Online Portfolio Selection with Adversarial Transaction Costs

William Brown Yikai Zhang

Abstract

We present a family of algorithms for online portfolio selection in the presence of returns and transaction costs which may both vary adversarially over time, obtaining log-returns competitive with the best constant rebalanced portfolio in hindsight. Building upon techniques for online optimization with memory, our algorithms allow reward functions to depend on both current and previous holdings in a nonparametric manner, yielding the first no-regret portfolio selection algorithms which allow for arbitrary and potentially non-convex transaction cost expressions rather than only fixed percentage fees. We show that the optimal $O(\log T)$ rate is attainable under only a mild monotonicity assumption, albeit via a computationally inefficient approach. When given access to an oracle for non-convex optimization, we obtain efficient $O(\sqrt{T})$ regret under arbitrary costs as an application of a novel algorithm for online non-convex optimization with memory. We identify a number of additional cases in which either $O(\sqrt{T})$ or $O(\log T)$ regret is attainable efficiently without an oracle, and we give negative results justifying our structural and computational assumptions. We show experimentally that our techniques enable consistent improvement over existing no-regret algorithms, across a range of portfolio sizes, when transaction costs are simulated from market data according to standard trade impact models.

1 Introduction

In financial portfolio optimization, the unpredictable nature of price movements motivates the design of algorithms which achieve strong performance guarantees without making statistical assumptions on asset returns. The "universal portfolio" setting introduced by Cover [1991] considers the problem of competing with the returns obtained by the best *constant rebalanced portfolio* in hindsight over a period of T rounds. This setting can also be formulated as an instance of online convex optimization (OCO) over log-returns, where algorithms which minimize regret over per-round concave reward functions have been studied extensively (see Li and Hoi [2013] for a broad overview, and Van Erven et al. [2020] for a recent discussion of optimal rates). An important consideration in portfolio optimization is the role of transaction costs; there is some prior work extending this setting to account for transaction costs [Blum and Kalai, 1997, Györfi and Vajda, 2008, Vittori et al., 2020, Guo et al., 2021], but to date these works consider only fixed percentage fee models or approximations thereof. However, for large portfolios managed by institutional investors, the dominant contributing factor to transaction costs is typically the market impact of trading rather than broker fees; these impact costs are affected by trade volume as well as market conditions, and may change unpredictably across time, assets, and portfolio sizes [Almgren et al., 2005, Park et al., 2016, Qiao et al., 2023, Li et al., 2023]. As such, existing no-regret methods are insufficient to properly account for transaction costs in portfolio optimization for large institutions.

In this paper, we give the first (to our knowledge) algorithms for online portfolio selection which obtain regret guarantees in the presence of adversarially changing and nonparametric transaction costs. We formalize this problem via the framework of online (non-convex) optimization with *memory*, wherein objective functions can depend on multiple prior actions [Arora et al., 2012, Anava et al., 2015], thus enabling a unifying perspective for analyzing several algorithms differing their in computational requirements and structural assumptions. When transaction costs are present, the total return of a portfolio selection strategy depends not only on returns of the assets held and the costs of exploration, but also the costs of *rebalancing* a portfolio when aiming to maintain consistent percentage holdings (such as in the case of many ETFs), which can vary across assets and holding volumes due to volatility or liquidity. Regret minimization algorithms with continuous objectives often bear some resemblance to gradient-based optimization; at a high level, our approach to accommodating adversarial transaction costs in our algorithms is to incorporate appropriately calibrated gradient penalty terms for the rebalancing cost of the current portfolio at each step. Taken as a whole, our results show that time-varying nonparametric transaction costs can be accommodated in many contexts which previously only accommodated fixed proportional costs, with minimal tradeoffs.

1.1 Our Contributions

Problem formulation. We show that online portfolio selection over n assets with adversarial transaction costs can be formulated as an instance of online non-convex optimization with memory, where the per-round objective decomposes into standard log-return as well as an additive term corresponding to log-cost.

Optimal rates. We give an algorithm inspired by the approaches from Cover [1991] and Blum and Kalai [1997], adapted to the setting of memory-dependent and non-parametric rewards; this algorithm obtains regret $O(n \log T)$ in the presence of adversarial transaction costs under a mild monotonicity assumption, which is optimal even in the *absence* of transaction costs, but is not computationally efficient.

Oracle-efficient methods. We give an $O(\sqrt{T})$ regret algorithm for online non-convex optimization with memory, which is efficient when given access to an oracle for offline non-convex optimization. This can be applied directly to our setting to yield an equivalent bound for arbitrary Lipschitz transaction costs.

Improved rates and computation. We identify a number of additional conditions under which either computational efficiency or rates of regret can be improved when adversarial transaction costs are present. We show that efficient $O(\sqrt{T})$ regret is feasible when considering a set of time-varying investment strategies (e.g. from a pool of financial analysts), or when joint log-returns in terms of *rebalancing* costs are concave. In the latter case, rates can be improved to $O(\log T)$ when a risk-aversion penalty for encouraging portfolio diversity is included each round. We also highlight examples which suggest barriers to further improvements and discuss remaining open questions.

Experimental evaluation. We conduct extensive empirical evaluation of our algorithms in comparison to previous no-regret methods using returns and transaction costs simulated from historical S&P500 data. We simulate time-varying transaction costs according to the model from Almgren et al. [2005], which estimates the market impact of a given trade in terms of price volatility and intra-day trading volume (as a proportion of market cap). We consider a range of asset subsets as well as portfolio sizes, and find that our algorithms which incorporate time-varying transaction costs consistently outperform other baselines, even in regimes without formal guarantees.

1.2 Related Work

Regret minimization for portfolio selection. Following the algorithm of Cover [1991], whose $O(n \log T)$ rate was shown to be optimal by Cover and Ordentlich [1996], there has been a long line of work aiming to design algorithms for this setting (without transaction costs) which improve along various lines, most notably in terms of computational efficiency. Whereas the computation of Cover [1991] scales exponentially in the number of assets, subsequent algorithms [Kalai and Vempala, 2000, Hazan et al., 2007] have improved this to a total computation time which is polynomial in all parameters while maintaining comparable regret guarantees, recently culminating in an algorithm from Jézéquel et al. [2022] which obtains near-optimal regret $O(d \log(T + d))$ with $\tilde{O}(d^2T^2 + d^3T)$ total computation. There are also a number of algorithms [Helmbold et al., 1998, Orseau et al., 2019] which obtain total computation $O(dT)$ (and often improved empirical performance) in exchange for a regret bound of only $\tilde{O}(\sqrt{T})$. These results are surveyed in Li and Hoi [2013] and Van Erven et al. [2020].

A frequent consideration for online portfolio selection is whether reward gradients must be bounded (corresponding to e.g. strictly positive and finite returns), as is assumed by several of the aforementioned works. We make this assumption in Sections 4 and 5, but do not require it in Section 3.

