Trading Prophets: How to Trade Multiple Stocks Optimally

Surbhi Rajput* Ashish Chiplunkar† Rohit Vaish‡

Abstract

In the (single stock) *trading prophet* problem formulated by Correa et al. [2023], an online algorithm observes a sequence of prices of a stock. At each step, the algorithm can either buy the stock by paying the current price if it doesn't already hold the stock, or it can sell the currently-held stock and collect the current price as a reward. The goal of the algorithm is to maximize its overall profit. Correa et al. [2023] showed an optimal competitive ratio of $1/2$ for this problem when the stock prices are identically and independently distributed.

In this work, we generalize the model and the results of Correa et al. [2023] by allowing the algorithm to trade multiple stocks. First, we formulate the (k, ℓ, ℓ')-Trading Prophet Problem, wherein there are k stocks in the market, and the online algorithm can hold up to ℓ stocks at any time, where $\ell \leq k$. The online algorithm competes against an offline algorithm that can hold at most $\ell' \leq \ell$ stocks at any time. Under the assumption that prices of different stocks are independent, we show that, for any ℓ, ℓ', and k, the optimum competitive ratio of the (k, ℓ, ℓ')-Trading Prophet Problem is $\min\left\{\frac{1}{2}, \frac{\ell}{k}\right\}$.

We further introduce the more general $\mathcal{M}$-Trading Prophet Problem over a matroid $\mathcal{M}$ on the set of k stocks, wherein the stock prices at any given time are possibly correlated (but are independent across time). The algorithm is allowed to hold only an independent subset of stocks at any time. We prove a tight bound of $\frac{1}{1+d}$ on the competitive ratio of the $\mathcal{M}$-Trading Prophet Problem, where d is the *density* of the matroid.

Our analysis of both problems is based on two key insights. First, any algorithm can be simulated by one that, on each time step, sells *all* its currently-held stocks before buying a suitable subset of stocks. Second, we prove that the general problem reduces to a restriction where the expected price of every stock is zero. These technical insights facilitate a simple analysis to establish tight competitive bounds on a substantially broader class of trading prophet problems than the single stock version by Correa et al. [2023].

1 Introduction

Consider a trader named Alex, who wants to invest her capital in the stock market and maximize her returns. In the stock market, prices of stocks fluctuate due to various factors including market demand, company performance, and economic indicators. Alex monitors these price shifts in real time, with the price of each stock being available as a sequence over time in an online manner. Alex would have loved to know the future behavior of stock prices, so that she could trade optimally and maximize her profit. However, the online setting, which mirrors real-world trading conditions, presents a challenge. Alex can't see the future prices, so she is not necessarily able to trade optimally. The framework of online computation and competitive analysis [Sleator and Tarjan, 1985] captures Alex's dilemma: Given the current prices and without knowing their future behavior, how should Alex decide which stocks to trade at every time step? Alex must run an *online algorithm* – one whose current output depends only on the current and the past inputs. Informally, such an algorithm is said to be *α-competitive* if it guarantees an expected payoff at least α times the optimal payoff.

If Alex is completely uninformed about future outcomes, it is impossible for *any* algorithm to give her a non-trivial competitive guarantee. Indeed, if the input is adversarial, every stock Alex buys crashes, and every stock she doesn't buy soars. Thankfully for Alex, such adversarial behavior of prices does not arise in practice. In fact, she knows in advance the probability distributions of future stock prices. This setting is referred to in the literature as the *prophet inequality* setting [Krengel and Sucheston, 1978, Hill and Kertz, 1982].[1]

*Indian Institute of Technology Delhi.
†Indian Institute of Technology Delhi.
‡Indian Institute of Technology Delhi.

[1] In the classic single-choice prophet inequality problem [Hajiaghayi et al., 2007, Lucier, 2017, Esfandiari et al., 2017, Correa et al., 2019, Peng and Tang, 2022], an online algorithm observes a sequence of random variables with known distributions. The value of a random variable is realized upon its arrival, and the algorithm should either accept it (in which case the process stops) or reject it (in which case the next variable is observed). The goal is to maximize the selected value while competing against an offline adversary, also referred to as the *prophet*, that observes all realizations at once and chooses the maximum out of them. This problem has also

The recent work of Correa et al. [2023] considered the simplest case of trading one stock and introduced the *Trading Prophet* problem. In this problem, an online algorithm is given a sequence of n prices, each sampled independently from the same probability distribution that is known to the algorithm. The algorithm trades a single (indivisible) stock and is limited to holding at most one unit of stock at any given time. At each time step, the algorithm must decide whether to *purchase* the stock at the current price (provided it does not hold the stock already) or to *sell* the stock at the current price (provided it already holds the stock). Correa et al. proved that the strategy of buying when the price falls below the *median*[2] and selling otherwise is $1/2$-competitive. Moreover, they also proved that no online strategy can achieve a competitive ratio strictly better than $1/2$. Their analysis extends to the scenario where the algorithm can trade multiple units of a single stock.

Our Contributions. While the single stock problem is instructive, it may not be representative of real-world trading scenarios. It is more natural to imagine that an investor trades more than one stock at any given time. Motivated by this, in this paper, we generalize the Trading Prophet model to accommodate the idea of holding *multiple* stocks. We believe this model better reflects the real-world trading dynamics.

Generalizing the Trading Prophet problem. We introduce the (k, ℓ)-Trading Prophet Problem in which we are given k different stocks and the algorithm can hold up to ℓ stocks (and at most one unit of any given stock; this is without loss of generality) at any time. The price of every stock at any time step is drawn independently from a known distribution. Note that the distributions of prices of the k stocks can be different. For any given time step, the prices of stocks need not be independently distributed. However, we continue with the assumption of Correa et al. [2023] that across time steps these prices are independent. At every time step, an algorithm must decide which subset of its currently owned stocks to sell, and which subset of stocks not currently owned to buy, subject to the constraint that the algorithm holds at most ℓ stocks at any given time. We establish tight bounds on the competitive ratio of the (k, ℓ)-Trading Prophet Problem. Specifically, we prove,

Theorem 1.1. *For every k and ℓ, there exists an online algorithm for the (k, ℓ)-Trading Prophet Problem that achieves a competitive ratio of $\min\left\{\frac{1}{2}, \frac{\ell}{k}\right\}$ when the stock prices at any given time are independent. Moreover, no algorithm can achieve a strictly better competitive ratio.*

Extension to resource augmentation setting. Our result in Theorem 1.1 extends to the *resource augmentation* setting [Roughgarden, 2020]. In this setting, the online algorithm competes with the offline algorithm having fewer resources than the online algorithm. Specifically, the offline algorithm is restricted to hold up to ℓ' stocks at any time, where $1 \le \ell' \le \ell$. We call this generalization the (k, ℓ, ℓ')-Trading Prophet Problem. Even in this setting, we prove that the tight bound from Theorem 1.1 continues to hold, making it a special case of the following two theorems for $\ell' = \ell$.

Theorem 1.2. *For every k, ℓ and ℓ', no algorithm for (k, ℓ, ℓ')-Trading Prophet Problem can achieve a competitive ratio greater than $\min\left\{\frac{1}{2}, \frac{\ell}{k}\right\}$ when the stock prices at any given time are independent.*

Theorem 1.3. *For every k, ℓ, and ℓ', there exists an algorithm that is $\min\left\{\frac{1}{2}, \frac{\ell}{k}\right\}$-competitive algorithm for the (k, ℓ, ℓ')-Trading Prophet Problem when the stock prices at any given time are independent.*

One might intuitively expect that the competitive ratio will improve if the adversary's power is reduced by lowering ℓ'. Somewhat surprisingly, the optimal competitive ratio established by Theorems 1.2 and 1.3 *does not* depend on ℓ'. Our instance to force the hardness result is designed so that, with high probability, at most one stock is profitable at any given time. The offline algorithm can easily spot such a stock and buy it (even when $\ell' = 1$), while the online algorithm is unlikely to guess it correctly, even though the possibility of holding up to ℓ stocks essentially gives the algorithm ℓ opportunities to guess the right stock. Another interesting takeaway is that for any fixed k, as long as the online algorithm can hold at least 50% of all stocks (i.e., $\ell \ge k/2$), increasing the number of stocks held *does not* affect the competitive ratio. We believe that these insights are not obvious a priori.

---

been studied under feasibility constraints beyond single choice; see, for example, [Alaei, 2014, Kleinberg and Weinberg, 2019, Ehsani et al., 2018].

[2]The median of the distribution of a random variable X is a number μ such that $\Pr[X < \mu] \le 1/2$ and $\Pr[X > \mu] \le 1/2$.

