
OPTIMIZING TRANSITION STRATEGIES FOR SMALL TO MEDIUM SIZED PORTFOLIOS

A PREPRINT

Nakul Upadhya

Department of Mechanical and Industrial Engineering
University of Toronto
Toronto, ON, M5S 2E4
nakul.upadhya@mail.utoronto.ca

Alexandre Granzer-Guay

Department of Mechanical and Industrial Engineering
University of Toronto
Toronto, ON, M5S 2E4
alexandre.granzerguay@mail.utoronto.ca

April 27, 2023

ABSTRACT

This paper discusses the benefits of constrained portfolio turnover strategies for small to medium sized portfolio. We propose a dynamic multi-period model that aims to minimize transaction costs and maximize terminal wealth levels whilst adhering to strict portfolio turnover constraints. We demonstrate that utilizing our framework along with reasonably accurate forecasts can result in, on average, 0.10% greater gain than naively converting the portfolio on any given day (over the course of a 30 day horizon). In addition, the proposed model lends itself to a reformulation that makes use of the column generation algorithm that can be leveraged in specific cases to significantly reduce the solving time. All of the discussed experiments and presented results can be reproduced using our code at github.com/upadhyan/Portfolio-Changeover-Optimization.

Keywords Portfolio Optimization · Integer Programming · Machine Learning

1 Introduction

Many retail investor strategies found in the literature, make certain assumptions for convenience. In fact, it is most common to ignore a non-partial constraint and fixed transaction costs in lieu of convexity and near optimality. Though reasonable when the portfolios in question boast wealth levels well over \$1 Million, they start breaking down when wealth level start decreasing to “young” or “novice” investor levels (below \$500 000). In this paper we propose to explore this fact through a common and simple strategy. Suppose, an investor current holds a diversified portfolio. The market experiences a regime switch and the investor responds buy converting her portfolio either to a risk-averse strategy or aggressive strategy to capitalize on upward trends. Such a portfolio conversion may incur significant trading fees and value if the conversion is not executed correctly. We propose a multi-period dynamic model that find the optimal trades to execute for a prescribed portfolio conversion over a given time horizon. In doing so, we consider small budget investors and the constraints that comes with lower wealth levels, namely, transaction fees, timing, and simplicity.

The presented strategy is a multi-period mixed-integer model that creates a trade plan for a prescribed horizon time-frame. At each period, within the horizon, the strategy is recalculated and reassessed for changes in market behavior. Given that the strategy plan relies on a valuable forecast, it’s important to adapt to any unforeseen behavior. We demonstrate the effectiveness of our proposed method through a set of numerical experiments using real-world financial data and scenarios. Our results show that our strategy outperforms classical approaches by successfully minimizing trading costs and maximizing portfolio value.

This paper is separated into 6 key sections. We first introduce the literary background that motivates our thesis. Then, we present a succinct problem statement before introducing the reader to the primary component of our model - the multi-period transition framework. Finally, we present our experimental methodology in detail before presenting

compelling numerical results which support our proposal. We conclude this paper with thoughts and comments on the observed performance and, we close the discussion with possible avenues for future work.

2 Related Works

Most classical work in portfolio allocation theory revolves around algorithms that have certain pragmatic benefits while also possessing some numerical advantages [1952, 1964, 2000]. When such advantages are not present - non-convex models, cardinality models, mixed-integer models, to name a few - the approach will either involve a problem relaxation of sorts or a reasonable assumption. In the case of models that consider transaction costs, until more recently, the empirical method was to ignore or “convexify” transaction costs. Though some will ignore transaction costs altogether, the status quo seems to be the use of a convex formulation of transaction costs [2015, 2007, 2017]. Such methods consider a linear function for the transaction costs, usually based on trading volume, and embed it into the formulation as a constraint or a Lagrangian relaxation, like in Boyd [2017]. Alternatively, fixed or fixed and linear transaction costs are considered. In such cases, the authors, Lobo, Fazel, and Boyd [2007], argue that this leads to a difficult combinatorial formulation, albeit, with a desirable sparse trading behavior. The nature of the trading constraint promotes the use of fewer trades but involves complicated permutations, in particular, in the case of large portfolios. This is discussed in greater detail in the cited manuscript. Later, we demonstrate how we can circumvent usual relaxations of the combinatorial problem by employing a column generation algorithm fit to our particular problem.

