both be won and trivially $\mathcal{T}$ is incompatible. Luckily, the converse also holds.

▶ **Lemma 2.** *Let $\mathcal{T}$ be a set of trades in which every pair of trades is compatible. Then $\mathcal{T}$ is compatible. Moreover, one can construct in integer sorting time a price movement M that wins every trade of $\mathcal{T}$.*

Proof. We use induction on $|\mathcal{T}|$. The case $|\mathcal{T}| = 1$ is trivial: it suffices to go to the single winning price. Now assume that $|\mathcal{T}| > 1$. Note that for any proper subset $\mathcal{T}'$ of $\mathcal{T}$, each pair of $\mathcal{T}'$ is compatible. We may thus assume by induction that the statement holds for any such $\mathcal{T}'$. For a price p, denote $\mathcal{T}(p) = \{(w, \ell, f) \in \mathcal{T} : p = w \text{ or } p = \ell\}$. Let p^+ be the minimum value above 0 such that $\mathcal{T}(p^+) \neq \emptyset$, and let p^- be the maximum value below 0 such that $\mathcal{T}(p^-) \neq \emptyset$. If all trades of $\mathcal{T}(p^+)$ are won at price p^+, we can raise the price from 0 to p^+, win these trades, bring the price back to 0 and apply induction on $\mathcal{T} \setminus \mathcal{T}(p^+)$ to win every other trade. The same applies if all trades of $\mathcal{T}(p^-)$ are won at p^-. So suppose that there are $T_1 = (w_1, \ell_1, f_1) \in \mathcal{T}(p^+)$ and $T_2 = (w_2, \ell_2, f_2) \in \mathcal{T}(p^-)$ that are losing, i.e. such that $\ell_1 = p^+$ and $\ell_2 = p^-$. Then T_1 is a *down* trade and T_2 is an *up* trade, and hence $sgn(w_1) \neq sgn(w_2)$. Moreover, $w_2 \geq \ell_1 > 0$ by our choice of p^+, and similarly $w_1 \leq \ell_2 < 0$. These imply that $|\ell_1| \leq |w_2|$ and $|\ell_2| \leq |w_1|$ and, by Lemma 1, T_1 and T_2 are incompatible, a contradiction. It follows that some price movement can win every trade of $\mathcal{T}$.

One can derive a sorting time algorithm from the above argument. First obtain a sorted list A_+ of the all the trades (w, ℓ, f) with respect to $\max\{w, \ell\}$ in ascending order, and a sorted list A_- of all the trades with respect to $\min\{w, \ell\}$ in descending order. The first element of A_+ has price p^+, and the first price in A_- is p^-. By advancing through A_+ (resp. A_-) as long as the trade prices are p^+ (resp. p^-), we build the set L_+ (resp. L_-) of the trades that are lost at price p^+ (resp. p^-). By the above argument, one of L_+ or L_- is empty. If, say, $L_+ = \emptyset$, we move the price to p^+, win every trade at that price, and add those trades to a set C of closed trades. We also remove from L_- those trades that just closed. We then move forward in A_+ to find the next p^+ and the next L_+ and, if necessary, we advance through A_- and find the next L_-. Again, one of A_+ or A_- will be empty, so we repeat. The process is the same if L_- is empty instead. Note that in subsequent iterations, one must check whether a trade is in C before adding it to L_+ or L_-. The whole procedure can be done by traversing A_+ and A_- once, and by adding/removing each trade in L_+ or L_- at most once, and $O(n)$ price movements are added, which takes overall $O(n)$ time after sorting. ◀

Lemma 2 essentially states that we can devote our efforts to finding a set of pairwise-compatible trades $\mathcal{T}' \subseteq \mathcal{T}$ that maximizes profit, although we need to account for the losses incurred by the trades *not* in $\mathcal{T}'$. We can thus reformulate the problem in graph-theoretical terms.

▶ **Definition 3.** *The* trade conflict graph $G(\mathcal{T}) = (V, E)$ *of a set of trades* $\mathcal{T}$ *is the graph with vertex set* $V = \mathcal{T}$, *and in which there is an edge between each pair of incompatible trades.*

Furthermore, the weighing *of* $\mathcal{T}$ *is the function* $h : \mathcal{T} \to \mathbb{R}$ *that assigns the weight* $f(w) - f(\ell)$ *to each trade* $(w, \ell, f) \in \mathcal{T}$.

Note that all weights of h are non-negative since $f(w) \geq f(\ell)$ is assumed to hold for all trades. Also observe that by Lemma 1, testing compatibility on a pair of trades can be done in constant time, and thus $G(\mathcal{T})$ can be built in time $O(n^2)$. Perhaps $G(\mathcal{T})$ could be constructed faster with more refined ideas, but we shall not dwell on this. Now, as per Lemma 2, $\mathcal{T}' \subseteq \mathcal{T}$ is compatible if and only if $\mathcal{T}'$ forms an independent set in $G(\mathcal{T})$ (a set of vertices with no shared edges). As we show below, the h weighing is adjusted so that our problem reduces to finding a maximum weight independent set.

▶ **Theorem 4.** *Let* $\mathcal{T}$ *be a set of trades. Then* $\mathcal{T}^* \subseteq \mathcal{T}$ *is an optimal set of winning trades if and only if* $\mathcal{T}^*$ *is a maximum weight independent set in* $(G(\mathcal{T}), h)$, *where* h *is the weighing function of* $\mathcal{T}$.

Proof. Let M be any price movement that closes every trade and let $\mathcal{T}'$ be the set of trades won by M. Note that $\mathcal{T}'$ must be an independent set of $G(\mathcal{T})$. Moreover, the profit incurred by M is

$$
\sum_{T \in \mathcal{T}} close_M(T) = \sum_{(w,\ell,f) \in \mathcal{T}'} f(w) + \sum_{(w,\ell,f) \in \mathcal{T} \setminus \mathcal{T}'} f(\ell)
$$

$$
= \sum_{(w,\ell,f) \in \mathcal{T}'} f(w) + \left(\sum_{(w,\ell,f) \in \mathcal{T}} f(\ell) - \sum_{(w,\ell,f) \in \mathcal{T}'} f(\ell) \right)
$$

$$
= \sum_{(w,\ell,f) \in \mathcal{T}'} (f(w) - f(\ell)) + \sum_{(w,\ell,f) \in \mathcal{T}} f(\ell)
$$

$$
= \sum_{T \in \mathcal{T}'} h(T) + \sum_{(w,\ell,f) \in \mathcal{T}} f(\ell)
$$