Online portfolios with transaction costs. The consideration of transaction costs in the online portfolio setting was initiated in Blum and Kalai [1997], where the approach from Cover [1991] was extended to accommodate fixed percentage transaction fees with a similar rate of regret. An algorithm from Uziel and El-Yaniv [2020] obtains $O(\sqrt{T})$ regret with respect to a set of *expert* portfolio strategies in the proportional cost model; efficient algorithms with $O(\sqrt{T})$ regret over constant rebalanced portfolios are given by Das et al. [2013] and Vittori et al. [2020], yet rather than direct proportional costs, approximations $c_t(\mathbf{x}_t, \mathbf{x}_{t-1}) \propto \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_1$ are used, which ignore the shift in holdings due to returns $\mathbf{r}_t$ and whose guarantees only hold directly when each daily return is considerably close to 1. Beyond the context of adversarial regret minimization, there are a number of algorithms in the proportional cost model [Sousa Lobo et al., 2007, Györfi and Vajda, 2008, Shen et al., 2014, Bin Li and Hoi, 2018, Guo et al., 2021, Tsang et al., 2022] whose performance is evaluated either empirically or with distributional assumptions over returns. To our knowledge, there are no prior algorithms which obtain sublinear regret guarantees over constant rebabalanced portfolios with either non-proportional or time-varying transaction costs, even when computational efficiency is not a concern.

Other approaches to transaction costs. Many aspects of portfolio optimization are widely studied beyond the context of adversarial regret minimization, often under the assumption of stochastic returns and mean-variance performance criteria. Of particular relevance to our work is the literature on transaction cost modeling, where theoretical and data-driven approaches are given for estimating the market impact of large trades [Almgren and Chriss, 2000, Almgren et al., 2005, Park et al., 2016, Benzaquen and Bouchaud, 2017, Li et al., 2023].

OCO with memory and oracle methods. The framework of online convex optimization with memory [Arora et al., 2012, Anava et al., 2015], which has been applied to problems ranging from statistical arbitrage [Anava et al., 2015] to online control [Agarwal et al., 2019, Gradu et al., 2020]. To date, these approaches consider only convex (surrogate) losses. A parallel line of work has considered online non-convex optimization via oracles, which culminated in an optimal $O(\sqrt{T})$ algorithm from Suggala and Netrapalli [2019]. We leverage both of these techniques in Section 4, obtaining the first algorithm no-regret algorithm for online non-convex optimization with memory, which we then apply to our portfolio selection problem.

1.3 Notation

We use $\|\cdot\|$ to indicate the ℓ_1 norm unless denoted otherwise, and we let $\Delta(n)$ denote the simplex over n items. We use $\mathbf{u}_n$ to denote the uniform distribution and e.g. $\mathbf{1}$ for the all-ones vector. $\otimes$ denotes elementwise multiplication. $O(\cdot)$ is the standard asymptotic notation, and $\tilde{O}(\cdot)$ hides polylogarithmic terms.

2 Setting

In the classic setting of online portfolio selection from Cover [1991], at each round $t \in [T]$ we choose some allocation $\mathbf{x}_t \in \Delta(n)$ over a set of given a set of n assets. An adversary chooses per-round price vectors $\mathbf{p}_0, \ldots, \mathbf{p}_T \in \mathbb{R}^n_+$, with the *return* vectors $\mathbf{r}_t$ then given by $\mathbf{r}_t(i) = \frac{\mathbf{p}_t(i)}{\mathbf{p}_{t-1}(i)}$ for each asset. We begin with total wealth $W_0 = 1$ at $t = 0$, and in the absence of transaction costs, our total wealth after round T is

$$W_T = \prod_{t=1}^{T} \mathbf{r}_t^\top \mathbf{x}_t \,.$$

This can be cast as an instance of online (convex) optimization by minimizing regret over log returns

$$\text{Reg}_{\text{no-cost}}(T) = \max_{\mathbf{x}^* \in \Delta(n)} \sum_{t=1}^{T} \log(\mathbf{r}_t^\top \mathbf{x}^*) - \sum_{t=1}^{T} \log(\mathbf{r}_t^\top \mathbf{x}_t)$$

with respect to the the best *constant rebalanced portfolio* $\mathbf{x}^* \in \Delta(n)$, i.e. the trading strategy where the portfolio is rebalanced to $\mathbf{x}^*$ after each round.

We augment this to allow for adversarial transaction costs in each round, given by functions c_t of both current and previous holdings. as well as previous returns. Following a round $t-1$ where we play the allocation $\mathbf{x}_{t-1}$ and observe returns $\mathbf{r}_{t-1}$, let $\mathbf{x}_t$ be our target portfolio for round t. Upon updating our holdings to $\mathbf{x}_t$, we incur a cost $c_t(\mathbf{x}_t, \mathbf{x}_{t-1})$ per unit of wealth in our portfolio, where the function c_t may implicitly depend on $\mathbf{r}_{t-1}$ as well as other adversarially determined features related to market conditions or earlier returns, and need not satisfy a particular functional form, but we assume it is C-Lipschitz in both arguments. We denote our wealth under the portfolio $\mathbf{x}_t$ (prior to observing returns $\mathbf{r}_t$) by $\widehat{W}_t$, where we have that

$$\frac{\widehat{W}_t}{W_{t-1}} = 1 - c_t(\mathbf{x}_t, \mathbf{x}_{t-1}).$$

Then, for a sequence of allocations $\mathbf{x}_0, \ldots, \mathbf{x}_T$ our resulting wealth is given by

$$W_T = \prod_{t=1}^{T} \mathbf{r}_t^\top \mathbf{x}_t \cdot (1 - c_t(\mathbf{x}_t, \mathbf{x}_{t-1}))$$

and our regret (in log-return scale) with respect to the best constant rebalanced portfolio $\mathbf{x}^*$ is given by

$$\begin{aligned}\text{Reg}(T) = \max_{\mathbf{x}^* \in \Delta(n)} & \sum_{t=1}^{T} \log(r_t^\top \mathbf{x}^*) + \log(1 - c_t(\mathbf{x}^*, \mathbf{x}^*)) \\ & - \sum_{t=1}^{T} \log(r_t^\top \mathbf{x}_t) + \log(1 - c_t(\mathbf{x}_t, \mathbf{x}_{t-1}))\end{aligned}$$

with $\mathbf{x}_0 = \mathbf{u}_n$. Upon the inclusion of transaction costs, our per-round log-return objectives may no longer be convex, and their dependence on multiple timesteps precludes the direct application of OCO algorithms. Our goal in this setting is to design algorithms which minimize this regret objective when both returns $\mathbf{r}_t$ and transaction costs c_t may be chosen adversarially.

3 Logarithmic Regret with Adversarial Transaction Costs

Here, we give our first algorithm which obtains the optimal logarithmic regret rate in terms of T. We make use of the following assumption on transaction costs, which enforces that price impact costs must increase monotonically in transaction size.