In addition to transaction cost constraints, we choose to constrain our framework to trading non-partial shares. Portfolio weights are traditionally computed in reference to the total portfolio value. In practice this means there would be some deviation between the prescribed weights and the equivalent shares to be bought. In large scale portfolio, a difference of ± 1 share in a stock won’t affect the relative holding weight of that stock in the portfolio. On the other hand, for small to medium sized portfolio, adding or removing a share may drastically alter the relative holding weight of this same position. As a result, we cannot assume the equivalency between a portfolio with fractional shares and one without and, instead, must enforce a whole share policy. To the best of our knowledge this is an unusual assumption that has not led to any major publication [Wickremasinghe and Gupta, 2022].

The main thesis of this paper revolves around a case where an investor must convert a portfolio A to a portfolio B under some prescribed constraint, namely, budget constraints and time constraints. The common approach is to use a continuously moving target and moving current portfolio, as is done in Smith [1967]. The future portfolio is determined using a portfolio allocation optimization model and the current portfolio is optimized to aim for the future portfolio while reducing transaction costs. This same approach is also explored in a probabilistic manner, where Collin-Dufresne et al. [2020], makes use of a stochastic model to find the optimal trading plan (future portfolio) to adjust between different market regimes. There, the authors, are considering models with wealth levels above 2 million and assessing the benefits and costs of continuously adapting a portfolio to the underlying regime. Gârleanu and Pedersen [2013] demonstrate that, in dynamic environments, there is a benefit in employing a rolling horizon method when attempting to aim for a target portfolio. Though their approach focuses on a continuously varying target portfolio, we will demonstrate that this claim also applies to our use case. The referenced literature makes use of larger sized portfolios and methods that consider, at most, one period in the future. Though this is perfectly reasonable, we choose to focus on smaller portfolios in addressing the transition problem, as opposed to the optimization and transition problem. This distinction is important as we believe that most retail investors are rarely rebalancing their portfolios. In fact, we maintain that such investors do not have the time or resources to continuously assess the optimality of their portfolio and therefore attribute a low level of importance to portfolio optimization. Instead, they tend to prefer minimal friction (losses) when they do elect to rebalance or adjust their portfolios. We present this thesis clearly in the next section.

3 Formal Problem Definition

The principle idea behind the portfolio turnover strategy is simple when we consider the turnover problem over a single period. Given a set of stocks A and a starting portfolio P_0 , we aim to find a set of stock trades Z (which can be split into purchases z^+ and sells z^-) to transition our holdings to one that satisfies a target set of assets \mathcal{T} . In such a scenario the goal is to minimize transaction costs and maximize portfolio value while converting the portfolio to its target. This portfolio value is equivalent to the combined value of the stocks in the portfolio plus any leftover cash at the final time step C .

As we are aiming to replicate the environment faced by retail traders, there are a few problem constraints imposed. For one, each time we buy or sell any asset, we must pay a trading fee F . Additionally, the transition must be made without borrowing cash (all C values must be positive) or short-selling (portfolios must be non-negative at each time step).

The portfolio turnover problem increases in difficulty when we extend the trading window from a single period to instead consider $T \gg 1$ periods. The trader now can examine the varying asset prices across time Y_t and identify opportune times to execute trades to ensure minimal portfolio value is lost while still attaining the target state. However, the true prices are unknown for any future period and instead, the trader must instead rely on forecasted prices \hat{Y}_t , which are often very error-prone. This means that the increased opportunities for portfolio growth in a multi-period transition come with equivalent risk. As such, it is useful to have a framework that allows a trader to specify the level of risk they are comfortable with and provides a trading strategy that accommodates their preference.