The term $t := \sum_{(w,\ell,f) \in \mathcal{T}} f(\ell)$ does not depend on the choice of M and can be ignored since it does not contribute to the optimization criterion. Therefore, each M of profit $k + t$ corresponds to an independent set $\mathcal{T}'$ of weight k, and conversely each independent set of weight k corresponds to a price movement of profit at least $k + t$, by Lemma 2. It follows that finding an optimal M is equivalent to finding an independent set in $G(\mathcal{T})$ of maximum weight with respect to h.   ◀

It is not hard to see that $G(\mathcal{T})$ is bipartite. Indeed, letting $\mathcal{T}^+ = \{(w, \ell, f) \in \mathcal{T} : w > 0\}$ and $\mathcal{T}^- = \{(w, \ell, f) \in \mathcal{T} : w < 0\}$, one can see by Lemma 1 that $\{\mathcal{T}^+, \mathcal{T}^-\}$ is a partition of $\mathcal{T}$ into two independent sets. Finding a maximum weight independent set in a bipartite graph G can be done in polynomial-time using a maximum-flow reduction. This can be implemented to run in time $O(n^3)$ using the Stoer-Wagner min-cut algorithm [12] or a variety of max-flow algorithms that run in time $O(nm)$, where m is the number of conflicts [11].
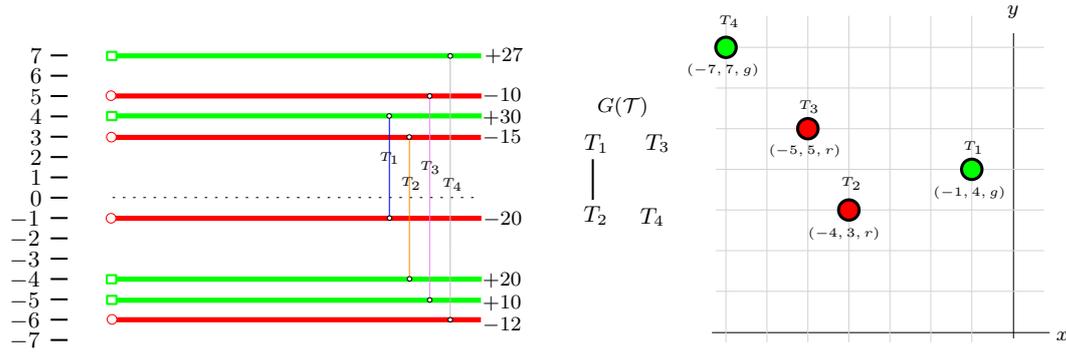
Figure 2 Left: the four trades from the previous example. Middle: the graph $G(\mathcal{T})$, which has only one incompatible pair. Right: A bicolored plane domination model for $G(\mathcal{T})$. This is the model used in the proof of Theorem 5.

3.1 Bicolored plane domination models for trade conflict graphs

We now reformulate trade conflict graphs in terms of two-colored points on the 2D plane. A *colored point* is a triple (x, y, c) where x and y refer to horizontal and vertical plane coordinates, respectively, and $c \in \{green, red\}$ is the color of the point. We say that a colored point (x, y, c) *dominates* another colored point (x', y', c') if $x \geq x'$ and $y \geq y'$.

Let $G = (V, E)$ be a graph. We say that G is a *bicolored plane domination graph* if one can assign to each $v \in V$ a colored point $p_v = (x_v, y_v, c_v)$ such that $uv \in E(G)$ if and only if $c_u \neq c_v$ and the green point of $\{p_u, p_v\}$ dominates the red point of $\{p_u, p_v\}$. In words, vertices correspond to colored points, and there is an edge between two vertices if their points are green and red, and if the green is to the upper-right of the red. If this is the case, the multiset of colored points p_u is called a *bicolored plane domination model* for G (multiset because two vertices could be assigned points with the same coordinates and color). As it turns out, this coincides with trade conflict graphs. See Figure 2 for an illustration.

▶ **Theorem 5.** *A graph G is a trade conflict graph if and only if G is a bicolored plane domination graph.*

Moreover, given a set of trades $\mathcal{T}$, a bicolored plane domination model for $G(\mathcal{T})$ can be found in time $O(n)$.

Proof. Let $\mathcal{T}$ be a set of trades and let $G := G(\mathcal{T})$ be its trade conflict graph. Let $\mathcal{T}^+$ be the set of trades of $\mathcal{T}$ that are won at a positive price, and let $\mathcal{T}^- = \mathcal{T} \setminus \mathcal{T}^+$. For each $T = (w, \ell, f) \in \mathcal{T}$, assign the corresponding colored point of T to

$$(x_T, y_T, c_T) = \begin{cases} (\ell, w, green) & \text{if } T \in \mathcal{T}^+ \\ (w, \ell, red) & \text{if } T \in \mathcal{T}^- \end{cases}$$

This model is illustrated in Figure 2. Now consider two distinct trades $T_1 = (w_1, \ell_1, f_1)$ and $T_2 = (w_2, \ell_2, f_2)$ of $\mathcal{T}$. First assume that T_1 and T_2 are incompatible. Then by Lemma 1, $sgn(w_1) \neq sgn(w_2)$. Assume without loss of generality that T_1 is in $\mathcal{T}^+$, and thus that T_2 is in $\mathcal{T}^-$. By incompatiblity, $|\ell_1| \leq |w_2|$. Since $w_2 < 0$, it follows that $\ell_1 \geq w_2$. Similarly, $|\ell_2| \leq |w_1|$ and, since $w_1 > 0$, we have $w_1 \geq \ell_2$. Therefore, the point $(\ell_1, w_1, green)$ corresponding to T_1 dominates the point (w_2, ℓ_2, red) corresponding to T_2, and the model correctly puts an edge between T_1 and T_2.

Assume that T_1 and T_2 are compatible. If $sgn(w_1) = sgn(w_2)$, then the points corresponding to T_1 and T_2 are either both green and both red and the model correctly puts a non-edge between T_1 and T_2. We may thus assume without loss of generality that $T_1 \in \mathcal{T}^+$ and

$T_2 \in \mathcal{T}^-$. By compatibility, one of $|\ell_1| > |w_2|$ or $|\ell_2| > |w_1|$ holds. Suppose that $|\ell_1| > |w_2|$. Since ℓ_1 is negative, we have $\ell_1 < w_2$, which means that the green point $(\ell_1, w_1, green)$ does not dominate the red point (w_2, ℓ_2, red). So suppose that $|\ell_2| > |w_1|$. Since ℓ_2 is positive, we have $\ell_2 > w_1$, and again $(\ell_1, w_1, green)$ does not dominate (w_2, ℓ_2, red). In either case, T_1 and T_2 do not share an edge according to the model. We deduce that G is a bicolored plane domination graph using the model described above.

Note that constructing the above set of colored points does not require constructing $G(\mathcal{T})$. It only requires looping through $\mathcal{T}$ and outputting each point in $O(1)$ time, which takes total time $O(n)$. This proves the second part of the theorem statement.