Assumption 1 (Cost Monotonicity). *Given a function c_t, the total transaction cost can be decomposed into per-asset costs (which are zero if no trading occurs), and the proportional cost-per-unit-traded (in terms of effective price) of an asset is monotonically increasing in the trade size (for buying or selling).*

We view this restriction as natural and quite mild; it yields similar properties to the set of assumptions made in Blum and Kalai [1997], and describes typical market behavior in which there is increasing demand to buy (or sell) at lower (higher) price levels. When this holds, we can simultaneously conduct rebalancing for many fractional portfolios and ensure that their observed costs are no larger than they would be for their non-fractional counterparts.

Algorithm 1 Universal Portfolios with Memory (UP-M)

```
Input: returns $\mathbf{r}_t$, costs $c_t(\mathbf{x}_t, \mathbf{x}_{t-1})$ for $t \in [T]$
Initialize $K = O(T^n)$ portfolios $\mathbf{x}^{(k)}$ uniformly distributed over $\Delta(n)$, allocate wealth $\frac{W_0}{K}$ to each
for $t = 1$ to $T$ do
    Observe returns $\mathbf{r}_t$, rebalance each fractional portfolio $\mathbf{r}_t^\top \otimes \mathbf{x}^{(k)}$ to $\mathbf{x}^{(k)}$, scheduling trades to yield proportional costs at most $1 - c_t(\mathbf{x}^{(k)}, \mathbf{x}^{(k)})$
end for
```

Theorem 1. *For a cost function c_t, For a sequence of returns $\mathbf{r}_t \in \mathbb{R}_+^n$ and C-Lipschitz cost functions c_t satisfying Assumption 1,* UP-M *obtains regret $O(n \log T)$ with respect to the log-return of the best constant rebalanced portfolio.*

Proof Sketch. Given a grid of $K = O(T^n)$ portfolios, for any $\mathbf{x}^*$ in $\Delta(n)$ we will have some $\mathbf{x}^{(k)}$ such that $\left\|\mathbf{x}^* - \mathbf{x}^{(k)}\right\| \leq \epsilon$, for $\epsilon = 1/(2C+1)T$. Consider the total returns of constant rebalanced portfolios $\mathbf{x}^*$ and an ϵ-close $\mathbf{x}^{(k)}$, which are given by:

$$W_T^* = \prod_t r_t^\top \mathbf{x}^* \cdot (1 - c_t(\mathbf{x}^*, \mathbf{x}^*));$$
$$W_T^{(k)} = \prod_t r_t^\top \mathbf{x}^{(k)} \cdot (1 - c_t(\mathbf{x}^{(k)}, \mathbf{x}^{(k)})).$$

As each c_t is C-Lipschitz, we then have that:

$$W_T^{(k)} \geq W_T^*(1 - (2C+1)\epsilon)^T \geq e^{-1} \cdot W_T^*.$$

Using Assumption 1, we can show that it suffices to consider $c_t(\mathbf{x}^{(k)}, \mathbf{x}^{(k)})$ as an upper bound for the transaction cost incurred by each fractional portfolio individually. We discuss this analysis and our trade scheduling approach in the appendix. We then have that our algorithm's total return is at least:

$$W_T^{\text{UP-M}} \geq \sum_{k \in [K]} \frac{W_T^{(k)}}{K} \geq \max_{k \in [K]} \frac{W_T^{(k)}}{K}$$

and so

$$\log(W_T^*) \leq \log(W_T^{\text{UP-M}}) + O(n \log T).$$

□

This rate is optimal even in the absence of transaction costs, as shown by Cover and Ordentlich [1996]. The algorithms of Cover [1991] and Blum and Kalai [1997] involve computing ratios of integrals at each step that express the limit of increasing the number of fractional portfolios, whose analysis has been has related to the *exp-concavity* of rewards by subsequent work [Hazan et al., 2007]. Perhaps surprisingly, our approach here does not even require that rewards be *concave*, which may not be the case even for proportional transaction costs (as we discuss in Section 5). While our computational and space complexity scales exponentially in the number of assets, we find experimentally in Section 6 that taking K substantially smaller is often sufficient when simulating returns and transaction costs from historical data.

4 Oracle-Efficient Portfolio Selection

Here we give algorithms for adversarial transaction which are computationally efficient when given access to an offline *oracle* for non-convex optimization, yet at the cost of only obtaining $O(\sqrt{T})$ regret. Our approach is to extend the analysis of non-convex Follow the Perturbed Leader from Suggala and Netrapalli [2019] to the setting of optimization with memory, wherein losses may depend on $m \geq 1$ additional previous actions, which we then use to optimize over constant rebalanced portfolios.

4.1 Non-Convex FTPL with Memory

We say that $\mathcal{O}_{\alpha,\beta}$ is an (α, β)-approximate optimization oracle if, when given a function $f : \mathcal{X} \to \mathbb{R}$ and a vector $\sigma \in \mathbb{R}^d$, it returns a point $\mathbf{x}^* \in \mathcal{X}$ satisfying

$$f(\mathbf{x}^*) - \langle \sigma, \mathbf{x}^* \rangle \leq \inf_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) - \langle \sigma, \mathbf{x}^* \rangle + \alpha + \beta \left\|\sigma\right\|_1.$$

Given such an oracle, we show the classical FTPL algorithm from Kalai and Vempala [2005] can simultaneously accommodate non-convexity and *memory* (dependence on $\mathbf{x}_t, \ldots, \mathbf{x}_{t-m}$) in our per-round objectives. Following

Algorithm 2 FTPL with Memory (FTPL-M)

```
Input: Loss functions with $m$-step memory $\{f_t\}_{t=1}^T$, exponential distribution parameter $\eta$, approximate optimization oracle $\mathcal{O}_{\alpha,\beta}$ over $\Delta(n)$
Choose $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{X}$ arbitrarily
for $t = m$ to $T$ do
    Sample $\sigma_t \in \mathbb{R}_+^n$ with each $\sigma_{t,i} \sim \text{Exp}(\eta)$ i.i.d
    Predict $\mathbf{x}_t$ as
        $$\mathbf{x}_t = \mathcal{O}_{\alpha,\beta}\left(\sum_{j=m}^{t-1} f_j(\mathbf{x}, \dots, \mathbf{x}) + \langle \sigma_t, \mathbf{x}\rangle\right)$$
    Observe loss function $f_t$
end for
```

the approach of Anava et al. [2015], our analysis proceeds by considering memoryless *surrogate* losses $\tilde{f}_t(\mathbf{x}_t, \dots, \mathbf{x}_t)$ in place of the true memory-dependent losses $f_t(\mathbf{x}_t, \dots, \mathbf{x}_{t-m})$, wherein the current action $\mathbf{x}_t$ is included for each entry of f_t, and then bounding the difference between these sets of losses.

Theorem 2. *For a sequence of L-Lipschitz losses with m-round memory $\{f_t\}_{t=1}^t$, and an oracle $\mathcal{O}_{\alpha,\beta}$ where $\alpha = O(\sqrt{1/T})$ and $\beta = O(1/T)$,* FTPL-M *obtains expected regret at most*

$$\mathbb{E}\left[\text{Reg}(T)\right] \le O\left(mn^{1.5}\sqrt{T}\right).$$

with T calls to $\mathcal{O}_{\alpha,\beta}$.