4 Transition Framework

Throughout this paper, we will make reference to a multi-period model called a ‘‘Receding Horizon’’ (RH) model - popularized in finance by Boyd [2017]. The model is explained in detail in the cited work and thus, will only be briefly covered in this section. The premise of the model is to leverage a rolling horizon approach over a finite horizon. In other words, the framework finds a desired plan of actions for a given number of future periods - Lookahead - and continuously recalculates this plan based on new observed data. The key insight is that the plan is calculated using data-driven forecasting methods (Section 5.1). The algorithm is presented below (Algorithm 1).

Algorithm 1 Multi-Period Transition Framework

Require: $P_0, \mathcal{T}, Y_{-L:0}, \mathbf{X}_{-L:0}, V_0, T, C_0, \text{OPT}, \text{Forecaster}, \text{MarketObserver}$

Ensure: $Y_0^T \mathcal{T} + C_0 \leq V_0$

$t \leftarrow 0$

$P_t \leftarrow P_0$

$C_t \leftarrow C_0$

$Y \leftarrow Y_{-L:t}$

while $t \leq T$ **do**

$V_t \leftarrow Y_t^T P_t + C_t$

▷ Calculate portfolio value at current timestep

$\hat{Y}_{t+1:T} \leftarrow \text{Forecaster}(Y_{-L:t}, \mathbf{X}_{-L:t})$

▷ Predict unknown time steps with known information

$Z_{t:T}, W_{t:T} \leftarrow \text{OPT}([Y_t \hat{Y}_{t+1:T}], P_t, C_t, \mathcal{T})$

▷ Run Optimization Policy

$P_{t+1} \leftarrow P_t + Z_t$

▷ Execute trades at current timestep

$C_{t+1} \leftarrow C_t - Y_t \cdot Z_t - F \vec{1} \cdot W_t$

▷ Calculate cash

$Y_{t+1} \leftarrow \text{MarketObserver}()$

▷ Observe the market for new price information

$t \leftarrow t + 1$

end while

Ensure: $P_T \geq \mathcal{T}$

At each timestep, we utilize known price information and external covariates (\mathbf{X}) to predict future price information. This information is passed into an integer programming based optimization policy which calculates a potential set of trades for the remaining periods. From the calculated buys and sells, we execute the trades prescribed for the current period (Z_t) and pay the required fees for these transactions (W_t). We then observe the changes that occur in the market, updating the value of our portfolio (V_t) based on new price information and repeating this process till the end of our trading horizon.

Rolling trades out iteratively can help an investor anticipate price movements that may impact one’s ability to liquidate or acquire certain position [Boyd, 2017]. Additionally, including the new price information can help correct errors in our forecast that result from incorrectly predicting the trend or unexpected market events. These factors combined help minimizes losses when performing a portfolio conversion.

4.1 Multi-Period Trading Policies

While multiple policies can be plugged into our portfolio transition framework, we propose various formulations that work well in this environment, all of which are variations of the following formulation:

$$\text{OPT}(Y, P_t, C_t, \mathcal{T}) = \underset{z, w}{\text{maximize}} \quad C_{\mathbf{T}} + Y_{\mathbf{T}}^T P_{\mathbf{T}} - \sum_{\tau=t}^{\mathbf{T}} \sum_{a \in A} F(w_{\tau,a}^+ + w_{\tau,a}^-) \quad (1)$$

$$\text{subject to } P_{\tau} = P_{\tau-1} + z_{\tau}^+ - z_{\tau}^- \quad \forall \tau = t+1, \dots, \mathbf{T} \quad (2)$$

$$C_{\tau} = C_{\tau-1} - Y_{\tau}^T (z_{\tau}^+ - z_{\tau}^-) - \sum_{a \in A} F(w_{\tau,a}^+ + w_{\tau,a}^-) \quad \forall \tau = t+1, \dots, \mathbf{T} \quad (3)$$

$$z_{\tau}^+ \leq M_{\tau} w_{\tau}^+ \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (4)$$

$$z_{\tau}^- \leq M_{\tau} w_{\tau}^- \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (5)$$

$$P_{\tau} \geq \vec{0} \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (6)$$

$$C_{\tau} \geq 0 \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (7)$$

$$P_{\mathbf{T}} \geq \mathcal{T} \quad (8)$$

$$w_{\tau}^+ + w_{\tau}^- \leq 1 \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (9)$$

$$w_{\tau}^+, w_{\tau}^- \in \{0, 1\}^{|A|} \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (10)$$

$$z_{\tau}^+, z_{\tau}^- \in \mathbb{Z}_+^{|A|} \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (11)$$