Now consider the converse direction of the equivalence. Assume that G is a bicolored plane domination graph. We show that there exists a set of trades $\mathcal{T}$ such that G is isomorphic to $G(\mathcal{T})$. Take any model for G and, for $v \in V(G)$, denote by (x_v, y_v, c_v) the colored point assigned to v. Notice that bicolored plane domination models are robust to translation. That is, if we translate every point by the same vector, the domination relationships remain unchanged. Using this, we can assume that every point is in the upper left quadrant of the plane, i.e. that every point (x_v, y_v, c_v) satisfies $x_v < 0$ and $y_v > 0$ (which can be done by translating to the left and upwards by large enough amounts). For each $v \in V(G)$ such that $c_v = green$, add to $\mathcal{T}$ the trade (y_v, x_v, f_v), where f_v is arbitrary for our purposes. Then for each $w \in V(G)$ such that $c_w = red$, add to $\mathcal{T}$ the trade (x_w, y_w, f_w) again with arbitrary f_w.

We show that dominating pairs of green-red points coincide with incompatible trades. Consider a green point $(x_v, y_v, green)$ that dominates a red point (x_w, y_w, red). Then the corresponding trades (y_v, x_v, f_v) and (x_w, y_w, f_w) are incompatible because $sgn(y_v) \neq sgn(x_w)$ and $y_w \leq y_v$ by domination, which implies $|y_w| \leq |y_v|$ since the y's are positive, and because $x_w \leq x_v$ by domination, which implies $|x_v| \leq |x_w|$ since the x's are negative. Next, consider two incompatible trades T_1, T_2 of $\mathcal{T}$. They must be of the form $T_1 = (y_v, x_v, f_v)$ and $T_2 = (x_w, y_w, f_w)$. By our construction, the corresponding points are $(x_v, y_v, green)$ and (x_w, y_w, red). By incompatibility, $|x_v| \leq |x_w|$ and $|y_w| \leq |y_v|$. Since all the y's are positive and all the x's are negative, this implies $x_w \leq x_v$ and $y_w \leq y_v$ and thus $(x_v, y_v, green)$ dominates (x_w, y_w, red). It follows that $G(\mathcal{T})$ has exactly the same edges as G. ◀

The above equivalence arguably simplifies our trading framework, but it is not immediately convenient to design algorithms using a generic bicolored plane domination model. We proceed to show that any such model can be transformed into an equivalent one on a discrete grid in which each column has exactly one point, and each row has exactly one point, without gaps. This will allow us to design a relatively simple dynamic programming algorithm.

We say that a multiset of n colored points P *has the permutation matrix property* if $\{x : (x, y, c) \in P\} = \{y : (x, y, c) \in P\} = [n]$. In other words, each point of P has a distinct x coordinate in $[n]$, and a distinct y coordinate in $[n]$. The property is named after the fact that if an $n \times n$ matrix is filled with 0s, but that we put at 1 in each (x, y) that occurs in P, then we have a permutation matrix.

▶ **Lemma 6.** *Let P be a multiset of n colored points representing a bicolored plane domination model of some graph G. Then in integer sorting time, one can construct from P another bicolored plane domination model P^* for G such that P^* has the permutation matrix property.*

Proof. Suppose that P does not already have the permutation matrix property. We will assume that the points of P are in the upper-right quadrant of the plane, i.e. that each (x, y, c) satisfies $x > 0$ and $y > 0$. This can be achieved through translation as we did previously.

We first argue that we can find another model P' for G in the same quadrant such that each x-coordinate is distinct and each y-coordinate is distinct. First, multiply the x coordinate of each point of P by n, and notice that we obtain a model of the same graph since this does not alter domination relationships. Then, sort the points of P in ascending lexicographic order, where *red* comes before *green*. That is, points are of the form (x, y, c) and we sort by x coordinates first, then by y, and then c to break ties (again, note that if there are points that have the same x and y coordinates, those that are *red* occur before the *green*). Traverse the points using this order and suppose that, for some x, we encounter several points $(x, y_1, c_1), \ldots, (x, y_k, c_k)$ with the same x-coordinate, in this sorted order. Replace them by the points $(x, y_1, c_1), (x + 1, y_2, c_2), \ldots, (x + k - 1, y_k, c_k)$. Because we previously multiplied all coordinates by n, these points now all have an x-coordinate not shared with any other point. Moreover, one can check that our sorting criteria ensure that domination relationships are unchanged (even for green-red pairs having the same x and y positions since the green is moved to the right of the red). After traversing all the points in this fashion, all x values are distinct. Note that none of the y values have changed in this process. We can thus repeat the same idea with the y-coordinates, resulting in another model P' for G with no repeated x nor y values. This requires sorting time, plus one pass through the points in time $O(n)$.

To get the permutation matrix property, it suffices to "squish" all the x's and the y's in $[n]$. That is, sort P' in ascending x-coordinates and let $(x_1, y_1, c_1), \ldots, (x_n, y_n, c_n)$ be the obtained ordering (there are no ties this time). Replace the points by $(1, y_1, c_1), (2, y_2, c_2), \ldots, (n, y_n, c_n)$, keeping the colors the same, and note that again, the domination relationships are unchanged. This takes integer sorting time, assuming we have to sort again. We can repeat with the y's and obtain an equivalent model with the permutation matrix property in total integer sorting time. ◀

We now proceed to finding a maximum weight independent set. Let (G, h) be a weighted bicolored plane domination graph with a model P that has the permutation matrix property. For simplicity, we will denote the vertices of G by their points, so that $V(G) = P$. For $i, j \in [n]$, denote by

$$V(i, j) = \{(x, y, c) \in V(G) : 1 \le x \le i \text{ and } j \le y \le n\}$$

In other words, $V(i, j)$ is the set of vertices that correspond to points located in the box with corners $(1, n)$ and (i, j). We will denote by $I(i, j)$ the maximum weight of an independent set of $G[V(i, j)]$ with respect to the h weights. The maximum weight of an independent set of G is $I(n, 1)$, the value we are ultimately looking for.

Since P has the permutation matrix property, for $j \in [n]$, we will denote by (x_j, j, c_j) the unique point of P whose y-coordinate is j. One last notation: for $i_1, i_2, j \in [n]$, denote by $R(i_1, i_2, j) = \{(x, y, c) \in P : i_1 \le x \le i_2 \text{ and } j \le y \le n \text{ and } c = red\}$. See Figure 3 for an illustration.