Generalization to correlated prices. For the (k, ℓ)-Trading Prophet Problem, we assumed that the prices are independent across time and independent across stocks in Theorem 1.1. We generalize this model to deal with *correlated* stock prices to better relate with real-world trading scenarios. We solve for the setting where the stock prices are possibly correlated across stocks for any given time step, but independent across time. For this correlated setting of the (k, ℓ)-Trading Prophet Problem, we prove the tight bound of $\frac{\ell}{\ell+k}$ (Corollary 3.1 and Corollary 4.1) as an easy consequence of Theorems 1.4 and 1.5 stated below.

Generalization to matroid settings. The constraint of holding up to ℓ stocks at any given time is not sufficient to capture traders' requirement to keep their portfolios sufficiently diversified. For instance, suppose the stocks are categorized into different types. To maintain diversity, a trader wants to hold at most ℓ_i stocks of the i'th type in addition to holding at most ℓ stocks, at all times. Such requirements are readily captured by combinatorial structures called *matroids* (refer Definition 2.1). Motivated by this, we consider the substantial generalization of the trading prophet problem to matroid constraints. We introduce the $\mathcal{M}$-Trading Prophet Problem, over the matroid $\mathcal{M}$ on the set of stocks. Here, the trader is allowed to hold a subset of stocks provided that the subset is an *independent set* of $\mathcal{M}$. We establish that the competitive ratio of the trading prophet problem on a matroid is determined by a property of the matroid called its *density*. Specifically, we prove,

Theorem 1.4. *Let $\mathcal{M}$ be an arbitrary matroid and d be the density of $\mathcal{M}$. Then there does not exist an algorithm for $\mathcal{M}$-*Trading Prophet Problem *whose competitive ratio is greater than $\frac{1}{1+d}$.*

Theorem 1.5. *Let $\mathcal{M}$ be an arbitrary matroid and d be the density of $\mathcal{M}$. Then there exists an algorithm for the $\mathcal{M}$-*Trading Prophet Problem *whose competitive ratio is $\frac{1}{1+d}$.*

Insights and techniques. We make the following crucial observations which help us get a much simpler analysis of Correa et al. [2023]'s result and provide a convenient way to analyze all our generalizations. The decision of opting to not sell a previously held stock at a given time step equates to selling the stock and repurchasing it at the same price and at the same time step, leaving the net profit unchanged. This simple yet crucial observation enables us to express the net profit of the algorithm over n time steps as the sum of $(n-1)$ i.i.d. random variables. This reduces our analysis to relating the expected *per-time-step* profit of the online and offline algorithms. It also enables us to identify the expected-profit-maximizing online algorithm (see Algorithm 2.2). Another important insight of our analysis is that in order to establish competitive guarantees, it suffices to consider instances where the expected value of each stock's price is zero, further simplifying our analysis.

2 Preliminaries

For any positive integer n, let $[n]$ denote the set $\{1, 2, \ldots, n\}$.

Definition 2.1. (Matroid) *A **matroid**[3] $\mathcal{M}$ is a pair $(E, \mathcal{I})$, where E is a finite set, called the **ground set**, and $\mathcal{I}$ is a family of subsets of E, called the set of **independent sets**, with the following properties:*

- *The empty set is independent, i.e., $\emptyset \in \mathcal{I}$.*
- *Every subset of an independent set is independent, i.e., $\mathcal{I}$ is downward-closed.*
- *If I and J are independent sets and $|I| > |J|$, there exists an element $e \in I \setminus J$ such that $J \cup \{e\}$ is independent.*

We say a matroid $\mathcal{M}$ is

- ***loopless** if every singleton subset of E is independent, i.e., for every $e \in E$, $\{e\} \in \mathcal{I}$ (an element e is called a **loop** if $\{e\} \notin \mathcal{I}$).*
- *the r-**uniform matroid** over E if for every set $A \subseteq E$, A is independent if and only if $|A| \leq r$.*

Definition 2.2. (Rank, Rank function, Span, Density) *The **rank** of a matroid is the size of the largest independent set in the matroid. The **rank function** of a matroid $\mathcal{M}$ is a function $\boldsymbol{rk} : 2^E \to \mathbb{N}$ that maps each*

[3] For a comprehensive discussion on matroids, we refer the reader to [Oxley, 1992].

subset $S \subseteq E$ to the maximum size of an independent subset of S. An element e is said to be ***spanned*** *by a set $S \subseteq E$ in a matroid $\mathcal{M}$ if the rank of S is equal to the rank of $S \cup \{e\}$.*

The ***density*** *d of a loopless matroid $\mathcal{M}$ is defined as the maximum value of the ratio $|X|/\boldsymbol{rk}(X)$ over all non-empty subsets X of the ground set E [Soto, 2013]. Formally,*

$$d = \max_{\emptyset \neq X \subseteq E} \frac{|X|}{\boldsymbol{rk}(X)}.$$

Observation 2.1. *The density of the r-uniform matroid on a set of n elements (where $r \leq n$) is n/r.*

The **maximum weight independent set problem** over a matroid $\mathcal{M} = (E, \mathcal{I})$ is defined as follows: Given a weight function $w : E \to \mathbb{R}^+$, find an independent set $I \in \mathcal{I}$ that maximizes the total weight. A simple greedy idea to solve the maximum weight independent set problem, known as Kruskal's algorithm, is given by Algorithm 2.1. The following is a folklore result.

Fact 2.1. *The* Kruskal's algorithm *stated in Algorithm 2.1 returns the maximum weight independent set over a matroid constraint.*

Algorithm 2.1. Kruskal's Algorithm

Input: A matroid $\mathcal{M} = (E, \mathcal{I})$, and a weight function $w : E \to \mathbb{R}$.
Output: The set $I \in \mathcal{I}$ that maximizes $w(I) = \sum_{e \in I} w(e)$.
Sort the elements of E in non-increasing order of weights and call them $e_1, \ldots, e_n$.
Initialize $H \leftarrow \emptyset$.
for $i = 1$ to n **do**
  if $w(e_i) < 0$ **then**
    {This implies $w(e_{i+1}), \ldots, w(e_n)$ are all negative.}
    break
  end if
  if $H \cup \{e_i\} \in \mathcal{I}$ **then**
    {Equivalently, if e_i is not spanned by $\{e_1, \ldots, e_{i-1}\}$.}
    $H \leftarrow H \cup \{e_i\}$
  end if
end for
return H

$\mathcal{M}$-Trading Prophet Problem. An instance of the $\mathcal{M}$-Trading Prophet Problem is specified by a matroid $\mathcal{M}$ on the ground set $\{1, \ldots, k\}$ for some $k \in \mathbb{N}$, the joint distribution $\mathcal{D}$ of k real-valued random variables, and a positive integer n. The integer n represents the length of the time horizon. The random variable X^t, taking values from $\mathbb{R}^k$, denotes the vector of stock prices at time $t \in [n]$. The stock prices for different time stamps are independent but the components $\{X^t_s \mid s \in [k]\}$ of each X^t are not necessarily independent. Additionally, if the components of each X^t are independent too, then we call that instance *independently distributed*. At each step $t \in [n]$, the vector X^t of prices of all stocks is announced. The random variable, X^t_s, denotes the stock price s at time step t. In response, the algorithm should decide the following at each step:

- which subset of previously held stocks should be *sold*, and
- which subset of stocks not currently held should be *bought*,

subject to the constraint that the set of stocks held by the algorithm is an independent set of $\mathcal{M}$ at all times. The algorithm starts with an empty set of stocks and at the end of the time horizon (i.e., after n time steps), the algorithm sells all the held stocks. Thus, the *profit* earned by the algorithm at any time t equals the amount received due to the sale of stocks minus the amount paid to buy stocks, and the *overall profit* is the sum of profits across all time steps $t \in [n]$. Our goal is to maximize the expected overall profit. Since an algorithm can never buy a stock which is a loop of the matroid, we assume that $\mathcal{M}$ is a loopless matroid.

To simplify our presentation, we work with an equivalent setup where, in every time step, the algorithm sells all currently-held stocks and then buys a subset of stocks S such that S is an independent set in $\mathcal{M}$ – not selling a stock generates the same profit as selling the stock at time step t and then buying the same stock at the same price at time step t.

(k,ℓ)-Trading Prophet Problem. This is the restriction of $\mathcal{M}$-Trading Prophet Problem where $\mathcal{M}$ is the ℓ-uniform matroid over a ground set of size k.