Proof Sketch. The existing analysis of FTPL can be directly applied to the surrogate losses $\tilde{f}_t(\mathbf{x}_t, \dots, \mathbf{x}_t)$ rather than $f_t(\mathbf{x}_t, \dots, \mathbf{x}_{t-m})$, and so the key step is to show that surrogate regret closely tracks true regret. We accomplish this by leveraging a stability argument proved in the analysis of Suggala and Netrapalli [2019], which gives a bound on $\mathbb{E}[\|\mathbf{x}_t - \mathbf{x}_{t-1}\|]$ in terms of η. Appropriately calibrating η yields the regret guarantee. $\square$

For our setting of optimizing over constant-rebalanced portfolios with adversarial transaction costs, we have $m = 1$ and can directly apply FTPL-M whenever our objectives are Lipschitz. We show that this holds whenever returns are bounded within a multiplicative factor of $R > 1$ from 1, which generally holds true in financial markets particularly for larger-cap assets.

Corollary 2.1. *For returns $\mathbf{r}_t \in [1/R, R]^n$ and C-Lipschitz costs c_t with value in $[0, \frac{1}{2}]$,* FTPL-M *obtains regret bounded by $O(\sqrt{T})$ with T calls to $\mathcal{O}_{\alpha,\beta}$.*

5 Improving Computation and Regret

Each of our results in Sections 3 and 4 is either not computationally efficient or requires oracle access for non-convex optimization. Here, we give a set of algorithmic results which are efficient without oracles, albeit under alternate assumptions on our regret objectives, and which enable $O(\log T)$ regret in certain cases. We also provide negative examples which highlight the difficulties of both efficient regret minimization and fast rates when transaction costs are present.

5.1 Optimizing over Experts with Transaction Costs

Here we consider a setting where our goal is to compete with a pool of S *experts* (who themselves also incur transaction costs), perhaps corresponding to a pool of financial analysts whose updating investment recommendations we are evaluating, rather than the best constant rebalanced portfolio. A similar goal is considered by Uziel and El-Yaniv [2020] in the presence of proportional transaction costs, and in a two-timescale model with "long-term" and "short-term" experts, where the short-term have access to additional market information for predicting future returns. Our model and objective is simpler, and we show that the performance best expert can be tracked when arbitrary bounded transaction costs are present via reduction to online prediction with *switching costs* [Geulen et al., 2010, Cesa-Bianchi et al., 2013, Gofer, 2014].

Theorem 3. *Given a set of S experts, where each $s \in S$ specifies a portfolio $\mathbf{x}_{s,t}$ at each step, there is an efficient algorithm which obtains regret $O(\sqrt{T})$ with respect to the return of the best expert, given by*

$$\max_{s \in S} \sum_{i=1}^{T} \log(\mathbf{r}_t^\top \mathbf{x}_{s,t}) + \log(1 - c_t(\mathbf{x}_{s,t}, \mathbf{x}_{s,t-1})).$$

Here, the difficulty of optimizing over non-convex transaction cost functions is essentially "pushed" onto the experts, who may not optimize perfectly, and we simply need to avoid excess transaction costs from expert switches by using any low-switching algorithm.

5.2 OGD with Memory for Concave Log-Return Objectives

For online optimization problems with m-step memory, Anava et al. [2015] show that Online Gradient Descent obtains sublinear regret bounds whenever surrogate losses $\tilde{f}_t(\mathbf{x}) = f_t(\mathbf{x}, \dots, \mathbf{x})$ are (strongly) convex and Lipschitz in $\mathbf{x}$ (fixed for all inputs). In our context, this corresponds to requiring (strongly) concave reward objectives $\log(\mathbf{r}_t^\top \mathbf{x}_t) + \log(1 - c_t(\mathbf{x}_t, \mathbf{x}_t))$ in terms of the rebalancing cost for a static portfolio. Here we restate their algorithm and its guarantees.

Algorithm 3 OGD with Memory (OGD-M)

```
Input: learning rate schedule $\{\eta_t\}_{t=1}^T$, loss functions with 1-step memory $\{f_t\}_{t=1}^T$
Choose $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{u}_n$
for $t = 2$ to $T$ do
    Play $\mathbf{x}_t$ and observe reward $f_t(\mathbf{x}_t, \mathbf{x}_{t-1})$
    Set $\mathbf{x}_{t+1} = \Pi_{\Delta(n)}\left(\mathbf{x}_t + \eta_t \nabla \tilde{f}_t(\mathbf{x}_t)\right)$
end for
```

Proposition 1 (Anava et al. [2015])**.** OGD-M *with appropriately calibrated learning rates obtains regret $O(\sqrt{T})$ for convex Lipschitz losses with constant memory, and $O(\log T)$ for strongly convex Lipschitz losses with constant memory.*

Corollary 3.1. *For any returns $\mathbf{r}_t$ and transaction cost functions c_t which yield (strongly) concave and Lipschitz surrogate rewards $\tilde{f}_t(\mathbf{x})$,* OGD-M *obtains $O(\sqrt{T})$ (or $\log T$) regret with respect to the best constant rebalanced portfolio.*

In Section 5.3, we give conditions under which efficient $O(\log T)$ regret can be attained, relating to objectives considered in mean-variance portfolio optimization.

5.3 Efficient Logarithmic Regret for Risk-Sensitive Portfolios

In mean-variance portfolio optimization, one typically assumes that returns at each step are stochastic (and perhaps correlated) in contrast to the adversarial setting where returns can be arbitrary, yet real-world financial markets often lie somewhere in between these regimes in practice. In particular, one could consider returns with both stochastic and adversarial components, where correlations between stochastic components persist over time. While negative results are known for mean-variance online learning in general [Even-Dar et al., 2006], algorithms which directly optimize for risk-aversion are widely explored in cost-aware settings nonetheless [Shen et al., 2014, Guo et al., 2021, Qiao et al., 2023]. As such, a natural criterion (similar to that in [Even-Dar et al., 2006], extended to portfolio selection) is to maximize total returns in conjunction with exogenously imposed penalty terms which encourage risk-aversion. For standard measures of portfolio diversity such as entropy, or the variance of $\mathbf{x}^\top A \mathbf{x}$ with some positive-definite correlation matrix A, the corresponding penalty term will be *strongly* concave. We consider the case where such a strongly concave penalty $p(\mathbf{x})$ is added to each log-return.

While algorithms for portfolio optimization generally rely on *exp-concavity* to obtain fast $O(\log T)$ rates, a key benefit of instead leveraging strong concavity when faced with transactions costs is due to its robustness under additive composition. The per-round objective functions we obtain will be strongly convex in the absence of transaction costs, and with sufficient curvature to often "overpower" the non-convexity introduced by transaction costs when these costs have Lipschitz gradients, yielding a composite objective which is still strongly concave (albeit with a weaker parameter). We first show this composition property.