For i, j such that $i \notin [n]$ or $j \notin [n]$, the value of $I(i, j)$ is 0. Otherwise, we can use the following recurrence:

$$I(i, j) = \begin{cases} I(i, j + 1) & \text{if } x_j > i \\ h((x_j, j, c_j)) + I(i, j + 1) & \text{if } x_j \le i \text{ and } c_j \text{ is green} \\ \max(I(i, j + 1), h(R(x_j, i, j)) + I(x_j - 1, j + 1)) & \text{if } x_j \le i \text{ and } c_j \text{ is red} \end{cases}$$

▶ **Lemma 7.** *The above recurrence for $I(i, j)$ correctly represents the maximum weight of an independent set of $G[V(i, j)]$.*
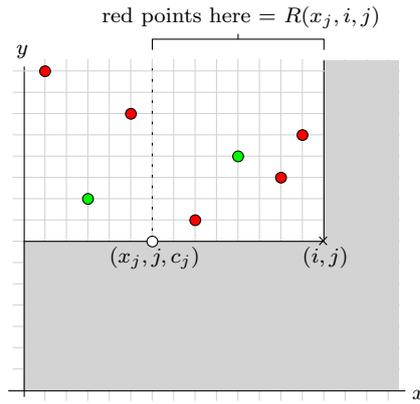
Figure 3 Illustration of the main components of the $I(i,j)$ recurrence. Points in the grayed area are ignored and only the points of $V(i,j)$ are shown. Here, (x_j, i, c_j) is the unique point with y-coordinate j, which could be red or green. The $R(x_j, i, j)$ set consists of the three red points in the area shown.

Proof. This can be shown inductively on j in reverse order. As a base case, notice that for $j > n$ and any i, then $I(i,j) = 0$ is correct since $V(i,j)$ is empty. Assume that $i, j \in [n]$ and that for any $j' > j$ and any i', the recurrence for $I(i', j')$ is correct. Let W be an independent set of $G[V(i,j)]$ of maximum weight. Consider the point (x_j, j, c_j) of P. If $x_j > i$, then point (x_j, j, c_j) is not in $V(i,j)$ and W does not contain it. Thus all points of W are in $V(i, j+1)$ and $I(i,j) = I(i, j+1) = h(W)$ follows by induction. We may assume that $x_j \leq i$. Suppose that (x_j, j, c_j) is green. By the permutation matrix property, (x_j, j, c_j) does not dominate any point in $V(i,j)$. Hence, (x_j, j, c_j) is an isolated vertex of $G[V(i,j)]$ and we may assume that $(x_j, j, c_j) \in W$ since all weights are positive. All other points of W must be in $V(i, j+1)$ and by induction $h(W \setminus \{(x_j, j, c_j)\}) = I(i, j+1)$. If follows that $I(i,j) = h((x_j, j, c_j)) + I(i, j+1) = h(W)$ is correct.

Finally, suppose that (x_j, j, c_j) is red. We first show that $I(i,j) \geq h(W)$ and deal with the converse bound after. There are two cases: either $(x_j, j, c_j) \in W$, or not. If not, then all points of W are in $V(i, j+1)$ and $h(W) = I(i, j+1)$. But if $(x_j, j, c_j) \in W$, then no green point (x_k, k, c_k) of $V(i,j)$ such that $x_k > x_j$ can be in W, since any such point dominates (x_j, j, c_j). Therefore, we may assume that every *red* point (x_k, k, c_k) of $V(i,j)$ with $x_k > x_j$ is in W. These points are $R(x_j, i, j)$, and so W consists of $R(x_j, i, j)$ plus a maximum weight independent set of $V(x_j - 1, j + 1)$ of weight $I(x_j - 1, j + 1)$. In either case, since $I(i,j)$ takes the maximum of the two possibilities, we get that $I(i,j) \geq h(W)$. For the converse bound, it is not hard to see that there exists an independent set of weight $I(i, j+1)$ in $V(i,j)$, and an independent set of weight $h(R(x_j, i, j)) + I(x_j - 1, j + 1)$ in $V(i,j)$ (the latter obtained by taking all red points in $R(x_j, i, j)$ and an optimal independent set in $V(x_j - 1, j + 1)$, noticing that none of the points from that set can dominate $R(x_j, i, j)$). The weight of W is at least the maximum of those, and so $I(i,j) \leq h(W)$. This concludes the proof. ◀

Algorithm 1 summarizes the above ideas and computes a maximum weight independent set in a bicolored plane domination graph in $O(n^2)$ time, as we argue below.

The algorithm simply computes $I(i,j)$ for every $j \in [n]$ in reverse order and for every $i \in [n]$ in order. There are $O(n^2)$ values of $I(i,j)$ to compute and each can take time $O(1)$, assuming that each $h(R(x_j, i, j))$ can be accessed in constant time. This can be achieved by storing all the relevant R values during a preprocessing step. Notice that the only relevant

Algorithm 1 How to abuse traders: a geometric interpretation.

---

function *abuseTraders(𝒯)*
  Compute the bicolored plane domination model P for $G(\mathcal{T})$ (Theorem 5)
  Compute a model P^* with the permutation matrix property (Lemma 6)
  computeRweights(P^*)
  for $j = n$ *down to* 1 **do**
    for $i = 1$ *to* n **do**
      Compute $I(i, j)$ using the recurrence
    end
  end
  return $I(1, n)$
function *computeRweights(P^*)*
  for $j = 1$ *to* n **do**
    Let (x_j, j, c_j) be the unique point of P^* with y-coordinate j
    $weight = 0$
    for $i = x_j$ *to* n **do**
      Let (i, y_i, c_i) be the unique point of P^* with x-coordinate i
      if $c_i = red$ *and* $y_i > j$ **then**
        $weight = weight + h((i, y_i, c_i))$
      $h(R(x_j, i, j)) = weight$
    end
  end

---

values to store for the recurrence have the form $R(x_j, i, j)$, where x_j is determined by j. Hence, for each x_j and j pair, we can compute $h(R(x_j, i, j))$ for each i in increasing order of i, adding for each i the weight of the new point with x-coordinate i if relevant. For each j, we need linear time to compute the necessary $h(R(x_j, i, j))$ values, so this preprocessing is done in time $O(n^2)$. All the above ideas are shown in Algorithm 1. We note that this algorithm only returns the maximum weight, but a standard backtracking procedure can find an actual set of optimal winning trades, which in turn can be converted to a price movement. We have argued the following.

▶ **Theorem 8.** *A maximum weight independent set of a bicolored plane domination graph can be found in time $O(n^2)$.*

3.2 Some open questions on trade conflict graphs

If we abstract away the financial motivations of this work for a moment, trade conflict graphs (i.e. bicolored plane domination graphs) can be studied from a purely graph theoretical perspective. One could ask whether such graphs are easy to recognize, and whether trade conflict graphs coincide, or at least share some similarity with a graph class that is already known. Although we reserve these questions for future work, we can provide a partial answer to the latter. A graph is *chordal bipartite* if it is bipartite and contains no induced cycle with at least 6 vertices.