In this formulation, z_{τ}^+ and z_{τ}^- are vectors that represent the magnitude of buys and sells suggested at time step τ and $Z_{\tau} = z_{\tau}^+ - z_{\tau}^-$. To model trading fees, we include two binary logical vectors w_{τ}^+ and w_{τ}^- that represent whether a buy or sell is executed and $W_{\tau} = w_{\tau}^+ + w_{\tau}^-$. The magnitude and logical vectors are linked to each other via constraints 4 and 5. For both of these constraints, M_{τ} is an upper bound on the number of shares that can be bought or sold during each time step τ . More formally,

$$U_t = V_t \quad (12)$$

$$U_{\tau} = V_t \prod_{i=t+1}^{\tau} \max\{1 + \frac{Y_{i,a} - Y_{i-1,a}}{Y_{i-1,a}} \mid a \in A\} \quad \forall \tau = t+1, \dots, \mathbf{T} \quad (13)$$

$$M_{\tau} = \lceil \frac{U_{\tau}}{\min Y_{\tau}} \rceil \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (14)$$

Where U_t represents the upper bound of the portfolio value at time t and V_t the portfolio value at time t . The intuition behind this bound is based on the perfect trade assumption. In order to maximize portfolio value at each time step, a trader would move the entire value of their portfolio into the asset with the largest positive price gain between the current period and the next. The upper bound of the accessible value at each time step is then equivalent to the initial portfolio value times the product of the maximum possible gains at each time step. Using this value, we can then divide this by the smallest priced asset at each τ to represent the maximum number of assets that could be transacted in that period, which is equivalent to M_{τ} .

Constraints 2 and 3 are linear expressions that model the held assets and cash at each time step in our horizon. As previously mentioned, the goal of our algorithm is to transition to a portfolio that satisfies a target portfolio \mathcal{T} . This target portfolio represents the minimum number of each stock we must have in our final portfolio, and this requirement is captured in constraint 8. Additionally, we represent the requirements prohibiting short selling and leverage in constraints 6 and 7 respectively.

Constraint 9 is a constraint introduced to reduce the number of effectively symmetric points and in turn reduce the solver runtime. Without this constraint, solutions, where we buy and sell the same asset at the same timestep, are permitted, resulting in a large feasible space with multiple solutions that result in the same net trade. However, this action will never be optimal due to the presence of trading costs, and constraint 9 trims these possibilities by constraining the model to take a single action with a given asset at each timestep, either buy or sell but not both.

For our objective, we aim to maximize our end portfolio value while also minimizing the amount wasted on trading fees (1). One may notice that the impact of trading fees is included both explicitly in the objective function (1) and through the cash relations between times (3). This double inclusion is intentional as we want our optimization to emphasize reducing wasted money.

This base formulation can also be thought of as a "day trading" formulation where the recommendations will be extremely aggressive.

4.1.1 Directional Trading Formulation

In our first formulation (also known as the directional formulation), only trades that move the portfolio towards the target are permitted. This means that any trades that attempt to generate profit are prohibited and the main purpose of this policy is to simply minimize the cost of the transition across our time period.

To implement this, we can split A into 2 sets at each time step the optimization is run on based on whether we are allowed to buy that stock ($A_t^+ := \{a \in A; P_{t,a} < \mathcal{T}_a\}$) or sell that stock ($A_t^- := \{a \in A/A_t^+\}$). We can then add the following constraints on w^+ and w^- to our base formulation:

$$w_{\tau,a}^+ \leq 0 \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad \forall a \in A_t^- \quad (15)$$

$$w_{\tau,a}^- \leq 0 \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad \forall a \in A_t^+ \quad (16)$$

One note is that A must be re-partitioned in each iteration of the multi-horizon transition framework. As all trades are executed with imperfect future information, the optimal set of trades may include purchasing more stock than is necessary to meet the target as the forecast for that asset may indicate a strong future performance. However, if the forecast was incorrect and the value of the stock instead falls drastically over multiple periods, a locked A^+ and A^- set will force the trader to accept these losses, reducing the benefits of rolling trades out gradually. By redistributing the assets between the buy and sell sets as new information arrives, we allow for course corrections and loss prevention, making our approach more robust to market shocks.