Lemma 2. *Suppose $f(\mathbf{x})$ is μ-strongly convex and $g(\mathbf{x})$ has gradients $\nabla g(\mathbf{x})$ which are λ-Lipschitz with respect to the ℓ_2 norm, for $\mathbf{x} \in \Delta(n)$. Then, $f(\mathbf{x}) + g(\mathbf{x})$ is $(\mu - \lambda)$-strongly convex whenever $\mu - \lambda > 0$.*

Proof. By strong convexity of f, for any $\mathbf{x}, \mathbf{y} \in \Delta(n)$,

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 .$$

By the Lipschitz gradient condition on g, we also have

$$g(\mathbf{y}) - g(\mathbf{x}) \geq \nabla g(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) - \frac{\lambda}{2} \|\mathbf{y} - \mathbf{x}\|_2^2$$

and so

$$\begin{aligned} f(\mathbf{y}) + g(\mathbf{y}) &\geq f(\mathbf{x}) + g(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) \\ &\quad + \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + (g(\mathbf{y}) - g(\mathbf{x})) \\ &\geq f(\mathbf{x}) + g(\mathbf{x}) + \nabla (f(\mathbf{x}) + g(\mathbf{x}))^\top (\mathbf{y} - \mathbf{x}) \\ &\quad + \frac{\mu - \lambda}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 , \end{aligned}$$

yielding $(\mu - \lambda)$-strong convexity for $f(\mathbf{x}) + g(\mathbf{x})$. □

Using this, we show that the inclusion of a risk penalty term at each round enables fast rates of regret when using OGD-M even when rebalancing costs are non-convex, provided that the transaction cost functions are sufficiently smooth.

Theorem 4. *Suppose each cost function $c_t(\mathbf{x}, \mathbf{x})$ is C-Lipschitz, has λ-Lipschitz gradients, and is at most $\frac{1}{2}$. Further, suppose each round's objective is given by $\log(\mathbf{r}_t^\top \mathbf{x}_t) + \log(1 - c_t(\mathbf{x}_t, \mathbf{x}_{t-1})) + g(\mathbf{x}_t)$, where g is a μ-strongly convex and G-Lipschitz risk penalty term with sufficiently large μ. Then,* OGD-M *obtains regret $O(\log T)$ with respect to the best constant-rebalanced risk-penalized portfolio.*

5.4 Non-Concavity of Proportional Costs

Here we give evidence of the inherent difficulty of portfolio optimization with transaction costs by showing that even the simple case of proportional costs above 0 can yield a *non-concave* objective to be maximized. For such transactions costs, a percentage fee C is applied to the total trading volume, which is equivalent to paying in terms of the total variation distance between portfolios and yields a cost function given each round by

$$c_t(\mathbf{x}_t, \mathbf{x}_{t-1}) = \frac{C}{2} \left\| \mathbf{x}_t - \frac{\mathbf{r}_{t-1} \otimes \mathbf{x}_{t-1}}{\mathbf{r}_{t-1}^\top \mathbf{x}_{t-1}} \right\|_1$$

for some $C \in (0, 1]$. Notably, we show that proportional transaction costs result in a non-concave objective for log-returns even for percentage costs C arbitrarily close to 0 and daily returns arbitrarily close to 1. Intuitively, this stems from the fact that more diversified portfolios will require larger transaction costs in order to rebalance each round, whereas a portfolio entirely concentrated on a single asset (or a set of highly correlated assets) will never (rarely) rebalance.

Theorem 5. *Suppose transaction costs are given by* $c_t(\mathbf{x}_t, \mathbf{x}_{t-1}) = \frac{C}{2} \left\| \mathbf{x}_t - \frac{\mathbf{r}_{t-1} \otimes \mathbf{x}_{t-1}}{\mathbf{r}_{t-1}^\top \mathbf{x}_{t-1}} \right\|_1$*, i.e. a fixed percentage is charged in terms of the total volume which must be rebalanced. For any* $C > 0$*, there is a sequence of returns* $\{\mathbf{r}_t\}_{t=1}^T$ *for which the log-return objective* $\sum_{t=1}^T \log(\mathbf{r}_t^\top \mathbf{x}) + \log(1 - c_t(\mathbf{x}, \mathbf{x}, \mathbf{r}_{t-1}))$ *over constant rebalanced portfolios* $\mathbf{x} \in \Delta(n)$ *is non-concave in* $\mathbf{x}$*.*

Proof. Consider two assets a and b with returns $\mathbf{r}_t = (\mathbf{r}_t^{(a)}, \mathbf{r}_t^{(b)})$ given by

$$\mathbf{r}_t = \begin{cases} (w, \frac{1}{w}), & t \text{ is even} \\ (\frac{1}{w}, w), & t \text{ is odd} \end{cases}$$

for some $w > 1$, and transaction costs given by

$$c_t(x_t, x_{t-1}) = \frac{\gamma}{2} \left\| x_t - \frac{r_{t-1} \otimes x_{t-1}}{r_{t-1}^\top x_{t-1}} \right\|$$

for some $\gamma \in (0, 1]$. Let $\mathbf{x}^{(a)} = (1, 0)$ be the portfolio entirely concentrated on asset a, let $\mathbf{x}^{(b)} = (0, 1)$ be the portfolio entirely concentrated on asset b, and let $\mathbf{x}^{(u)} = (\frac{1}{2}, \frac{1}{2})$ be the uniform portfolio. For initial wealth $W_0 = 1$ and any even T, it is straightforward that both $\mathbf{x}^{(a)}$ and $\mathbf{x}^{(b)}$ result in final wealth $W_T = 1$ for any C, as no rebalancing ever occurs. For $\mathbf{x}^{(u)}$, a per-round return $\mathbf{r}_t^\top \mathbf{x}^{(u)} = \frac{w}{2} + \frac{1}{2w}$ is observed upon rebalancing to $\mathbf{x}^{(u)}$ from $(\frac{w^2}{w^2+1}, \frac{1}{w^2+1})$, incurring a transaction cost of

$$\begin{aligned} c_t(\mathbf{x}^{(u)}, \mathbf{x}^{(u)}) &= \frac{C}{2} \left\| \mathbf{x}^{(u)} - \frac{\mathbf{r}_{t-1} \otimes \mathbf{x}^{(u)}}{\mathbf{r}_{t-1}^\top \mathbf{x}^{(u)}} \right\| \\ &= C \left(\frac{w^2}{w^2+1} - \frac{1}{2} \right). \end{aligned}$$