▶ **Proposition 9.** *All bicolored plane domination graphs are chordal bipartite.*

Proof. We already know that G is bipartite, so we only need to argue that it has no induced cycle of length 6 or more. Let G be a bicolored plane domination graph and let P be a set of colored points that is a model for G. Assume that G contains an induced cycle
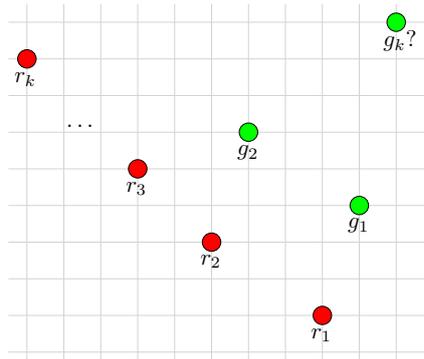
■ **Figure 4** An illustration of how some of the relative locations of the r_i and g_i points are forced.

$r_1 g_1 r_2 g_2 \ldots r_k g_k r_1$ with $k \geq 3$. Suppose without loss of generality that the r_i's correspond to red points and the g_i's to green points in P (otherwise, start the cycle at g_1 and rename accordingly). We will denote by $r_i.x$ and $r_i.y$ the coordinates of the point corresponding to r_i in P, and use the same notation $g_i.x, g_i.y$ for g_i, where $i \in [k]$.

Observe that for any $i \in [k]$, it is not possible that $r_i.x \geq r_{i+1}.x$ and $r_i.y \geq r_{i+1}.y$ both hold (where r_{k+1} is taken to be r_1). Indeed, this would imply that any green point that dominates r_i would also dominate r_{i+1}, and we know that g_{i-1} dominates r_i but not r_{i+1} (note that this is where we need the cycle to be of length at least 6). By a symmetric argument, we cannot have $r_{i+1}.x \geq r_i.x$ and $r_{i+1}.y \geq r_i.y$ simultaneously.

The rest of the following argument is illustrated in Figure 4. Suppose without loss of generality that $r_2.x \leq r_1.x$ (otherwise, start the cycle at r_2 and list the vertices in reverse order by renaming accordingly). We then deduce from the above observation that $r_2.y > r_1.y$. Moreover, g_2 dominates r_2 but not r_1, which means that $g_2.y \geq r_2.y > r_1.y$ but $g_2.x < r_1.x \leq g_1.x$. Now consider the location of r_3, which is dominated by g_2. We must have $r_3.x \leq g_2.x \leq g_1.x$ and, since g_1 does not dominate r_3, we must have $r_3.y > g_1.y \geq r_2.y$. Again using our previous observation, we get that $r_3.x \leq r_2.x$.

In other words we have deduced from $r_2.x \leq r_1.x$ and $r_2.y > r_1.y$ that $r_3.x \leq r_2.x$ and $r_3.y > r_2.y$. This argument can be used inductively to infer that $r_k.x \leq r_{k-1}.x \leq \ldots \leq r_1.x$ and $r_k.y > r_{k-1}.y > \ldots > r_1.y$. Now consider the point g_k, which dominates both r_k and r_1. One can see that such a point must dominate each point in $r_1, r_2, \ldots, r_k$, a contradiction. Thus the cycle cannot exist.   ◀

Let us mention that chordal bipartite graphs are known to admit $O(|V| + |E|)$ time algorithms for the maximum *unweighted* independent set problem, assuming that an ordering of the vertices called a *strong ordering* is given [6]. However, it is not clear whether the same time bound can be achieved for the weighted version. In any case, the geometric perspective may lead to even better algorithms in the future. We close this section with some question that this leads us to.

1. Is the class of bicolored plane domination graphs equal to the class of chordal bipartite graphs?
2. If not, can bicolored plane domination graphs be characterized as bipartite graphs that forbid induced cycles of length at least 6, plus a finite set of forbidden induced subgraphs?
3. Given a graph G, can one decide in polynomial time whether G is a bicolored plane domination graph?

4. Given a bicolored plane domination model P that has the permutation matrix property, can a maximum weight independent set be found in time $O(n)$ (regardless of the number of edges of the underlying graph)?

A positive answer to (1) would be a purely accidental find, and would be surprising since bicolored plane domination graphs appear to have a simple structure. However, it seems hard to construct a chordal bipartite graphs that can be shown to not be a trade conflict graph. This would probably give clues on (2) though. As for (3), trade conflict graphs are probably polynomial-time recognizable, given their simplicity, but this requires a more careful look. As for (4), it is plausible that $o(n^2)$ could be achieved, since our recurrence needs to go through lots of (i, j) locations that contain no point, and only n such locations appear truly relevant.

4   Online trades

We now allow the trader to interrupt the broker's price manipulation at any time by adding a new trade or closing an open trade. One may ask whether, given this new power, the trader can make money. It does appear that a small proportion of real-life traders makes profit, so perhaps the broker can be beaten. On the other hand, if there was a strategy for the trader that guarantees profit even against conspiratorial prices, someone would have figured it out by now. But surprisingly, there doesn't seem to be a clear answer in the literature. One impossibility result is given by the *efficient market hypothesis* (EMH) [7], which, roughly speaking, states that if prices perfectly reflect their environment, then no strategy can win *consistently*, i.e. against every price movement (note that the author is far from being an economist and that the actual theory is much deeper). In the same vein, assuming the price is a random walk, the best trader strategy has expected profit 0, so there exists a price movement with no win for the trader (unless the trader has infinite funds, in which case a Martingale betting system can be used for profit). Thus the broker can "mimic" a worst-case EMH-driven price or a random walk to guarantee that, on expectation, traders make no money. The broker can still lose money if unlucky, raising the question of whether there is a strategy for the broker that guarantees profitability. To our knowledge, none of aforementioned frameworks is applicable to our model.

4.1   The online model

In the online setting, a trade can be opened at any current price, unlike the previous section where the trade opening price was always assumed to be 0. Moreover, a trader could decide to close a trade at any moment, not only when w or ℓ is reached. To model this, an *online trade T* will be seen as a quadruple (w, ℓ, p, f) where w and ℓ are the winning and losing prices, respectively, $p \in \mathbb{Z}$ is the opening price and $f : [\min(w, \ell), \max(w, \ell)] \to \mathbb{R}$ gives the profit the broker makes at any possible closing price. We require that $f(p) = 0$, and that one of the following holds:

- $\ell < p < w$, in which case $f(q) > 0$ for each $q > p$ and $f(q) < 0$ for each $q < p$ (*up*); or
- $w < p < \ell$, in which case $f(q) < 0$ for each $q > p$ and $f(q) > 0$ for each $q < p$ (*down*).

Note that in a real-life setting, the trader does not have to communicate the desired limit prices w and ℓ. However in our model, this would allow the trader to leave losing trades open forever. Moreover, a limit on losses always exists, since brokers will never allow the trader's asset values to go below the account equity. In other words, the broker can always set a worst-case w and ℓ value if not given by the trader.