(k,ℓ,ℓ')-Trading Prophet Problem. This is the (k,ℓ)-Trading Prophet Problem extended to the resource augmentation setting. In this problem, just as in the (k,ℓ)-Trading Prophet Problem, we are given a joint distribution for $k \in \mathbb{N}$ stocks, integers $\ell,\ell' \in \mathbb{N}$, and a positive integer n. An online algorithm for the (k,ℓ,ℓ')-Trading Prophet Problem is allowed to hold at most ℓ stocks and its goal, as before, is to maximize its profit across n time steps. The online algorithm competes against an offline algorithm that is allowed to hold at most $\ell' \leq \ell$ stocks. Thus, when $\ell = \ell'$, the (k,ℓ,ℓ')-Trading Prophet Problem becomes the (k,ℓ)-Trading Prophet Problem.

An *online algorithm* must decide which subset of stocks to buy and sell at each time step based only on the history and the current prices of stocks. On the other hand, an *offline algorithm* has access to all the realizations of stock prices. In general, an online algorithm will fail to obtain as much profit as the offline optimal algorithm due to its inability to see the future trend of stock prices. We use the framework of competitive analysis [Sleator and Tarjan, 1985] to quantify the performance of an online algorithm relative to that of the optimal offline algorithm. The *competitive ratio* of an online algorithm is defined as follows.

Competitive ratio. An online algorithm is *α-competitive* if, on every instance, the expected profit of the online algorithm is at least α times the expected profit of the optimal offline algorithm.

Notation. We frequently refer to the following notation in our analysis:

- $\mathtt{top}_{\mathcal{M}}(w)$ denotes the weight of the maximum weight independent set of the matroid $\mathcal{M} = (E,\mathcal{I})$ with respect to the weight function $w : E \longrightarrow \mathbb{R}^+$, and $\mathtt{top}_{\ell}(w)$ is $\mathtt{top}_{\mathcal{M}}(w)$ when $\mathcal{M}$ is the ℓ-uniform matroid. Therefore, $\mathtt{top}_{\ell}(w)$ is equal to be the sum of the ℓ largest numbers out of $w_1,\dots,w_k$ if at least ℓ of them are positive, and the sum of all positive numbers in $w_1,\dots,w_k$ if fewer than ℓ of them are positive.
- We assume that the probability distribution $\mathcal{D}$ over $\mathbb{R}^k$ has a well-defined expectation. $\mu = (\mu_1,\dots,\mu_k)$ denotes the expectation of the probability distribution $\mathcal{D}$.
- The ordered pair (X',X) denotes two samples drawn from $\mathcal{D}$ independently. X' and X are only used in the analysis.
- The expression $(x)^+$ denotes $\max(0,x)$.

Lemma 2.1. (Optimal offline profit) *For every instance of the $\mathcal{M}$-Trading Prophet Problem, the expected overall profit of the optimal offline algorithm is*

$$(n-1)\mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(X'-X\right)\right].$$

Proof. Recall the equivalent setup where at time step t, the algorithm sells all currently-held stocks and then buys a set of stocks. Suppose we buy a subset S of stocks at time t and sell it at time $t+1$, then our net profit will be $\sum_{s\in S}(X_s^{t+1}-X_s^t)$. Therefore, at every time t, the optimal algorithm buys the maximum weight independent subset of stocks with respect to the weight function $X^{t+1}-X^t$. Thus, the resulting profit of the algorithm at time t is $\mathtt{top}_{\mathcal{M}}\left(X^{t+1}-X^t\right)$. Since the random variables X^{t+1} and X^t are two independent samples drawn from the distribution $\mathcal{D}$, the distribution of (X^{t+1},X^t) is same as (X',X). Therefore, the expected profit is $\mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(X'-X\right)\right]$. Summing over time steps $t \in [n-1]$, we get the result. □

Lemma 2.2. (Optimal online profit) *For every instance of the $\mathcal{M}$-Trading Prophet Problem, the expected overall profit of the optimal online algorithm is*

$$(n-1)\,\mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(\mu-X\right)\right].$$

Furthermore, this profit is attained by Algorithm 2.2.

Proof. Consider the execution of an arbitrary online algorithm at timestamp t, after selling all the previously held stocks. If the algorithm buys stock s (for the price X_s^t), the expected price at which it will get sold at time $t+1$ is $\mathbb{E}[X_s^{t+1}] = \mu_s$, because X_s^t and X_s^{t+1} are independent. Thus, conditional on the current price X_s^t, the expected profit gained from stock s is $\mu_s - X_s^t$ if stock s is bought (and zero otherwise). Thus, the best strategy for the online algorithm finds the maximum weight independent subset of stocks with respect to the weight function $(\mu - X^t)$. Therefore, conditioned on the current prices of the stocks X^t, the expected profit of the algorithm resulting from buying stocks in the current time step and selling them in the next one is given by $\mathtt{top}_{\mathcal{M}}(\mu - X^t)$. Since X^t is identically distributed as X, the unconditional expected profit of the algorithm resulting from buying stocks in any time step and selling them in the next one is given by $\mathbb{E}[\mathtt{top}_{\mathcal{M}}(\mu - X)]$. Summing over time steps $t \in [n-1]$, we get the result. The complete optimal online algorithm is given by Algorithm 2.2. □

Algorithm 2.2. Optimal Online Algorithm for Matroid Trading Prophet

Input: Matroid $\mathcal{M} = (E, \mathcal{I})$, where $E = [k]$, joint distribution $\mathcal{D}$, and n.
for $t = 1$ to n **do**
 Observe the prices $X_1^t, X_2^t, \ldots, X_k^t$ (drawn from the joint distribution $\mathcal{D}$).
 Sell all currently held stocks at the observed prices.
 $H \leftarrow$ maximum weight independent set of $\mathcal{M}$ with respect to the weight function $\mu - X^t$ (compute H using Algorithm 2.1).
 Buy all stocks in H.
end for

As a consequence of Lemma 2.1 and Lemma 2.2, the task of proving bounds on competitive ratio reduces to proving bounds on the ratio of $\mathbb{E}[\mathtt{top}_{\mathcal{M}}(\mu - X)]$ to $\mathbb{E}[\mathtt{top}_{\mathcal{M}}(X' - X)]$. Here, the quantities $\mathbb{E}[\mathtt{top}_{\mathcal{M}}(\mu - X)]$ and $\mathbb{E}[\mathtt{top}_{\mathcal{M}}(X' - X)]$ as the per-time-step expected profit of the optimal online algorithm and the optimal offline algorithm respectively. Similarly, for the (k, ℓ, ℓ')-Trading Prophet Problem, we have,

Corollary 2.1. (to Lemma 2.1) *For every instance of* (k, ℓ, ℓ')-Trading Prophet Problem, *the expected overall profit of the optimal offline algorithm is*

$$(n-1)\mathbb{E}[\mathtt{top}_{\ell'}(X' - X)].$$

Corollary 2.2. (to Lemma 2.2) *For every instance of* (k, ℓ, ℓ')-Trading Prophet Problem, *the expected overall profit of the optimal online algorithm is*

$$(n-1)\,\mathbb{E}[\mathtt{top}_{\ell}(\mu - X)].$$

3 Hardness Results

3.1 Arbitrary matroids, correlated distributions. In this section, we prove an upper bound on the competitive ratio of the $\mathcal{M}$-Trading Prophet Problem.

Theorem 3.1. (Restated Theorem 1.4) *Let $\mathcal{M}$ be an arbitrary matroid and d be the density of $\mathcal{M}$. Then there does not exist an algorithm for* $\mathcal{M}$-Trading Prophet Problem *whose competitive ratio is greater than* $\frac{1}{1+d}$.

Proof. Using Lemma 2.1, Lemma 2.2, it is sufficient to construct an instance which forces

$$\mathbb{E}[\mathtt{top}_{\mathcal{M}}(\mu - X)] \le \frac{1}{1+d} \cdot \mathbb{E}[\mathtt{top}_{\mathcal{M}}(X' - X)].$$

Since the density of $\mathcal{M}$ is d, there exists a subset E' of the ground set E of $\mathcal{M}$ such that $\frac{|E'|}{\mathtt{rk}(E')} = d$. In our instance, the weights of all elements in the set $E \setminus E'$ are zero with probability 1. Consequently, our analysis can be focused on the restriction of $\mathcal{M}$ to the set E'. Let $k = |E'|$ and $r = \mathtt{rk}(E')$, which implies $d = k/r$.