For any $w \in (1, \frac{2+C}{2-C})$, defining R^* as $\mathbf{r}_t^\top \mathbf{x}^{(u)}(1 - c_t(\mathbf{x}^{(u)}, \mathbf{x}^{(u)}))$, we then have

$$\begin{aligned} R^* &= \left(\frac{w}{2} + \frac{1}{2w} \right) \left(1 - C \left(\frac{w^2}{w^2+1} - \frac{1}{2} \right) \right) \\ &= \frac{1}{2w} \left(w^2 + 1 \right) \left(1 + \frac{C}{2} - \frac{Cw^2}{w^2+1} \right) \\ &= \frac{w(2-C)}{4} + \frac{2+C}{4w} \\ &< 1 \end{aligned}$$

with equality holding for positive w only at 1 and $\frac{2+C}{2-C}$ (assuming $C < 2$). As such, the final wealth of $\mathbf{x}^{(u)}$ given by $W_T^{(u)} = \prod_{t=1}^T R^*$ is strictly below 1. We have that both $\log(W_T^{(u)}) < \log(W_T^{(a)}) = 0$ and $\log(W_T^{(u)}) < \log(W_T^{(b)}) = 0$ by monotonicity of the log function, which implies that the log-return objective is non-concave over constant rebalanced portfolios $\mathbf{x} \in \Delta(n)$. For w which is bounded away from both 1 and $\frac{2+C}{2-C}$, and sufficiently large T, this holds even if transaction costs for rebalancing only occur in $T - 1$ rounds (e.g. if we take $\mathbf{r}_0 = \mathbf{1}$). □

This suggests that we likely cannot apply online convex optimization techniques to obtain efficient sublinear regret, even in the case of proportional transaction costs, without additional assumptions. Further, even when optimization oracles are accessible, it seems quite difficult to improve over $O(\sqrt{T})$ regret when transaction cost functions can be arbitrary: there are $\Omega(\sqrt{T})$ lower bounds known for smooth non-convex optimization [Suggala and Netrapalli, 2019], and it is unlikely that UP-M or the integration-based methods from Cover [1991] and Blum and Kalai [1997] can be simulated efficiently with bounded-precision optimization oracles when c_t can be arbitrary, as several numerical integration problems are known to be #P-complete (and thus likely outside of NP).

6 Experiments

We implement and simulate several of our algorithms and compare their performance to various benchmarks using returns and transaction costs estimated from historical market data for S&P500 stocks.

Simulating transaction costs. The model of Almgren et al. [2005] gives a functional expression for estimating the market impact and corresponding transaction cost of conducting a large trade in terms of price volatility and liquidity relative to an asset's market cap, with a parametric form and coefficients estimated from historical trade-level data. We compute a representation this expression for daily transaction costs which can be programmatically differentiated and optimized over.

Algorithms and baselines. We evaluate the following set of portfolio selection strategies:

- OGD-M with Almgren cost penalties, proportional cost penalties, and no penalties
- OGD-M with proportional cost penalties
- OGD (no penalties)
- FTPL-M with Almgren cost penalties
- UP-M with Almgren cost penalties
- Hedge for portfolios, under return rewards
- Hedge for sampling stocks, under return rewards
- Uniform constant-rebalanced portfolios

While there are indeed a wide range of other algorithms developed for portfolio selection, the bulk of the algorithms with no-regret guarantees we have discussed resemble those considered here (up to e.g. choice of regularizer or learning rate schedule). We conduct hyperparameter optimization over learning rates for OGD variants, and observed that choice of regularizer has minimal impact on practical performance for our data. Our primary aim experimentally is to evaluate the impact of incorporating transaction cost functions into algorithms directly (when these costs determine true performance) rather than omitting or approximating with percentage fees, and to compare behavior across qualitative classes of algorithms.

Setup and experiment details. We collect data from Yahoo! Finance for S&P500 stocks, and compute transaction costs using the volume-dependent impact expression from Almgren et al. [2005], where we assume rebalancing occurs over the final 45 minutes of each trading day. We run algorithms for $T = 1000$ days (steps) on trials of randomly chosen subets of 20 stocks, beginning on January 1, 2010, and average across subsets. We vary the initial wealth allocation of portfolios considered, and consider a range fixed percentage fees as cost estimates for baseline algorithms. Additional setup details are discussed in the appendix.

Results. We measure returns as W_T/W_0 across values of W_0, and find that incorporating transaction penalty terms for into algorithms reliably improves performance for portfolios up to \$100MM in initial assets when computing transaction costs according to the Almgren model; note that transaction volumes above this range are often larger than the regime of reliable accuracy of the model. Despite potential non-convexity of rewards, our results suggest that OGD-M with appropriate cost penalties may generally be more useful in practice than FTPL-M while still outperforming other baselines. Empirically, the runtime for each trial of FTPL-M scales as $O(T^2)$, owing to the computational bottleneck of averaging gradients across each past reward function, and further there is substantial variance across runs; results for FTPL-M at $W_0 = 1\text{e}9$ and sampling-based Hedge (where a single stock is chosen each round) across all W_0 values were significantly worse than the other baselines and are omitted.

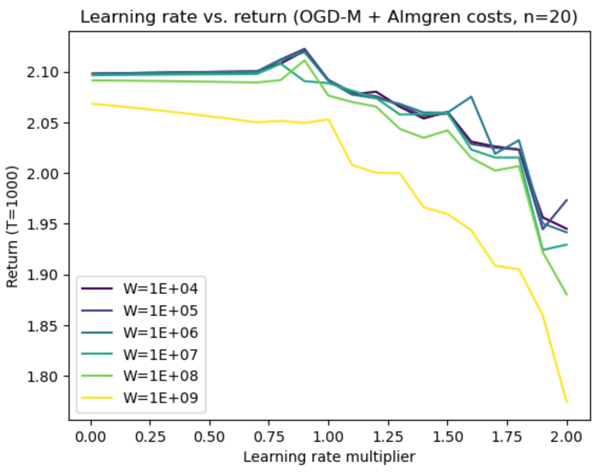

Figure 1: Optimizing over learning rates for OGD+M

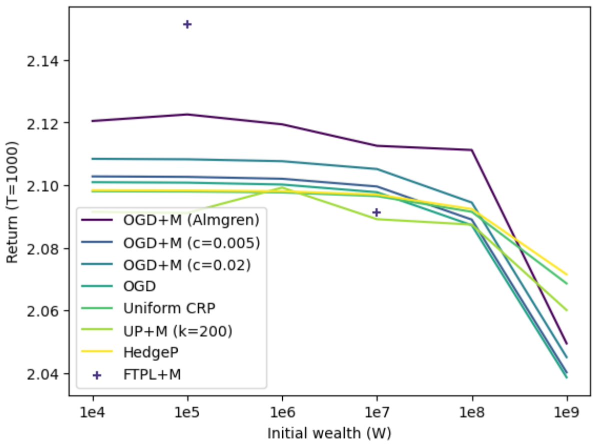

Figure 2: Initial wealth vs. algorithm performance

References

Naman Agarwal, Brian Bullins, Elad Hazan, Sham M. Kakade, and Karan Singh. Online control with adversarial disturbances. *CoRR*, abs/1902.08721, 2019.

Robert Almgren and Neil A Chriss. Optimal execution of portfolio trans-actions. 2000.

Robert Almgren, Chee Thum, Emmanuel Hauptmann, and Hong Li. Direct estimation of equity market impact. *RISK*, 18, 04 2005.

Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: Price of past mistakes. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret, 2012.

Michael Benzaquen and Jean-Philippe Bouchaud. Market impact with multi-timescale liquidity, 2017.

Dingjiang Huang Bin Li, Jialei Wang and Steven C. H. Hoi. Transaction cost optimization for online portfolio selection. *Quantitative Finance*, 18(8):1411–1424, 2018. doi: 10.1080/14697688.2017.1357831.