The addition of constraints 15 and 16 also remove the need for constraint 9 as now each asset can only be bought or sold in all time steps.

4.1.2 Penalized Directional Trading Formulation

While the directional trading policy allows for a safe transition to the target, the policy does not subscribe trades that do not directly move towards the target portfolio. This can result in lost profit opportunities as some of these superfluous trades could take advantage of price fluctuations to generate extra cash which may be used to move our portfolio closer to satisfying the target while reducing the number of assets that need to be sold.

However, we also want to retain some of the benefits of allowing trades in only a single direction, namely the reduction in volatility that comes from limiting the number of trades. To balance these issues, we introduce our Directional Penalty formulation. Instead of hard constraining trades that move us away from our portfolio, we instead implement a soft constraint that instead penalize our objective by the value of the trades that move in the wrong direction by adding the following term to the objective function of our base formulation:

$$-\lambda \sum_{\tau=t}^{\mathbf{T}} \left[\sum_{a \in A_t^+} Y_{\tau,a} z_{\tau,a}^- + \sum_{a \in A_t^-} Y_{\tau,a} z_{\tau,a}^+ \right] \quad (17)$$

Instead of prohibiting trades that move us away from the target, we instead penalize our objective by a portion of the value of those trades, encouraging the algorithm to only recommend trades if the profit potential is sufficiently high. As we show in our experimental results, this formulation has the benefit of allowing users to specify the risk they are comfortable with by increasing λ for less risk and decreasing it for more risk. When $\lambda = 0$, this formulation is identical to the base formulation and is an extremely aggressive day trading formulation.

4.2 Naive Trading

As a comparison baseline of the multi-period trading policies, we introduce a naive policy where trading is only permitted on the first day, therefore the target portfolio must be satisfied on this day. The formulation for this policy is as follows:

$$\underset{z, w}{\text{maximize}} Y_0^T P + C - \sum_{a \in A} F(w_a^+ + w_a^-) \quad (18)$$

$$\text{subject to } P = P_0 + z^+ - z^- \quad (19)$$

$$C = C_0 + Y_0^T (z^- - z^+) - \sum_{a \in A} F(w_a^+ + w_a^-) \quad (20)$$

$$z^+ \leq Mw^+ \quad (21)$$

$$z^- \leq Mw^- \quad (22)$$

$$P \geq \mathcal{T} \quad (23)$$

$$P \geq 0 \quad (24)$$

$$C \geq 0 \quad (25)$$

$$z^+, z^- \in \mathbb{Z}^{|A|} \quad (26)$$

$$w^+, w^- \in \{0, 1\}^{|A|} \quad (27)$$

For this formulation, $M = \lceil (Y_0^T P_0 + C_0) / \min Y_0 \rceil$ or the value of the portfolio divided by the price of the least valuable asset. The purpose of this formulation is to solely to transition to a satisfactory portfolio in the smallest number of trades possible and by solving this formulation, we are able to receive an estimate of the number of trades required to transition from the initial portfolio to one that satisfies the target for any given portfolio.

Additionally, we can track the resultant portfolio generated by this optimization and evaluate how it performs during and after our trading window to evaluate the utility of the multi-period policies.

5 Methodology

To evaluate our policies, we generated 254 different scenarios using using historical stock price data from the Toronto Stock Exchange (TSX) from 1999 to 2022. In each scenario, we chose 20 to 50 different stocks and randomly created initial and target portfolios by first deciding a budget between \$15,000 and \$350,000 and then randomly picking stocks from our chosen set until the budget is exhausted. Each scenario was also assigned a fixed trading fee between \$2 and \$9.

We then ran our transition framework on each instance to simulate how our approach would perform given real market conditions. While the trading fees and portfolio values were varied, the length of the trading horizon (T) was kept constant for every scenario and set to 30 business days. At the end of each simulation, we measured the runtime required to solve our optimization formulations, the percent change in the portfolio value between the first and last periods of the simulation, the number of trades executed, and the total value lost to trading fees.