This online model can then be seen as a two-player game as follows. The game alternates turns between the trader and the broker, with the trader starting. At any turn i, there is a current price p_i and a set of open online trades $\mathcal{T}_i$. On the first turn $i = 1$, the price is $p_1 = 0$ and $\mathcal{T}_1 = \emptyset$.

On the i-th turn, the trader acts first and can apply the following actions any number of times:

- add an online trade $T = (w, \ell, p_i, f)$ to $\mathcal{T}_i$;
- close any open trade $T = (w, \ell, p, f)$ in $\mathcal{T}_i$. This has the effect of removing T from $\mathcal{T}_i$ and adding an amount of $f(p_i)$ to the broker's profit.

We say that the trader is *passive* if the only action used by the trader is to add trades (never closing them unless w or ℓ is reached). When the trader is done, the broker looks at $\mathcal{T}_i$ and p_i, and acts by choosing one of the two following actions:

- move the price up by one unit, so that the price for the next turn becomes $p_{i+1} = p_i + 1$;
- move the price down by one unit, so that the price for the next turn becomes $p_{i+1} = p_i - 1$.

Finally, we impose three restrictions on the broker:

R1. if $\mathcal{T}_i \neq \emptyset$, the broker must eventually close a trade if the trader does nothing;

R2. if $\mathcal{T}_i = \emptyset$ on the broker's turn, the broker moves towards price 0;

R3. the broker's decision is based solely on the current open trades $\mathcal{T}_i$ and the price p_i, and not on the past closed trades.

These can be justified as follows. R1 ensures constant activity, as it prevents the broker from zig-zagging in the same price area indefinitely. R2 states that when no trade is active, the game can be reset and the current price p_i might as well be interpreted as 0. R3 is akin to treating the broker as a Markov chain, as it makes sure that it always acts in the same manner when facing the same trade state. Another way to justify this restriction is that the broker may assume optimal play from the trader, in which case the same move should always be preferred in a given configuration. In real-life, traders are imperfect and brokers might profit from analyzing their past behavior, but this is beyond the scope of this work.

After the broker's turn, any trade $T = (w, \ell, p, f)$ in $\mathcal{T}_i$ such that $p_{i+1} \in \{w, \ell\}$ becomes closed. In this case, T is removed from $\mathcal{T}_i$ and a profit of $f(p_{i+1})$ is added to the broker's profit. The updated $\mathcal{T}_i$ becomes $\mathcal{T}_{i+1}$. We also restrict $\mathcal{T}_i$ to be finite at any turn, meaning the trader must satisfy the requirement that there exists $t \in \mathbb{N}$ such that, regardless of the broker's strategy, for each turn i we have $|\mathcal{T}_i| \leq t$.

It is not hard to show that if we impose no restriction on f, then the trader wins effortlessly. For instance on turn 1 at price 0, the trader can simply open two trades $T_1 = (1, -1, 0, f_1)$ and $T_2 = (-1, 1, 0, f_2)$ such that $f_1(1) = f_2(-1) = -2$ and $f_1(-1) = f_2(1) = 1$. Whichever direction the broker chooses, a profit of -1 will be incurred and both trades will be closed. By R2, the trader can then wait until price 0 is reached. The game resets and by R3, the trader can repeat this forever, forcing a profit of $-\infty$.

For this reason, we will require trades to be linear, as defined below.

▶ **Definition 10.** *An online trade $T = (w, \ell, p, f)$ is* linear *if there exists a positive rational number $\delta \in \mathbb{Q}_{>0}$ such that:*

- *if $\ell < p < w$, then $f(q) = \delta \cdot (q - p)$ for all $\ell \leq q \leq w$;*
- *if $w < p < \ell$, then $f(q) = \delta \cdot (p - q)$ for all $w \leq q \leq \ell$.*

Note that real-life trading is done on linear trades and δ is sometimes called the *lot size*.

4.2 The only good online broker strategy

Perhaps the first strategy that comes to mind is to use the results from the previous section in the online setting. That is, on the broker's i-th turn, we look at $\mathcal{T}_i$, translate the trades so that the current price can be interpreted as 0, and compute an optimal price trajectory M using a maximum weight independent set on $G(\mathcal{T}_i)$. We then go in the same direction as M for one price unit, and this M will be recalculated on the $(i+1)$-th turn. Let us call this the *offline strategy*. Unfortunately, this is not a good strategy.

▶ **Proposition 11.** *If the broker uses the offline strategy, then the trader can force the broker to have an infinite negative profit, even if the trader is passive and restricted to linear trades.*

Proof. The trader's strategy is illustrated in Figure 5.
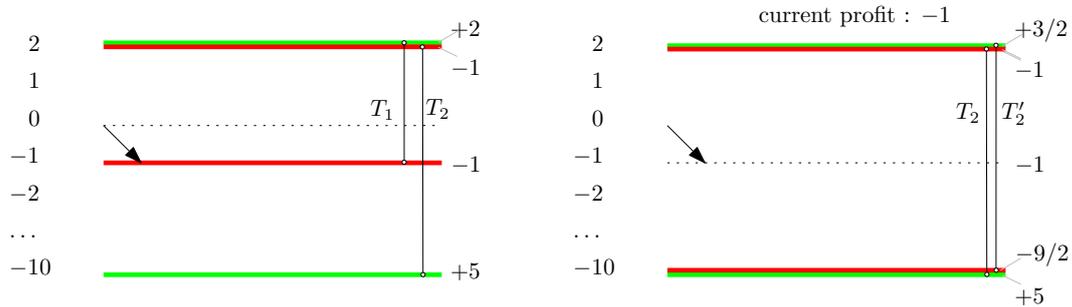
Figure 5 The offline strategy is not good.

On turn 1 at price $p_1 = 0$, the trader opens a linear trade $T_1 = (2, -1, 0, f)$ with $f(-1) = -1, f(1) = 1, f(2) = 2$, and another trade $T_2 = (-10, 2, 0, g)$ such that $g(-10) = 5$ and $g(2) = -1$ (and the other values made linear with $\delta = 1/2$). These trades are incompatible, so the broker using the offline strategy will choose to lose T_1 and win T_2, and will thus go down on its first turn. The price will become -1 and the T_1 trade will be lost, incurring a (negative) profit of -1 for the broker.