Avrim Blum and Adam Kalai. Universal portfolios with and without transaction costs. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, COLT '97, page 309–313, New York, NY, USA, 1997. Association for Computing Machinery. ISBN 0897918916. doi: 10.1145/267460.267518.

Nicolo Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries, 2013.

Thomas M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991. doi: https://doi.org/10.1111/j.1467-9965.1991.tb00002.x.

T.M. Cover and E. Ordentlich. Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2):348–363, 1996. doi: 10.1109/18.485708.

Puja Das, Nicholas Johnson, and Arindam Banerjee. Online lazy updates for portfolio selection with transaction costs. *Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013*, 27:202–208, 06 2013. doi: 10.1609/aaai.v27i1.8693.

Eyal Even-Dar, Michael Kearns, and Jennifer Wortman. Risk-sensitive online learning. In José L. Balcázar, Philip M. Long, and Frank Stephan, editors, *Algorithmic Learning Theory*, pages 199–213, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-46650-5.

Sascha Geulen, Berthold Vöcking, and Melanie Winkler. Regret minimization for online buffering problems using the weighted majority algorithm. volume 17, pages 132–143, 10 2010.

Eyal Gofer. Higher-order regret bounds with switching costs. In Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors, *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 210–243, Barcelona, Spain, 13–15 Jun 2014. PMLR.

Paula Gradu, Elad Hazan, and Edgar Minasyan. Adaptive regret for control of time-varying dynamics. *CoRR*, abs/2007.04393, 2020.

Sini Guo, Jia-Wen Gu, and Wai-Ki Ching. Adaptive online portfolio selection with transaction costs. *European Journal of Operational Research*, 295(3):1074–1086, 2021. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2021.03.023.

László Györfi and István Vajda. Growth optimal investment with transaction costs. In *Algorithmic Learning Theory: 19th International Conference, ALT 2008, Budapest, Hungary, October 13-16, 2008. Proceedings*, page 108–122, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-87986-2. doi: 10.1007/978-3-540-87987-9_13.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007. ISSN 1573-0565. doi: 10.1007/s10994-007-5016-8.

David P. Helmbold, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998. doi: https://doi.org/10.1111/1467-9965.00058.

Rémi Jézéquel, Dmitrii M. Ostrovskii, and Pierre Gaillard. Efficient and near-optimal online portfolio selection, 2022.

A. Kalai and S. Vempala. Efficient algorithms for universal portfolios. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 486–491, 2000. doi: 10.1109/SFCS.2000.892136.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2004.10.016. Learning Theory 2003.

Bin Li and Steven C. H. Hoi. Online portfolio selection: A survey, 2013.

Fengpei Li, Vitalii Ihnatiuk, Ryan Kinnear, Anderson Schneider, and Yuriy Nevmyvaka. Do price trajectory data increase the efficiency of market impact estimation?, 2023.

Laurent Orseau, Tor Lattimore, and Shane Legg. Soft-bayes: Prod for mixtures of experts with log-loss. *CoRR*, abs/1901.02230, 2019.

Saerom Park, Jaewook Lee, and Youngdoo Son. Predicting market impact costs using nonparametric machine learning models. *PLOS ONE*, 11:e0150243, 02 2016. doi: 10.1371/journal.pone.0150243.

W. Qiao, D. Bu, A. Gibberd, Y. Liao, T. Wen, and E. Li. When "time varying" volatility meets "transaction cost" in portfolio selection. *Journal of Empirical Finance*, 73:220–237, 2023. ISSN 0927-5398. doi: https://doi.org/10.1016/j.jempfin.2023.06.006.

Weiwei Shen, Jun Wang, and Shiqian Ma. Doubly regularized portfolio with risk minimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1), Jun. 2014. doi: 10.1609/aaai.v28i1.8906.

Miguel Sousa Lobo, Maryam Fazel, and Stephen Boyd. Portfolio optimization with linear and fixed transaction costs. *Annals OR*, 152:341–365, 03 2007. doi: 10.1007/s10479-006-0145-1.

Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. *CoRR*, abs/1903.08110, 2019.

Man Yiu Tsang, Tony Sit, and Hoi Ying Wong. Adaptive robust online portfolio selection, 2022.

Guy Uziel and Ran El-Yaniv. Long-and short-term forecasting for portfolio selection with transaction costs. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Tim Van Erven, Dirk Van der Hoeven, Wojciech Kotłowski, and Wouter M. Koolen. Open problem: Fast and optimal online portfolio selection. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 3864–3869. PMLR, 09–12 Jul 2020.

Edoardo Vittori, Martino Luca, Francesco Trovò, and Marcello Restelli. Dealing with transaction costs in portfolio optimization: online gradient descent with momentum. pages 1–8, 10 2020. doi: 10.1145/3383455.3422531.

7 Proofs for Section 3

7.1 Additional Details for Theorem 1

To complete the proof, we discuss how the monotonicity assumption enables bounding each fractional portfolio's incurred per-round per-unit rebalancing cost in terms of $c_t(\mathbf{x}^{(k)}, \mathbf{x}^{(k)})$. Given a portfolio $\mathbf{x}_t$ fractionally allocated across initial portfolios $\mathbf{x}^{(k)}$, we first can have these portfolios "trade amongst themselves" for free without incurring transaction costs (as in Blum and Kalai [1997]) to reduce the total volume of market transactions, as we assume transaction costs are non-negative (although this does not affect worst-case guarantees). Then, we schedule rebalancing trades for buying or selling an asset j on a per-asset basis (as permitted by Assumption 1) for each portfolio $\mathbf{x}^{(k)}$ in proportion with the fraction of that portfolio which needs to be traded in or out of that asset (ignoring that portfolio's relative volume), e.g. each portfolio buys or sells a τ fraction of its holdings in some asset j in a round-robin fashion until its rebalancing for that asset is complete. By splitting trade sizes across portfolios $\mathbf{x}^{(k)}$ in this manner for arbitrarily small τ, this ensures that the cost paid by a fractional portfolio $\mathbf{x}^{(k)}$ to buy (or sell) an asset j is below the cost it would pay in the event that the entire holdings were concentrated on $\mathbf{x}^{(k)}$ (up to τ), as marginal cost is increasing on a per-unit basis.

8 Proofs for Section 4

8.1 Additional Details for Theorem 2

We recall the result from Suggala and Netrapalli [2019] for Follow the Perturbed Leader (FTPL); this is equivalent to our algorithm when $f_t(x_t, \dots, x_{t-m}) = \tilde{f}(x_t)$ for all t and x_t.

Proposition 3 (Theorem 1 from Suggala and Netrapalli [2019]). *Given access to an oracle $\mathcal{O}_{\alpha,\beta}$ with $\alpha = O(\sqrt{1/T})$ and $\beta = O(1/T)$, FTPL obtains expected regret*

$$\mathbb{E}[\mathrm{Reg}(T)] = O\left(\eta n^2 L^2 T + \frac{n}{\eta}\right)$$

with respect to the optimum $\mathbf{x}^* \in \mathcal{X}$ over non-convex Lipschitz losses $\tilde{f}_t$.