5.1 Forecasting

While any forecasting method can be used to predict future stock prices, we chose to use N-HiTS [Challu et al., 2022]. N-HiTS is a deep-learning architecture that leverages residual connections, multi-rate signal sampling, and interpolation mechanisms to analyze patterns at different time scales and iteratively construct the end forecast [Challu et al., 2022]. The architecture of N-HiTS is organized into blocks and stacks, with stacks containing multiple blocks and each block consisting of a few fully connected layers that predict weights on a basis vector. For our experiments, we utilized the implementation provided by Darts [Herzen et al., 2022], a user-friendly Python library for time-series forecasting.

Each scenario has its own instance of the model which was trained for 125 epochs using 5 years of historical data from the start of the training period with a Mean Absolute Percent Error loss function. The model took in 48 days of historical price information and interest rate history to predict price of the stocks over the next 30 days. Each instance had 3 total stacks, 2 blocks per stacks, and 4 fully connected linear layers of 128 width per block. The rest of the hyperparameters such as the learning rate, activation function, and dropout were kept to the defaults provided by Darts. Since each scenario had a different training set, the performance of the forecasts also varied from scenario to scenario. To mitigate this, the average MAPE of the generated forecasts were recorded and scenarios with a MAPE higher than 10% were excluded from experimentation as often these corresponded to unusual events in the market such as earnings releases, governmental policies, and worldwide pandemics. While this exclusion is not valid in practice, the focus of this work is on the integer programming framework rather than developing a strong forecasting algorithm since there is

a large amount of work in this field already [Gandhmal and Kumar, 2019, Sharma et al., 2017]. On average the mean average percent error of our forecasts is approximately 8%.

5.2 Environment

All experiments were run on a device with a 12th Gen Intel(R) Core i7-12700 @ 2.10 GHz and 32 GB of memory. All the forecasting models were trained using a NVIDIA GeForce RTX 3080. Our optimization models were implemented using Gurobi’s Python API [Gurobi Optimization, LLC, 2023].

6 Experimental Results

6.1 Percent Change Analysis

Looking at the statistics on the the percent change in portfolio value (Figure 1), we can see a clear difference in the performance of our various optimization policies.

policy	Mean	Std. Dev.	Median	Max	Min
Naive	1.14	3.44	0.88	11.27	-10.35
Directional	1.25	3.22	1.04	11.84	-8.29
DirP_500	1.31	3.43	0.95	14.51	-10.83
DirP_75.0	1.39	3.91	0.86	30.08	-11.15
DirP_50.0	1.47	4.26	0.97	23.62	-12.07
DirP_25.0	2.01	7.55	1.46	58.36	-34.55
DirP_0	3.18	12.59	1.62	58.59	-36.50

Figure 1: Summary Statistics on the Percent Change (%) in Portfolio Value

On average, every one of our policies had higher portfolio gains than the naive baseline, clearly showing the value of trading over multiple periods. Additionally, we can see that our non-restricted base formulation (DirP_0) on average almost tripled the expected returns of the Naive formulation. However, this increase in average return can be explained by the extremely high standard deviation in the observed changes. We can observe from looking at the other Directional Penalty Formulations (DirP_500, DirP_75, DirP_50, and DirP_25) that our penalty factor λ (equal to 500%, 75%, 50%, and 25% respectively) is inversely correlated to the deviation in returns with higher penalties resulting in a lower return but also lower deviation in returns. In order words, our formulation allows investors to represent their acceptable risk in the problem via the penalty parameter λ allowing them to balance the trade off between profit-generation and safe transitions. Additionally, if an investor is extremely risk averse, they can instead opt to use the alternative, pure directional formulation. This formulation not only has a higher average return than the naive formulation, but also a lower deviation therefore lower risk.

These insights are made more clear when performing a win-loss-tie analysis of the different policies against each other.