On turn 2 at price $p_2 = -1$, the passive trader opens the trade $T_2' = (2, -10, -1, g')$ with $g'(-10) = -9/2$ and $g'(2) = 3/2$ (linear with $\delta = 1/2$). At this point the trader only waits until T_2 and T_2' are closed, which must happen by R1. If price 2 is reached first, both trades close and the broker's profit for them is $-1 + 3/2 = 1/2$. If price -10 is reached first instead, then the profit is $5 - 9/2 = 1/2$. In either case, the broker's total profit is $-1 + 1/2 = -1/2$. At this point, all trades are closed. By R2 and R3, the trader can repeat this pattern indefinitely. ◀

The above is actually part of a more general result that says that any strategy that does not go for the maximum *potential* from the start loses infinitely. The interest of Proposition 11 is that it serves as a concrete example of losing to a concrete strategy. We next show that in fact, maximizing potential profits is essentially the only good strategy. Anything that diverges from it is either suboptimal or has infinite negative profit.

For a set of online trades $\mathcal{T}$ and a price p, the *potential profit* of p on $\mathcal{T}$ is defined as

$$potent(\mathcal{T}, p) = \sum_{(w, \ell, q, f) \in \mathcal{T}} f(p)$$

which represent the profit that would be made if every trade closed at price p.

The *maximum potential* strategy is defined as follows. On the broker's i-th turn,

- if $potent(\mathcal{T}_i, p_i + 1) \geq potent(\mathcal{T}_i, p_i - 1)$, then move the price *up*;
- otherwise if $potent(\mathcal{T}_i, p_i + 1) < potent(\mathcal{T}_i, p_i - 1)$, then move the price *down*.

Intuitively speaking, this strategy is based on the fact that the trader either has more buys than sells or vice-versa. Under linear trades, one of the direction has to be worse than the other (unless they both achieve profit 0) and we simply go in the direction that is worse for the trader. Do note that this strategy can change direction even if the trader does nothing, since winning a trade can change the direction of maximum potential.

We now analyze this strategy against others, but from a general perspective where the broker starts applying it after a certain number of turns i.

▶ **Theorem 12.** *Suppose that the trader is restricted to linear online trades. Let $\mathcal{T}_i$ be a set of open trades at the start of the i-th turn, let p_i be the current price, and let $profit(i)$ be the total profit of the broker at the start of turn i. Then the following holds:*

1. *if $profit(i) + potent(\mathcal{T}_i, p_i) \geq 0$, then if the broker applies the maximum potential strategy from turn i and onwards, it achieves a total profit of at least $profit(i) + potent(\mathcal{T}_i, p_i)$ against any trader. Moreover, this is the maximum possible profit achieved against a trader with optimal play, passive or not;*
2. *if the broker does not move the price in the direction of maximum potential profit on turn i, then the broker makes a profit that is strictly less than $profit(i) + potent(\mathcal{T}_i, p_i)$ against a trader with optimal play, passive or not;*
3. *if $profit(i) + potent(\mathcal{T}_i, p_i) < 0$, then the broker incurs an infinite negative profit against a trader with optimal play, passive or not.*

Proof. Let us first consider the first part of (1). Let $\mathcal{T}_i'$ be the set of trades after the trader has finished the i-th turn, and let $profit'(i)$ be the profit of the broker at this point. Note that if the trader is passive, then $\mathcal{T}_i \subseteq \mathcal{T}_i'$, but if it is not passive, a certain set of trades might be closed. In this case, for each $(w, \ell, q, f) \in \mathcal{T}_i \setminus \mathcal{T}_i'$, a profit of $f(p_i)$ is added for the broker and an amount of $f(p_i)$ is removed from the potential of p_i. Also, recall that for each new trade $(w, \ell, q, f) \in \mathcal{T}_i' \setminus \mathcal{T}_i$, we have $f(p_i) = 0$. It follows that

$$profit(i) + potent(\mathcal{T}_i, p_i) = profit'(i) + potent(\mathcal{T}_i', p_i)$$

In other words, the trader's turn does not affect the potential profit of the broker. We next argue that one of the price directions does not decrease this potential. Write $\mathcal{T}_i' = \{T_1, \ldots, T_m\}$ and for $j \in [m]$, let δ_j be the linear factor affecting trade T_j. We say that $sgn(T_j) = 1$ if T_j is an *up* trade and $sgn(T_j) = -1$ if T_j is a *down* trade. Then by linearity, raising the price by one unit changes the potential of T_j by $sgn(T_j) \cdot \delta_j$ (*up* trades increase in potential, and *down* trades decrease), and lowering the price by one unit changes it by $-sgn(T_j) \cdot \delta_j$. In other words,

$$potent(\mathcal{T}_i', p_i + 1) - potent(\mathcal{T}_i', p_i) = \sum_{(w, \ell, p_j, f_j) \in \mathcal{T}_i'} (f_j(p_i + 1) - f_j(p_i)) = \sum_{T_j \in \mathcal{T}_i'} sgn(T_j) \cdot \delta_j$$

and likewise

$$potent(\mathcal{T}_i', p_i - 1) - potent(\mathcal{T}_i', p_i) = \sum_{T_j \in \mathcal{T}_i'} -sgn(T_j) \cdot \delta_j$$

One of $\sum_{T_j \in \mathcal{T}_i'} sgn(T_j) \cdot \delta_j$ or $\sum_{T_j \in \mathcal{T}_i'} -sgn(T_j) \cdot \delta_j$ is greater than or equal to 0, and so one direction leads to a potential greater than or equal to $potent(\mathcal{T}_i', p_i)$. Suppose that this direction is up. Let $profit(i+1)$ be the broker's profit after changing the price to $p_{i+1} = p_i + 1$ and let $\mathcal{T}_{i+1}$ be the set of trades still open at this point. Some trades (w, ℓ, q, f) might become closed at price $p_i + 1$ and incur a profit of $f(p_i + 1)$, but remove an amount of $f(p_i + 1)$ from $potent(\mathcal{T}_i', p_i + 1)$. That is, similarly as we did above, we obtain that

$$profit(i+1) + potent(\mathcal{T}_{i+1}, p_i + 1) = profit'(i) + potent(\mathcal{T}_i', p_i + 1)$$

which is greater than or equal to $profit(i) + potent(\mathcal{T}_i, p_i)$. The idea is the same if going down is better for the potential instead. By repeating this argument, this shows that the profit plus potential can be made to never decrease, guaranteeing the broker at least a profit of $profit(i) + potent(\mathcal{T}_i, p_i)$ when every trade is closed and the potential has reached 0. Also note that it is not hard to see that unless the trader opens new trades, the broker will keep the same direction until at least one trade is closed, as required by our model. This shows the first part of (1).

We now argue that this is what the optimal trader can achieve. If the trader is not passive, then the trader can close every trade at price p_i at the start of the i-th turn and stop playing. The broker's profit will be exactly $profit(i) + potent(\mathcal{T}_i, p_i)$, which is the best the trader can hope for since this is a lower bound. This proves (1) in the case of a non-passive trader.