Let ε be an arbitrarily small positive constant. Consider the instance in which X is distributed as follows,

$$X = (X_1, X_2, \ldots, X_k) = \begin{cases} (0, 0, \ldots, 0) & \text{w.p. } 1 - \varepsilon, \\ (-\frac{1}{\varepsilon}, -\frac{1}{\varepsilon}, \ldots, -\frac{1}{\varepsilon}) & \text{w.p. } \varepsilon - \frac{k\,\varepsilon^2}{1+k\varepsilon}, \\ (\frac{1}{\varepsilon^2}, 0, \ldots, 0) & \text{w.p. } \frac{\varepsilon^2}{1+k\varepsilon}, \\ \vdots & \vdots \\ (0, \ldots, 0, \frac{1}{\varepsilon^2}) & \text{w.p. } \frac{\varepsilon^2}{1+k\varepsilon}. \end{cases}$$

Note that $\mu = \mathbb{E}[X] = (0, \dots, 0)$. Recall that X' and X are independent and identically distributed. Let $\mathcal{E}'$ be the event where X' takes the value $(0, \dots, 0)$ and $\mathcal{E}$ be the event where X takes the value $(0, \dots, 0)$. Notice that the events $\mathcal{E}$ and $\mathcal{E}'$ are independent. The per-time-stamp expected profit of the optimal offline algorithm is

$$\begin{aligned}\mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(X'-X\right)\right] &\geq \mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(X'-X\right)\middle|\,\mathcal{E}\right]\cdot\mathbb{P}[\mathcal{E}] + \mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(X'-X\right)\middle|\,\mathcal{E}'\right]\cdot\mathbb{P}[\mathcal{E}'] \\ &= \frac{1}{\varepsilon^2}\cdot\frac{k\varepsilon^2}{1+k\varepsilon}\cdot(1-\varepsilon) + \frac{r}{\varepsilon}\cdot\left(\varepsilon - \frac{k\varepsilon^2}{1+k\varepsilon}\right)\cdot(1-\varepsilon) \\ &= (1-\varepsilon)\frac{k}{1+k\varepsilon} + r(1-\varepsilon)\left(1-\frac{k\varepsilon}{1+k\varepsilon}\right).\end{aligned}$$

And therefore, $\lim_{\varepsilon\to 0}\mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(X'-X\right)\right] \geq k + r$.

Let $\mathcal{E}_1$ be the event where X takes the value $(-\frac{1}{\varepsilon}, \dots, -\frac{1}{\varepsilon})$. Except for the event $\mathcal{E}_1$, the coordinates of X are non-negative, therefore the weight of the maximum weight independent set with respect to the weight function $\mu - X = -X$ is 0. Thus, the per-time-stamp expected profit of the optimal offline algorithm is

$$\begin{aligned}\mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(\mu - X\right)\right] &= \mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(-X\right)\right] \\ &= \mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(-X\right)\middle|\,\mathcal{E}_1\right]\cdot\mathbb{P}[\mathcal{E}_1] \\ &= \frac{r}{\varepsilon}\cdot\left(\varepsilon - \frac{k\varepsilon^2}{1+k\varepsilon}\right) \\ &= r\left(1-\frac{k\varepsilon}{1+k\varepsilon}\right),\end{aligned}$$

and therefore, $\lim_{\varepsilon\to 0}\mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(\mu - X\right)\right] = r$.

It follows that

$$\lim_{\varepsilon\to 0}\frac{\mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(\mu - X\right)\right]}{\mathbb{E}\left[\mathtt{top}_{\mathcal{M}}\left(X'-X\right)\right]} \leq \frac{r}{k+r} = \frac{1}{d+1},$$

as required. □

Corollary 3.1. (to Theorem 3.1) *There does not exist an algorithm for* (k, ℓ)-Trading Prophet Problem *whose competitive ratio is greater than* $\frac{\ell}{k+\ell}$.

Proof. Follows from applying Theorem 3.1 to the ℓ-uniform matroid over the set of k stocks and using Observation 2.1. □

3.2 Uniform matroids, independent distributions. In this section, we prove an upper bound on the competitive ratio of the (k, ℓ, ℓ')-Trading Prophet Problem.

Theorem 3.2. (Restated Theorem 1.2) *For every* k, ℓ *and* ℓ', *there exists an independently distributed instance of the* (k, ℓ, ℓ')-Trading Prophet Problem *(i.e., one in which stock prices at any given time are independent) for which no algorithm can achieve a competitive ratio greater than* $\min\left\{\frac{1}{2}, \frac{\ell}{k}\right\}$.

Proof. Using Corollary 2.1, Corollary 2.2, it is sufficient to construct an instance which forces

$$\mathbb{E}\left[\mathtt{top}_{\ell}\left(\mu - X\right)\right] \leq \min\left\{\frac{1}{2}, \frac{\ell}{k}\right\}\cdot\mathbb{E}\left[\mathtt{top}_{\ell'}\left(X'-X\right)\right].$$

To force the upper bound of ℓ/k on the competitive ratio, consider the instance in which each X_s is distributed as follows,

$$X_s = \begin{cases} 0 & \text{w.p. } 1-\varepsilon, \\ \frac{1}{\varepsilon} & \text{w.p. } \varepsilon, \end{cases}$$

where ε is an arbitrarily small positive constant. Recall that each X'_s is identically distributed as X_s. Let $\mathcal{E}$ be the event that there exists a unique $s^* \in [k]$ such that X'_{s^*} takes the value $\frac{1}{\varepsilon}$, and the other $2k-1$ random variables

X_s and X'_s take value 0. Thus, the per-time-stamp expected profit of the optimal offline algorithm is

$$\mathbb{E}\left[\mathtt{top}_{\ell'}\left(X'-X\right)\right] \geq \mathbb{E}\left[\mathtt{top}_{\ell'}\left(X'-X\right)\middle|\, \mathcal{E}\right] \cdot \mathbb{P}[\mathcal{E}]$$
$$= \frac{1}{\varepsilon} \cdot k\varepsilon(1-\varepsilon)^{2k-1} = k(1-\varepsilon)^{2k-1}.$$

In this instance, for every $s \in [k]$, $\mu_s = \mathbb{E}[X_s] = 1$. Therefore, the per-time-stamp expected profit of the optimal online algorithm is

$$\mathbb{E}\left[\mathtt{top}_{\ell}\left(\mu - X\right)\right] = \mathbb{E}\left[\mathtt{top}_{\ell}\left(\mathbb{1} - X\right)\right] \leq \mathbb{E}\left[\mathtt{top}_{\ell}\left(\mathbb{1}\right)\right] = \ell,$$

where $\mathbb{1}$ is the all ones vector. The above inequality holds because each X_s is non-negative with probability one. It follows that

$$\frac{\mathbb{E}\left[\mathtt{top}_{\ell}\left(\mu - X\right)\right]}{\mathbb{E}\left[\mathtt{top}_{\ell'}\left(X'-X\right)\right]} \leq \frac{\ell}{k(1-\varepsilon)^{2k-1}},$$

which approaches ℓ/k as ε approaches 0.

To force the upper bound of $1/2$ on the competitive ratio, consider the instance in which each X_s is distributed as follows,

$$X_s = \begin{cases} -\frac{1}{\varepsilon} & \text{w.p. } \varepsilon, \\ 0 & \text{w.p. } 1-2\varepsilon, \\ \frac{1}{\varepsilon} & \text{w.p. } \varepsilon, \end{cases}$$

where ε is an arbitrarily small positive constant. Each X'_s is distributed identically as X_s. Let $\mathcal{E}$ be the event where exactly one X_s takes the value $-\frac{1}{\varepsilon}$ and all other $2k-1$ random variables are 0. Let $\mathcal{E}'$ be the event where exactly one X'_s takes the value $\frac{1}{\varepsilon}$ and all other $2k-1$ random variables are 0. Notice that the events $\mathcal{E}$ and $\mathcal{E}'$ are disjoint, and therefore, $\mathbb{P}[\mathcal{E} \cup \mathcal{E}'] = \mathbb{P}[\mathcal{E}] + \mathbb{P}[\mathcal{E}'] = k\varepsilon(1-2\varepsilon)^{2k-1} + k\varepsilon(1-2\varepsilon)^{2k-1} = 2k\varepsilon(1-2\varepsilon)^{2k-1}$. The per-time-stamp expected profit of the optimal online algorithm is

$$\mathbb{E}\left[\mathtt{top}_{\ell'}\left(X'-X\right)\right] \geq \mathbb{E}\left[\mathtt{top}_{\ell'}\left(X'-X\right)|(\mathcal{E} \cup \mathcal{E}')\right] \cdot \mathbb{P}[\mathcal{E} \cup \mathcal{E}']$$
$$= \frac{1}{\varepsilon} \cdot 2k\varepsilon(1-2\varepsilon)^{2k-1} = 2k(1-2\varepsilon)^{2k-1}.$$