	Directional	DirP_0	DirP_25.0	DirP_50.0	DirP_75.0	DirP_500	Naive
Directional	—	117/137/0	104/150/0	114/136/4	113/137/4	116/135/3	144/110/0
DirP_0	137/117/0	—	129/122/3	139/114/1	138/116/0	138/116/0	137/117/0
DirP_25.0	150/104/0	122/129/3	—	146/99/9	140/107/7	144/101/9	151/102/1
DirP_50.0	136/114/4	114/139/1	99/146/9	—	81/81/92	95/87/72	142/112/0
DirP_75.0	137/113/4	116/138/0	107/140/7	81/81/92	—	92/89/73	144/109/1
DirP_500	135/116/3	116/138/0	101/144/9	87/95/72	89/92/73	—	144/110/0
Naive	110/144/0	117/137/0	102/151/1	112/142/0	109/144/1	110/144/0	—

Figure 2: Win Loss Tie Comparison of Different policies (Row Win/Row Loss/Tie)

This analysis shows us that all of our policies have higher returns than the naive formulation most of the time, but the ratio of wins to losses between remains relatively similar across all the formulations, showing that the main difference between the various formulations is not whether they win over the naive formulation, but rather the magnitude of their win.

Another trend highlighted by this analysis as well as the summary statistics in figure 1 is the diminishing returns from increasing λ . The number of ties between $\lambda = 25\%$ and $\lambda = 50\%$ is 9, but the number of ties between $\lambda = 75\%$ and $\lambda = 500\%$ is 73, despite the later having over a 400% difference in value. This is supported by the difference in average gain where the increasing lambda by 25% with low values results in an average decrease of .54% but increasing it by 25% from 50% or by 425% from 75% both only result in an .08% decrease in value. Additionally, as λ increases, the behavior converges closer to the directional formulation and in practice, if users specify their safety factor to be incredibly high, any tool leveraging our method should convert that to instead utilize the directional formulation.

One large factor impacting the observed returns of our policies is the market conditions at the time of the simulation. As mentioned before, the performance of the naive baseline provides information on the how are target portfolio performs at the end of the trading horizon, helping us understand whether or not the optimization was done in a bear or bull market with respect to A .