We next prove a claim that will be useful for the remaining statements of the theorem that concern passive traders. Let $\mathcal{T}_j$ be any set of open trades at some current price p_j, and assume that it is the trader's turn. We claim that the passive trader can add a set of trades to $\mathcal{T}_j$ such that the broker's profit on $\mathcal{T}_j$ is forced to be exactly $potent(\mathcal{T}_j, p_j)$. This effectively simulates the action of closing all trades at current price. Let $\mathcal{T}_j^+ = \{(w, \ell, p, f) \in \mathcal{T} : w > p_j\}$ and let $\mathcal{T}_j^- = \mathcal{T}_j \setminus \mathcal{T}_j^+$. Consider $T = (w, \ell, p, f) \in \mathcal{T}_j^+$ and let δ be the linear factor affecting T. Add the corresponding linear trade $T' = (\ell, w, p_j, f')$ such that $f'(\ell) = \delta(p_j - \ell)$ and $f'(w) = \delta(p_j - w)$. Suppose that price $q \in \{w, \ell\}$ is reached first. Since the trades are linear with the same δ the profit on T and T' will be $f(q) + f'(q) = \delta(q - p) + \delta(p_j - q) = \delta(p_j - p)$. Again by linearity, this is exactly $f(p_j)$. By symmetry, one can add a similar trade T' for each $(w, \ell, p, f) \in \mathcal{T}^-$ to force a profit of $f(p_j)$ for T and T'. Because of restriction R1, the trader can wait until the broker has closed every trade. It follows that the trader can force the current trades to close for a total profit of $\sum_{(w, \ell, p, f) \in \mathcal{T}_j} f(p_j) = potent(\mathcal{T}_j, p_j)$.

This claim shows that (1) holds also for passive traders, since the trader can force a final profit of $profit(i) + potent(\mathcal{T}_i, p_i)$ at the start of the i-th turn.

Statement (2) is now easy to prove. If the broker moves against the optimal potential, then we will get $profit(i+1) + potent(\mathcal{T}_{i+1}, p_{i+1}) < profit(i) + potent(\mathcal{T}_i, p_i)$. As we have seen, the trader (passive or not) can force a final profit of $profit(i+1) + potent(\mathcal{T}_{i+1}, p_{i+1})$, which is strictly less than what the maximum potential strategy gives.

As for statement (3), suppose that the trader has done a sequence of $i - 1$ actions such that $profit(i) + potent(\mathcal{T}_i, p_i) < 0$. Again, the trader, passive or not, can force this negative profit on the broker. After the latter has closed every trade, the trader can simply repeat the same sequence of actions to make the broker lose the same amount again indefinitely (which works because of restrictions R2 and R3, which imply that the game always resets after this negative profit, and that the broker repeats the same mistake every time). ◄

As a corollary, we see that if the trader makes a move that makes the profit plus potential positive, then the broker will make profit. Conversely, if the broker makes a mistake to make this value negative, then the trader will make it lose infinitely. Therefore, the only source of

positive profit is due to a trader mistake, and the only source of negative profit is due to a broker mistake. If both players play optimally, no mistake is ever made and a profit of 0 is made. However, in real life trading, each trade opened by the trader has a fee that goes into the broker's pockets. We conclude that the best option for the trader is to not trade and try to get rich by other means.

5   Conclusion

In this work, we have provided some useful algorithmic tools for malevolent brokers to plot against traders. Our models are combinatorial and do not consider any form of stochasticity. This may serve as criticism towards our model, but one must reckon that stochastic trading models have been studied for a very long time. Our work offers a new perspective on trade analysis and shows how price movements can be optimized when given full control over them. Let us also mention that our graph theoretical view on trading is novel and allowed us to discover a potentially new graph class (although they might just be chordal bipartite graphs – the future will tell).

One future direction would be to reach a middle ground between combinatorial and stochastic by incorporating various forms of randomness into our price manipulation possibilities. For instance, we may be able to move prices in the desired direction only with certain probabilities. Also, in the online model, we often assumed a trader with optimal play, whereas a randomized trader could be considered more realistic. Finally, one can ask whether a trader knowledgeable of the broker's scheme is able to profit from that information. In the offline setting, if the trader was able to see every trade, he or she could predict the broker's movements and add appropriate winning trades. This would need to be done so that the broker would not alter its trajectory because of the new trades, leading to another algorithmic problem of interest.

References

1   How crypto investors can avoid the scam that captured $2.8 billion in 2021. time.com: `https://time.com/nextadvisor/investing/cryptocurrency/` `protect-yourself-from-crypto-pump-and-dump/`. Accessed: February 17, 2022.

2   Market makers vs. electronic communications networks. investopedia.com: `https://www.investopedia.com/articles/forex/06/ecnmarketmaker.asp`. Accessed: February 17, 2022.

3   Robinhood restricts trading in gamestop, other names involved in frenzy. cnbc.com: `https://www.cnbc.com/2021/01/28/` `robinhood-interactive-brokers-restrict-trading-in-gamestop-s.html`. Accessed: February 17, 2022.

4   Why do many forex traders lose money? here is the number 1 mistake. dailyfx.com: `https://www.dailyfx.com/forex/fundamental/article/special_report/2015/06/25/` `what-is-the-number-one-mistake-forex-traders-make.html`. Accessed: February 17, 2022.

5   Andreas Brandstädt, Van Bang Le, and Jeremy P Spinrad. *Graph classes: a survey*. SIAM, 1999.

6   Feodor F Dragan. Strongly orderable graphs a common generalization of strongly chordal and chordal bipartite graphs. *Discrete applied mathematics*, 99(1-3):427–442, 2000.

7   Eugene F Fama. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417, 1970.

8   Gianni Franceschini, Shan Muthukrishnan, and Mihai Pătraşcu. Radix sorting with no extra
        space. In *European Symposium on Algorithms*, pages 194–205. Springer, 2007.

9   Yijie Han. Deterministic sorting in o (n log log n) time and linear space. In *Proceedings of the
        thiry-fourth annual ACM symposium on Theory of computing*, pages 602–608, 2002.

10   Nicklas Larsen, Helmut Mausser, and Stanislav Uryasev. Algorithms for optimization of
        value-at-risk. In *Financial engineering, E-commerce and supply chain*, pages 19–46. Springer,
        2002.

11   James B Orlin. Max flows in O(nm) time, or better. In *Proceedings of the forty-fifth annual
        ACM symposium on Theory of computing*, pages 765–774, 2013.

12   Mechthild Stoer and Frank Wagner. A simple min-cut algorithm. *Journal of the ACM (JACM)*,
        44(4):585–591, 1997.

13   Gerold Studer. *Maximum loss for measurement of market risk.* PhD thesis, ETH Zurich, 1997.

14   Mu Yang. Fx spot trading and risk management from a market maker's perspective. Master's
        thesis, University of Waterloo, 2011.