Let $\mathcal{E}_1$ be the event where exactly one X_s takes the value $-\frac{1}{\varepsilon}$ and all other $k-1$ coordinates of X are 0. Let $\mathcal{E}_0$ be the event where all X'_s are 0. Notice that the events $\mathcal{E}_1$ and $\mathcal{E}_0$ are disjoint. Here $\mathbb{P}[\mathcal{E}_1] = k\varepsilon(1-2\varepsilon)^{2k-1} \leq k\varepsilon$ and $\mathbb{P}[\neg(\mathcal{E}_0 \cup \mathcal{E}_1)] \leq \binom{k}{2} \cdot \varepsilon^2$ by union bound.
In this instance, for every $s \in [k]$, $\mu_s = \mathbb{E}[X_s] = 0$. Therefore, the per-time-stamp expected profit of the optimal online algorithm is

$$\begin{aligned}\mathbb{E}\left[\mathtt{top}_{\ell}\left(\mu - X\right)\right] &= \mathbb{E}\left[\mathtt{top}_{\ell}\left(-X\right)\right] \\ &= \mathbb{E}\left[\mathtt{top}_{\ell}\left(-X\right)|\mathcal{E}_0\right] \cdot \mathbb{P}[\mathcal{E}_0] + \mathbb{E}\left[\mathtt{top}_{\ell}\left(-X\right)|\mathcal{E}_1\right] \cdot \mathbb{P}[\mathcal{E}_1] \\ &+ \mathbb{E}\left[\mathtt{top}_{\ell}\left(-X\right)|\neg(\mathcal{E}_0 \cup \mathcal{E}_1)\right] \cdot \mathbb{P}[\neg(\mathcal{E}_0 \cup \mathcal{E}_1)] \\ &\leq 0 \cdot \mathbb{P}[\mathcal{E}_0] + \frac{1}{\varepsilon} \cdot k\varepsilon + \mathbb{E}\left[\mathtt{top}_{\ell}\left(\frac{1}{\varepsilon}, \ldots, \frac{1}{\varepsilon}\right)\right] \cdot \binom{k}{2} \cdot \varepsilon^2 \\ &= k + \frac{l}{\varepsilon} \cdot \binom{k}{2} \cdot \varepsilon^2 = k + l\varepsilon \cdot \binom{k}{2},\end{aligned}$$

where the inequality holds because each $X_s \geq -\frac{1}{\varepsilon}$ with probability one. It follows that

$$\frac{\mathbb{E}\left[\mathtt{top}_{\ell}\left(\mu - X\right)\right]}{\mathbb{E}\left[\mathtt{top}_{\ell'}\left(X'-X\right)\right]} \leq \frac{k + l\varepsilon \cdot \binom{k}{2}}{2k(1-2\varepsilon)^{2k-1}},$$

which approaches $1/2$ as ε approaches 0. $\square$

4 Algorithmic Results

For analysis, we consider a subclass of instances called *zero-expectation instances* and argue that it is sufficient to prove a competitiveness guarantee of Algorithm 2.2 on such zero-expectation instances.

DEFINITION 4.1. (ZERO-EXPECTATION INSTANCE) *An instance of the $\mathcal{M}$-TRADING PROPHET PROBLEM with joint distribution $\mathcal{D}$ is called a* zero-expectation instance *if $\mathbb{E}_{X\sim\mathcal{D}}[X] = \bar{0}$, where $\bar{0} = (0, \ldots, 0)$.*

PROPOSITION 4.1. *If Algorithm 2.2 is α-competitive on zero-expectation instances, then it is α-competitive on all instances.*

Proof. Given an arbitrary instance with joint CDF[4] $F : \mathbb{R}^k \longrightarrow [0, 1]$, construct the instance with joint CDF G defined as, for all $x \in \mathbb{R}^k$, $G(x) = F(x + \mu)$, where $\mu = (\mu_1, \ldots, \mu_k) \in \mathbb{R}^k$ is the vector of expectations of random variables whose joint CDF is F. Recall that the joint CDFs of X, X' are both F. Define $Y = X - \mu$ and $Y' = X' - \mu$, so that $X' - X = Y' - Y$. Observe that the joint CDF of Y and Y' is G, and their expectation is 0. Thus, the constructed instance is a zero-expectation instance.

Since

$$\texttt{top}_{\mathcal{M}}(X' - X) = \texttt{top}_{\mathcal{M}}(Y' - Y),$$

the per-time-stamp profits of the offline optimal algorithm are the same on both the given and the constructed instance with probability one. Similarly, since

$$\texttt{top}_{\mathcal{M}}(\mu - X) = \texttt{top}_{\mathcal{M}}(-Y) = \texttt{top}_{\mathcal{M}}(\mathbb{E}[Y] - Y),$$

the per-time-stamp profits of Algorithm 2.2 are the same on both the given and the constructed instance with probability one. Thus, if Algorithm 2.2 is α-competitive on the constructed zero-expectation instance, it is α-competitive on the given instance. ∎

As a consequence of the above proposition, the task of getting a competitive guarantee on arbitrary instances reduces to getting a competitive guarantee on zero-expectation instances. Therefore, we assume henceforward that the instance given to Algorithm 2.2 is a *zero expectation instance.*

4.1 Arbitrary matroids, correlated distributions. In this section, we prove the competitive bound for the $\mathcal{M}$-TRADING PROPHET PROBLEM. We first prove a result that quantifies the following intuitive arguments: if the matroid is dense then most of its elements are present in a maximum weight independent set, and therefore, its weight is close to the initial weight of all the elements of the matroid.

LEMMA 4.1. *Let M be a loopless matroid of density d on the ground set E, and let $w : E \longrightarrow \mathbb{R}^+$ be a weight function, then the following inequality holds,*

$$\sum_{e\in E} w_e \leq d \cdot \texttt{top}_{\mathcal{M}}(w).$$

Proof. Let $e_1, \ldots, e_n$ be the elements of E in non-increasing order of their weights. Let $w_{e_{n+1}} = 0$. The summation of weights can be written as,

$$\sum_{e\in E} w_e = \sum_{i=1}^{n} w_{e_i} = \sum_{i=1}^{n} i \cdot (w_{e_i} - w_{e_{i+1}}).$$

By applying the definition of density to the set $\{e_1, \ldots, e_i\}$, we get $e_i \leq d \cdot \texttt{rk}(\{e_1, \ldots, e_i\})$,

$$\begin{aligned}\sum_{i=1}^{n} w_{e_i} &\leq d \sum_{i=1}^{n} \texttt{rk}(\{e_1, \ldots, e_i\}) \cdot (w_{e_i} - w_{e_{i+1}}) \\ &= d \sum_{i=1}^{n} (\texttt{rk}(\{e_1, \ldots, e_i\}) - \texttt{rk}(\{e_1, \ldots, e_{i-1}\})) \cdot w_{e_i},\end{aligned}$$

[4]The *joint CDF* of a vector $(X_1, \ldots, X_k)$ of random variables is the function F given by, $F(x_1, \ldots, x_k) = \mathbb{P}[X_1 \leq x_1 \wedge \cdots \wedge X_k \leq x_k]$.

where the notation $\{e_1, ..., e_0\}$ denotes the empty set – whose rank is zero. The possible values of $\texttt{rk}(\{e_1, \ldots, e_i\}) - \texttt{rk}(\{e_1, \ldots, e_{i-1}\})$ are 1 and 0, because the sets differ by just one element, namely e_i. If e_i is picked by Kruskal's algorithm in the maximum weight independent set, then e_i is not spanned by $\{e_1, \ldots, e_{i-1}\}$, resulting in $\texttt{rk}(\{e_1, \ldots, e_i\}) - \texttt{rk}(\{e_1, \ldots, e_{i-1}\}) = 1$. Conversely, if Kruskal's algorithm does not pick e_i, then it means that e_i is spanned by $\{e_1, \ldots, e_{i-1}\}$, resulting in $\texttt{rk}(\{e_1, \ldots, e_i\}) - \texttt{rk}(\{e_1, \ldots, e_{i-1}\}) = 0$. Since Kruskal's algorithm is guaranteed to find a maximum weight independent set, we have,

$$\sum_{i=1}^{n} (\texttt{rk}(\{e_1, \ldots, e_i\}) - \texttt{rk}(\{e_1, \ldots, e_{i-1}\})) \cdot w_{e_i} = \texttt{top}_{\mathcal{M}}(w),$$

and therefore,

$$\sum_{i=1}^{n} w_{e_i} \leq d \cdot \texttt{top}_{\mathcal{M}}(w),$$

as required. □

Consider two arbitrary weight assignments, x and x', to the elements of a matroid $\mathcal{M}$. We next prove a helpful upper bound on $\texttt{top}_{\mathcal{M}}(x' - x)$.