We also recall the following stability result used in the analysis of Suggala and Netrapalli [2019], wherein the expected distance between subsequent points $\mathbf{x}_t$ is bounded.

Proposition 4 (Stability Lemma from Suggala and Netrapalli [2019]). *When running FTPL as above, consecutive points satisfy:*

$$\mathbb{E}[\|\mathbf{x}_t - \mathbf{x}_{t-1}\|] = O\left(\eta n^2 L + \frac{\beta n}{\eta L}\right).$$

Suppose we have $\beta = O(T^{-1}m^{-2})$. We can then bound the regret of FTPL-M for losses with memory in terms of the error between true and surrogate losses:

$$\begin{aligned}
\mathrm{Reg}(T) &\leq \max_{\mathbf{x}^* \in \mathcal{X}} \sum_{t=m}^{T} \tilde{f}_t(\mathbf{x}_t) - \tilde{f}_t(\mathbf{x}^*) + \sum_{t=m}^{T} \left| \tilde{f}_t(\mathbf{x}_t) - f_t(\mathbf{x}_t, \dots, \mathbf{x}_{t-m}) \right| \\
&\leq O\left(\eta n^2 L^2 T + \frac{n}{\eta}\right) + m^2 L \sum_{t=m}^{T} [\|\mathbf{x}_t - \mathbf{x}_{t-1}\| \\
&= O\left(\eta n^2 L^2 T + \frac{n}{\eta}\right) + O\left(\eta n^2 m^2 L^2 T + \frac{\beta n m^2 T}{\eta}\right) \\
&= O\left(m n^{1.5} L \sqrt{T}\right)
\end{aligned}$$

upon setting $\eta = \frac{1}{Lm(nT)^{0.5}}$.

8.2 Additional Details for Corollary 2.1

Here we prove a Lipschitz condition on reward functions satisfying the given conditions on returns and costs.

Lemma 5. *Suppose each return vector $\mathbf{r}_t$ lies in $[1/R, R]^n$ for some $R \in \mathbb{R}_+$, and that each $c_t(\mathbf{x}_t, \mathbf{x}_{t-1})$ is C-Lipschitz in both $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ (with respect to the ℓ_1 norm). Then:*

- *each $f_t(\mathbf{x}_t, \mathbf{x}_{t-1})$ is $(R^2 + 2C)$-Lipschitz in $\mathbf{x}_t$ and $2C$-Lipschitz in $\mathbf{x}_{t-1}$ over $\Delta(n)$, and*
- *each $\tilde{f}_t(\mathbf{x})$ is $(R^2 + 4C)$-Lipschitz in $\mathbf{x}$ over $\Delta(n)$.*

Proof. First consider the concave function $\log(\mathbf{r}_t^\top \mathbf{x}_t)$ in isolation. Each partial derivative is given by

$$\frac{\partial}{\partial \mathbf{x}_t(i)} \log(\mathbf{r}_t^\top \mathbf{x}_t) = \frac{\mathbf{r}_t(i)}{\mathbf{r}_t^\top \mathbf{x}_t},$$

yielding

$$\nabla \log(\mathbf{r}_t^\top \mathbf{x}_t) = \frac{\mathbf{r}_t}{\mathbf{r}_t^\top \mathbf{x}_t}$$

as the gradient. The quantity $\mathbf{r}_t^\top \mathbf{x}_t$ lies in $[1/R, R]$ for $\mathbf{x}_t \in \Delta(n)$ and so $\nabla \log(\mathbf{r}_t^\top \mathbf{x}_t) \in [1/R^2, R^2]^n$ for any $\mathbf{x}_t$; as such, $\log(\mathbf{r}_t^\top \mathbf{x}_t)$ is R^2-Lipschitz with respect to the ℓ_1 norm.

Next, given that $c_t(\mathbf{x}_t, \mathbf{x}_{t-1})$ is C-Lipschitz in both $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$, we have that $c_t(\mathbf{x}, \mathbf{x})$ is $2C$-Lipschitz in x. As such, for the function $\log(1 - c_t(\mathbf{x}, \mathbf{x}))$ we have that

$$\nabla_{\mathbf{x}} \log(1 - c_t(\mathbf{x}, \mathbf{x})) = -\frac{\nabla_{\mathbf{x}} c_t(\mathbf{x}, \mathbf{x})}{1 - c_t(\mathbf{x}, \mathbf{x})},$$

and so $\log(1 - c_t(\mathbf{x}, \mathbf{x}))$ is $4C$-Lipschitz as $1 - c_t(\mathbf{x}, \mathbf{x}) \geq \frac{1}{2}$. By a symmetric argument, we have that $\log(1 - c_t(\mathbf{x}_t, \mathbf{x}_{t-1}))$ is $2C$-Lipschitz in both $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ as well. Summing Lipschitz terms yields the result. □

The corollary then follows by plugging this bound in to Theorem 2 with $m = 1$.

9 Proofs for Section 5

9.1 Proof of Theorem 3

Here, we assume the same conditions as in Section 4 (that returns lie in $[1/R, R]$, and that costs are C-Lipschitz and bounded away from 1), and focus primarily on rates in terms of T (all other dependencies will be polynomial). As such, both rewards and switching costs are bounded when casting this as an instance of expert prediction with switching costs. The theorem then follows from the existence of algorithms which simultaneously satisfy regret and switching number $O(\sqrt{T})$ [Geulen et al., 2010, Cesa-Bianchi et al., 2013, Gofer, 2014].

9.2 Proof of Corollary 3.1

Consider the following assumption on reward functions:

Assumption 2. *For each $\mathbf{r}_t$ and c_t, the function $\log(\mathbf{r}_t^\top \mathbf{x}) + \log(1 - c_t(\mathbf{x}, \mathbf{x}))$ is convex in $\mathbf{x}$.*

When this holds, we can apply result of Anava et al. [2015] for OGD-M directly to the (strongly) convex surrogate losses in our setting. Here we consider Lipschitz conditions in terms of the ℓ_2 norm, adding at most a $\sqrt{n}$ factor to the bound from 5.

9.3 Proof of Theorem 4

We can apply a similar analysis as that in Lemma 5 to show that the joint reward function has Lipschitz gradients, as the log function has Lipschitz gradients when inputs are bounded away from zero. By Lemma 2, for sufficiently large μ we have that each joint surrogate reward function is strongly convex, and thus we can apply Corollary 3.1 to obtain the result.

10 Additional Details for Experiments

We implement our algorithms using standard numerical optimization libraries in Python; we write our reward functions and compute their gradients using Autograd, and we use SLSQP as our oracle for FTPL-M. Each trial consists of running an algorithm over 20 randomly chosen S&P500 stocks over 1000 days (starting on January 1, 2010). On a 2021 Macbook Pro, trials for each algorithm besides FTPL-M run in under 20 seconds; trials for FTPL-M take approximately 10 minutes. Results for each configuration shown are averaged across 10 trials.