Lemma 4.2. *Let $\mathcal{M}$ be an arbitrary matroid over $[k]$ with density d. Then for every $x, x' \in \mathbb{R}^k$, the following inequality holds*

$$\texttt{top}_{\mathcal{M}}(x' - x) \leq \texttt{top}_{\mathcal{M}}(-x) + \sum_{i=1}^{k} x'_i + d \cdot \texttt{top}_{\mathcal{M}}(-x').$$

Proof. Let Z be the maximum weight independent set with respect to the weight function $(x' - x)$. Then

$$\texttt{top}_{\mathcal{M}}(x' - x) = \sum_{i \in Z} (x'_i - x_i) = \sum_{i \in Z} x'_i + \sum_{i \in Z} (-x_i) \leq \sum_{i \in Z} x'_i + \texttt{top}_{\mathcal{M}}(-x).$$

Here, $\sum_{i \in Z} x'_i \leq \sum_{i \in Z} (x'_i)^+ \leq \sum_{i=1}^{k} (x'_i)^+$. Observe that every real number y can be written as $y = y^+ - (-y)^+$, and therefore, $\sum_{i=1}^{k} (x'_i)^+$ can be written as $\sum_{i=1}^{k} x'_i + \sum_{i=1}^{k} (-x'_i)^+$. Thus, we get,

$$\texttt{top}_{\mathcal{M}}(x' - x) \leq \texttt{top}_{\mathcal{M}}(-x) + \sum_{i=1}^{k} x'_i + \sum_{i=1}^{k} (-x'_i)^+ \leq \texttt{top}_{\mathcal{M}}(-x) + \sum_{i=1}^{k} x'_i + d \cdot \texttt{top}_{\mathcal{M}}(-x'),$$

where the last inequality follows from Lemma 4.1. □

Having gathered all the necessary prerequisites, we are now prepared to present Theorem 1.5, which ensures the desired competitive ratio.

Theorem 4.1. (Restated Theorem 1.5) *For every matroid $\mathcal{M}$, Algorithm 2.2 is $\frac{1}{1+d}$-competitive, where d is the density of $\mathcal{M}$.*

Proof. Recall from Lemma 2.1 and Lemma 2.2 that it suffices to prove

$$\mathbb{E}\left[\texttt{top}_{\mathcal{M}}\left(X' - X\right)\right] \leq (d+1)\, \mathbb{E}\left[\texttt{top}_{\mathcal{M}}\left(-X\right)\right].$$

By Lemma 4.2, the following holds with probability one:

$$\texttt{top}_{\mathcal{M}}(X' - X) \leq \texttt{top}_{\mathcal{M}}(-X) + \sum_{i=1}^{k} X'_i + d \cdot \texttt{top}_{\mathcal{M}}(-X').$$

Taking expectations on both sides, we get,

$$\mathbb{E}\left[\texttt{top}_{\mathcal{M}}(X' - X)\right] \leq \mathbb{E}\left[\texttt{top}_{\mathcal{M}}(-X)\right] + \sum_{i} \mathbb{E}\left[X'_i\right] + d \cdot \mathbb{E}\left[\texttt{top}_{\mathcal{M}}(-X')\right].$$

Observe that $\mathbb{E}[\mathtt{top}_{\mathcal{M}}(-X)] = \mathbb{E}[\mathtt{top}_{\mathcal{M}}(-X')]$ because X and X' are identically distributed. Since we are working with zero-expectation instances, $\mathbb{E}[X_i'] = 0$ for every i. Therefore, we get,

$$\mathbb{E}\left[\mathtt{top}_{\mathcal{M}}(X' - X)\right] \leq (d+1)\,\mathbb{E}\left[\mathtt{top}_{\mathcal{M}}(-X)\right],$$

as required. ◻

COROLLARY 4.1. (TO THEOREM 4.1) *For every k and l, Algorithm 2.2 is $\frac{\ell}{k+\ell}$-competitive for the (k,ℓ)-TRADING PROPHET PROBLEM.*

Proof. Follows from applying Theorem 4.1 to the ℓ-uniform matroid over the set of k stocks and using Observation 2.1. ◻

4.2 Uniform matroids, independent distributions. In this section, we will prove that our algorithm for the (k,ℓ,ℓ')-TRADING PROPHET PROBLEM has a competitive ratio of $\min\left\{\frac{1}{2},\frac{\ell}{k}\right\}$. We start by bounding $\mathbb{E}\left[\mathtt{top}_{\ell'}(X' - X)\right]$, the per-time-stamp expected profit of the offline optimal algorithm in Lemma 4.3 given below.

LEMMA 4.3. *For every zero-expectation instance,*

$$\mathbb{E}\left[\mathtt{top}_{\ell'}(X' - X)\right] \leq \int_0^\infty \left(2\sum_s \mathbb{P}[-X_s \geq x] - \frac{2}{k-1}\sum_{s<s'} \mathbb{P}[-X_s \geq x]\mathbb{P}[-X_{s'} \geq x]\right) dx.$$

Proof. From the definition of the $\mathtt{top}_{\ell'}$ function, we have,

$$\begin{aligned}\mathbb{E}\left[\mathtt{top}_{\ell'}(X' - X)\right] &\leq \mathbb{E}\left[\sum_{s=1}^{k}(X_s' - X_s)^+\right] \\ &= \mathbb{E}\left[\sum_{s=1}^{k}\int_{-\infty}^{\infty} \mathbb{1}[X_s' \geq x \geq X_s]\,dx\right],\end{aligned}$$

where $\mathbb{1}[\cdot]$ denotes the indicator function. By linearity of expectation and the fact that the expectation of the indicator of an event is the probability of the event, we have,

$$\mathbb{E}\left[\mathtt{top}_{\ell'}(X' - X)\right] \leq \sum_{s=1}^{k}\left(\int_{-\infty}^{\infty} \mathbb{P}[X_s' \geq x \geq X_s]\,dx\right) = \sum_{s=1}^{k}\left(\int_{-\infty}^{\infty} \mathbb{P}[X_s' \geq x]\,\mathbb{P}[X_s \leq x]\,dx\right),$$

where the first equality holds because X_s and X_s' are independent. Since X_s and X_s' are identically distributed, we have,

$$\begin{aligned}\mathbb{E}\left[\mathtt{top}_{\ell'}(X' - X)\right] &\leq \sum_{s=1}^{k}\left(\int_{-\infty}^{\infty} \mathbb{P}[X_s \geq x]\,\mathbb{P}[X_s \leq x]\,dx\right) \\ &= \sum_{s=1}^{k}\left(\int_{-\infty}^{0} \mathbb{P}[X_s \geq x]\,\mathbb{P}[X_s \leq x]\,dx + \int_0^\infty \mathbb{P}[X_s \geq x]\,\mathbb{P}[X_s \leq x]\,dx\right) \\ &\leq \sum_{s=1}^{k}\left(\int_{-\infty}^{0} (1 - \mathbb{P}[X_s \leq x])\,\mathbb{P}[X_s \leq x]\,dx + \int_0^\infty \mathbb{P}[X_s \geq x]\,dx\right). \qquad (4.1)\end{aligned}$$

Since we are given a *zero-expectation instance*, we have for all $s \in [k]$,

$$0 = \mu_s = \int_0^\infty \mathbb{P}[X_s \geq x]\,dx - \int_{-\infty}^{0} \mathbb{P}[X_s \leq x]\,dx,$$

and thus,

$$\int_0^\infty \mathbb{P}[X_s \geq x]\,dx = \int_{-\infty}^{0} \mathbb{P}[X_s \leq x]\,dx.$$

Substituting in Equation (4.1), we have,

$$
\begin{aligned}
\mathbb{E}\left[\mathtt{top}_{\ell'}\left(X'-X\right)\right] &\leq \sum_{s=1}^{k}\left(\int_{-\infty}^{0}\left(1-\mathbb{P}[X_s\leq x]\right)\mathbb{P}[X_s\leq x]\,dx+\int_{-\infty}^{0}\mathbb{P}[X_s\leq x]\,dx\right)\\
&= \int_{-\infty}^{0}\sum_{s}\left(2\,\mathbb{P}[X_s\leq x]-\mathbb{P}[X_s\leq x]^2\right)\,dx. \qquad (4.2)
\end{aligned}
$$

Now, we have

$$
0\leq \sum_{1\leq s<s'\leq k}\left(\mathbb{P}[X_s\leq x]-\mathbb{P}[X_{s'}\leq x]\right)^2=(k-1)\sum_{s\in[k]}\mathbb{P}[X_s\leq x]^2-2\sum_{1\leq s<s'\leq k}\mathbb{P}[X_s\leq x]\cdot\mathbb{P}[X_{s'}\leq x],
$$

and thus,

$$
\sum_{s\in[k]}\mathbb{P}[X_s\leq x]^2\geq\frac{2}{k-1}\sum_{1\leq s<s'\leq k}\mathbb{P}[X_s\leq x]\cdot\mathbb{P}[X_{s'}\leq x].
$$

Substituting in Equation (4.2), we get,

$$
\begin{aligned}
\mathbb{E}\left[\mathtt{top}_{\ell'}\left(X'-X\right)\right] &\leq \int_{-\infty}^{0}\left(2\sum_{s}\mathbb{P}[X_s\leq x]-\frac{2}{k-1}\sum_{s<s'}\mathbb{P}[X_s\leq x]\mathbb{P}[X_{s'}\leq x]\right)\,dx\\
&= \int_{0}^{\infty}\left(2\sum_{s}\mathbb{P}[X_s\leq -x]-\frac{2}{k-1}\sum_{s<s'}\mathbb{P}[X_s\leq -x]\mathbb{P}[X_{s'}\leq -x]\right)\,dx\\
&= \int_{0}^{\infty}\left(2\sum_{s}\mathbb{P}[-X_s\geq x]-\frac{2}{k-1}\sum_{s<s'}\mathbb{P}[-X_s\geq x]\mathbb{P}[-X_{s'}\geq x]\right)\,dx,
\end{aligned}
$$

as required. ∎

In Algorithm 4.4, we derive an expression for $\mathbb{E}\left[\mathtt{top}_{\ell}\left(\mu-X\right)\right]$, the pre-time-stamp expected profit of the online Algorithm 2.2 when the stock prices at any given time are independent.

Lemma 4.4. *For every* independently distributed zero-expectation instance*, the following equality holds:*

$$
\mathbb{E}\left[\mathtt{top}_{\ell}\left(\mu-X\right)\right]=\int_{0}^{\infty}\sum_{S\subseteq[k]}\min\{|S|,\ell\}\left(\prod_{s\in S}\mathbb{P}[-X_s\geq x]\right)\cdot\left(\prod_{s'\notin S}\left(1-\mathbb{P}[-X_{s'}\geq x]\right)\right)\,dx.
$$

Proof. In a *zero-expectation instance*, we have $\mu_s=0$ for all $s\in[k]$. Therefore, we have,

$$
\mathbb{E}\left[\mathtt{top}_{\ell}\left(\mu-X\right)\right]=\mathbb{E}\left[\mathtt{top}_{\ell}\left(-X\right)\right].
$$

Define random variables $M_1,\dots,M_\ell$ as follows: M_i is the i'th largest value in the multi-set $\{-X_s\mid s\in[k]\}$ if this value is positive, and 0 otherwise. Observe that $\mathtt{top}_{\ell}\left(-X\right)=\sum_{i=1}^{\ell}M_i$. Therefore,

$$
\mathbb{E}\left[\mathtt{top}_{\ell}\left(-X\right)\right]=\mathbb{E}\left[\sum_{i=1}^{\ell}M_i\right]=\sum_{i=1}^{\ell}\mathbb{E}\left[M_i\right]=\sum_{i=1}^{\ell}\int_{0}^{\infty}\mathbb{P}\left[M_i\geq x\right]\,dx=\int_{0}^{\infty}\sum_{i=1}^{\ell}\mathbb{P}\left[M_i\geq x\right]\,dx,
$$

where the second equality holds by linearity of expectation, and the third one holds because M_i is non-negative with probability one. Using the fact that the probability of an event is the expectation of its indicator, we get,

$$
\mathbb{E}\left[\mathtt{top}_{\ell}\left(-X\right)\right]=\int_{0}^{\infty}\sum_{i=1}^{\ell}\mathbb{E}\left[\mathbb{1}\left[M_i\geq x\right]\right]\,dx=\int_{0}^{\infty}\mathbb{E}\left[\sum_{i=1}^{\ell}\mathbb{1}\left[M_i\geq x\right]\right]\,dx, \qquad (4.3)
$$

where the second equality holds by linearity of expectation. For $x \in \mathbb{R}_+$, let the set-valued random variable $\mathcal{S}(x)$ be defined as $\mathcal{S}(x) = \{s \in [k] \mid -X_s \geq x\}$. Since X_s's are independent, we have that for every $S \subseteq [k]$,

$$\Pr[\mathcal{S}(x) = S] = \left(\prod_{s \in S} \mathbb{P}[-X_s \geq x]\right) \cdot \left(\prod_{s' \notin S} (1 - \mathbb{P}[-X_{s'} \geq x])\right).$$

Conditioned on the event $\mathcal{S}(x) = S$, the event $M_i \geq x$ happens if and only if $|S| \geq i$, and hence, $\sum_{i=1}^{\ell} \mathbb{1}[M_i \geq x] = \min\{|S|, \ell\}$ with probability one. Using this fact and the expression for $\Pr[\mathcal{S}(x) = S]$ in Equation (4.3), we get,

$$\begin{aligned}
\mathbb{E}\left[\mathtt{top}_\ell(-X)\right] &= \int_0^\infty \sum_{S \subseteq [k]} \mathbb{E}\left[\sum_{i=1}^{\ell} \mathbb{1}[M_i \geq x] \,\middle|\, \mathcal{S}(x) = S\right] \cdot \Pr[\mathcal{S}(x) = S]\, dx \\
&= \int_0^\infty \sum_{S \subseteq [k]} \min\{|S|, \ell\} \left(\prod_{s \in S} \mathbb{P}[-X_s \geq x]\right) \cdot \left(\prod_{s' \notin S} (1 - \mathbb{P}[-X_{s'} \geq x])\right) dx,
\end{aligned}$$

as required. ∎

Next, we derive an inequality that will be used to relate $\mathbb{E}\left[\mathtt{top}_{\ell'}(X' - X)\right]$ and $\mathbb{E}\left[\mathtt{top}_\ell(\mu - X)\right]$ in Theorem 1.3.

LEMMA 4.5. *For every k, ℓ and $\{a_i\}_{i \in [k]} \in [0,1]^k$, the following inequality holds:*

$$\max\left\{2, \frac{k}{\ell}\right\} \cdot \left(\sum_{S \subseteq [k]} \min\{|S|, \ell\} \prod_{s \in S} a_s \prod_{s' \notin S} (1 - a_{s'})\right) \geq \left(2 \sum_{s \in [k]} a_s - \frac{2}{k-1} \sum_{s, s' \in [k],\ s < s'} a_s a_{s'}\right).$$

Proof. Define $f : [0,1]^k \longrightarrow \mathbb{R}$ as

$$\begin{aligned}
f(a_1, \ldots, a_k) = \max\left\{2, \frac{k}{\ell}\right\} \cdot &\left(\sum_{S \subseteq [k]} \min\{|S|, \ell\} \prod_{s \in S} a_s \prod_{s' \notin S} (1 - a_{s'})\right) \\
&- \left(2 \sum_{s \in [k]} a_s - \frac{2}{k-1} \sum_{s, s' \in [k],\ s < s'} a_s a_{s'}\right).
\end{aligned}$$

We need to prove that f is non-negative on $[0,1]^n$. Since f is multilinear in $a_1, \ldots, a_k$, it is sufficient to prove that f is non-negative on the corners of its domain $[0,1]^n$, that is, $f(a_1, \ldots, a_k) \geq 0$ for all $(a_1, \ldots, a_k) \in \{0,1\}^k$ (see Corollary 2.3 of Laneve et al. [2010]).

Consider an arbitrary $(a_1, \ldots, a_k) \in \{0,1\}^k$. Let z be the number of a_s's are 1 so that $k - z$ is the number of the a_s's are 0. Thus $z \in \{0, \ldots, k\}$. Also,

$$f(a_1, \ldots, a_k) = \max\left\{2, \frac{k}{\ell}\right\} \cdot \min\{z, \ell\} - 2z + \frac{2}{k-1} \cdot \binom{z}{2} = \max\left\{2, \frac{k}{\ell}\right\} \cdot \min\{z, \ell\} - 2z + \frac{z(z-1)}{k-1}.$$

If $z \leq \ell$, then

$$f(a_1, \ldots, a_k) = \max\left\{2, \frac{k}{\ell}\right\} \cdot z - 2z + \frac{z(z-1)}{k-1} \geq \frac{z(z-1)}{k-1} > 0.$$

If $z > \ell$, then

$$f(a_1, \ldots, a_k) = \max\left\{2, \frac{k}{\ell}\right\} \cdot \ell - 2z + \frac{z(z-1)}{k-1} \geq \frac{k}{\ell} \cdot \ell - 2z + \frac{z(z-1)}{k-1} = k - \frac{z(2k - z - 1)}{k-1}.$$

It is easy to check that the function $z \longmapsto k - z(2k - z - 1)/(k-1)$ is 0 at $z = k$ and $z = k - 1$, and is decreasing in $[0, k-1]$. Thus $k - z(2k - z - 1)/(k-1) \geq 0$ for all $z \in [k]$, which implies $f(a_1, \ldots, a_k) \geq 0$, as required. ∎

Having gathered all the necessary prerequisites, we are now prepared to present Theorem 1.3, which ensures the desired competitive ratio.

Theorem 4.2. (Restated Theorem 1.3) *For every k, ℓ, and ℓ', Algorithm 2.2 is a* $\min\left\{\frac{1}{2}, \frac{\ell}{k}\right\}$*-competitive algorithm for the* (k, ℓ, ℓ')-Trading Prophet Problem *when the stock prices at any given time are independent.*

Proof. In Lemma 4.3, we obtained the following upper bound on the pre-time-stamp expected profit of offline algorithm:

$$\mathbb{E}\left[\mathtt{top}_{\ell'}\left(X' - X\right)\right] \leq \int_0^\infty \left(2\sum_s \mathbb{P}[-X_s \geq x] - \frac{2}{k-1}\sum_{s<s'} \mathbb{P}[-X_s \geq x]\mathbb{P}[-X_{s'} \geq x]\right) dx.$$

In Lemma 4.4, we obtained the following expression for the per-time-stamp expected profit of Algorithm 2.2:

$$\mathbb{E}\left[\mathtt{top}_{\ell}\left(\mu - X\right)\right] = \int_0^\infty \sum_{S\subseteq[k]} \min\{|S|, \ell\} \left(\prod_{s\in S} \mathbb{P}[-X_s \geq x]\right) \cdot \left(\prod_{s'\notin S} \left(1 - \mathbb{P}[-X_{s'} \geq x]\right)\right) dx.$$

For an arbitrary $x \geq 0$, using Lemma 4.5 with $a_s = \mathbb{P}[-X_s \geq x]$ for all $s \in [k]$, we get,

$$\begin{aligned}\max\left\{2, \frac{k}{\ell}\right\} \cdot \sum_{S\subseteq[k]} \min\{|S|, \ell\} \left(\prod_{s\in S} \mathbb{P}[-X_s \geq x]\right) \cdot \left(\prod_{s'\notin S} \left(1 - \mathbb{P}[-X_{s'} \geq x]\right)\right)& \\ \geq 2\sum_s \mathbb{P}[-X_s \geq x] - \frac{2}{k-1}\sum_{s<s'} \mathbb{P}[-X_s \geq x]\mathbb{P}[-X_{s'} \geq x]&.\end{aligned}$$

Integrating both sides of the above inequality with respect to x from 0 to ∞, the left-hand-side is $\max\left\{2, \frac{k}{\ell}\right\}$ times $\mathbb{E}\left[\mathtt{top}_{\ell}\left(\mu - X\right)\right]$, and the right-hand-side is the upper bound on $\mathbb{E}\left[\mathtt{top}_{\ell'}\left(X' - X\right)\right]$. Therefore, we get,

$$\max\left\{2, \frac{k}{\ell}\right\} \cdot \mathbb{E}\left[\mathtt{top}_{\ell}\left(\mu - X\right)\right] \geq \mathbb{E}\left[\mathtt{top}_{\ell'}\left(X' - X\right)\right].$$

Thus, Algorithm 2.2 is $\min\left\{\frac{1}{2}, \frac{\ell}{k}\right\}$-competitive. □

5 Concluding Remarks

We formulated generalizations of the *trading prophet* problem [Correa et al., 2023], which we call the (k, ℓ, ℓ')-Trading Prophet Problem and $\mathcal{M}$-Trading Prophet Problem, wherein multiple stocks can be bought and sold at a time. In the (k, ℓ, ℓ')-Trading Prophet Problem on *independently distributed* instances, we showed that the competitive ratio is $\min\left\{\frac{1}{2}, \frac{\ell}{k}\right\}$.

In the $\mathcal{M}$-Trading Prophet Problem, we proved a tight bound of $\frac{1}{1+d}$ on the competitive ratio, where d is the *density* of the matroid. This result holds even if the stock prices at any given time are arbitrarily correlated. While our analysis assumes that the stock prices at different times are independent, it is reasonably easy to verify that our results continue to hold even under the assumption that stock prices at different times are only *pairwise* independent.

We achieved significant simplification of, and generalizations over the results of Correa et al. [2023] using the following observations. First, any algorithm can be simulated by one that sells all held stocks before buying an appropriate subset of stocks. Second, the analysis of the general problem reduces to one where the expected price of every stock is zero.

We believe that the trading prophet under more general constraints, e.g. matchings or intersections of matroid constraints, would be an interesting setting for future investigation.

Acknowledgments

We are grateful to anonymous reviewers for their helpful comments. RV acknowledges support from DST INSPIRE grant no. DST/INSPIRE/04/2020/000107 and SERB grant no. CRG/2022/002621.

References

S. Alaei. Bayesian Combinatorial Auctions: Expanding Single Buyer Mechanisms to Many Buyers. *SIAM Journal on Computing*, 43(2):930–972, 2014.

J. Correa, P. Foncea, R. Hoeksma, T. Oosterwijk, and T. Vredeveld. Recent Developments in Prophet Inequalities. *ACM SIGecom Exchanges*, 17(1):61–70, 2019.

J. Correa, A. Cristi, P. Dütting, M. T. Hajiaghayi, J. Olkowski, and K. Schewior. Trading Prophets. In *EC*, pages 490–510, 2023.

S. Ehsani, M. T. Hajiaghayi, T. Kesselheim, and S. Singla. Prophet Secretary for Combinatorial Auctions and Matroids. In *SODA*, pages 700–714. SIAM, 2018.

H. Esfandiari, M. Hajiaghayi, V. Liaghat, and M. Monemizadeh. Prophet Secretary. *SIAM Journal on Discrete Mathematics*, 31(3):1685–1701, 2017.

M. T. Hajiaghayi, R. Kleinberg, and T. Sandholm. Automated Online Mechanism Design and Prophet Inequalities. In *AAAI*, volume 7, pages 58–65, 2007.

T. P. Hill and R. P. Kertz. Comparisons of Stop Rule and Supremum Expectations of I.I.D. Random Variables. *The Annals of Probability*, 10(2):336 – 345, 1982. doi: 10.1214/aop/1176993861. URL `https://doi.org/10.1214/aop/1176993861`.

R. Kleinberg and S. M. Weinberg. Matroid Prophet Inequalities and Applications to Multi-dimensional Mechanism Design. *Games Econ. Behav.*, 113:97–115, 2019. doi: 10.1016/J.GEB.2014.11.002. URL `https://doi.org/10.1016/j.geb.2014.11.002`.

U. Krengel and L. Sucheston. On Semiamarts, Amarts, and Processes with Finite Value. *Adv. in Probability*, 4: 197–266, 1978.

C. Laneve, T. A. Lascu, and V. Sordoni. The Interval Analysis of Multilinear Expressions. *Electronic Notes in Theoretical Computer Science*, 267(2):43–53, 2010.

B. Lucier. An Economic View of Prophet Inequalities. *ACM SIGecom Exchanges*, 16(1):24–47, 2017.

J. G. Oxley. *Matroid Theory*. Oxford University Press, 1992. ISBN 978-0-19-853563-8.

B. Peng and Z. G. Tang. Order Selection Prophet Inequality: From Threshold Optimization to Arrival Time Design. In *FOCS*, pages 171–178. IEEE, 2022.

T. Roughgarden. Resource Augmentation. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*, pages 72–92. Cambridge University Press, 2020. doi: 10.1017/9781108637435.006. URL `https://doi.org/10.1017/9781108637435.006`.

D. D. Sleator and R. E. Tarjan. Amortized Efficiency of List Update and Paging Rules. *Commun. ACM*, 28(2): 202–208, 1985. doi: 10.1145/2786.2793. URL `https://doi.org/10.1145/2786.2793`.

J. A. Soto. Matroid Secretary Problem in the Random-Assignment Model. *SIAM Journal on Computing*, 42(1): 178–211, 2013.