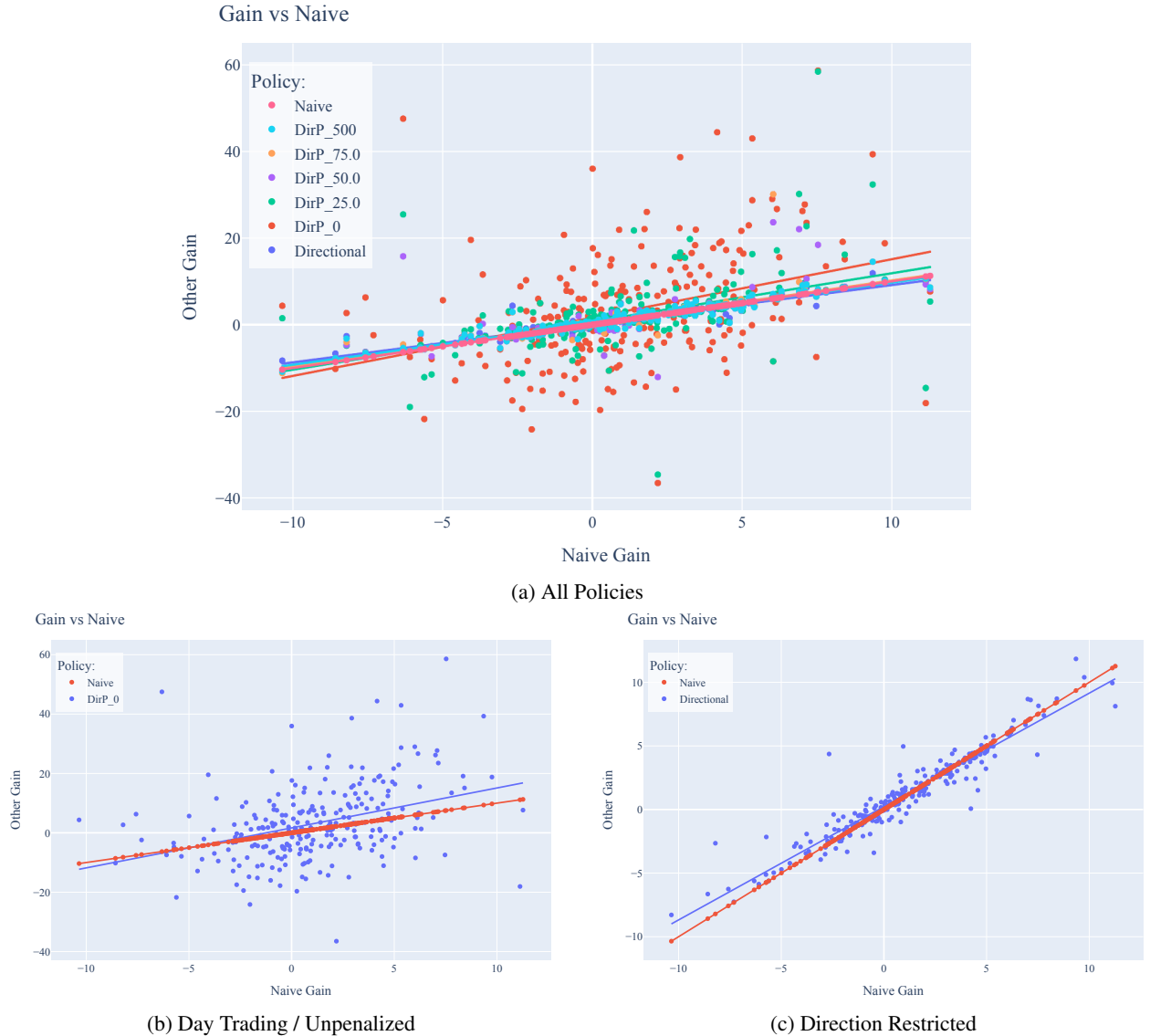


Figure 3: Comparison of Returns to the Naive Baseline

Looking at the returns of our formulations with respect to market conditions (Figure 3), we can see that our different formulations have clear advantages depending on whether or not we are in a bear market. In this figure, the pink line represents the naive performance, or the baseline market condition.

When examining the performance our unpenalized formulation (Figure 3b), we can see that that the trendline fit to the data is steeper than that of the expected market conditions. This behavior shows that this formulation simply magnifies the direction of the market, performing incredibly well in a bull market and generating high expected returns, but conversely does poorly in a bear mark.

This behavior is a direct contrast to the behavior of the pure directional formulation (figure 3c) Where the trend line has a more gradual change with a smaller slope than the baseline. This shows that while the directional trading policy does not capitalize on opportunities to generate cash in a bull market and may perform worse than simply holding the portfolio, it is extremely good at minimizing losses in bad market conditions, a trait that is extremely helpful in recessionary conditions.

6.2 Trading Cost Analysis

policy	Number of Trades Executed					Trading Cost Incurred				
	Mean	Std. Dev.	Median	Max	Min	Mean	Std. Dev.	Median	Max	Min
DirP_0	54.76	22.26	52.0	113.0	19.0	306.85	180.54	277.5	1017.0	42.0
DirP_25.0	46.83	18.16	44.5	104.0	12.0	263.79	151.25	232.0	832.0	48.0
DirP_50.0	39.10	13.17	36.0	106.0	12.0	219.65	116.61	204.0	630.0	42.0
DirP_75.0	38.81	11.79	36.5	78.0	12.0	217.62	110.39	207.0	621.0	44.0
DirP_500	38.74	11.63	36.5	69.0	12.0	217.76	111.59	208.5	621.0	46.0
Directional	20.22	7.06	20.0	42.0	6.0	112.76	60.10	102.0	288.0	20.0
Naive	27.16	8.40	26.0	45.0	0.0	153.76	81.13	144.0	387.0	0.0

Figure 4: Summary Statistics on Trading Behavior

When examining the number of executed trades and the corresponding incurred trading cost, we can see that the directional formulation significantly reduced the number of trades needed to achieve our target portfolio compared to our naive baseline and as a result the average amount of money wasted on trading fees is also significantly lower. In contrast, the directional penalty formulations as well as the unpenalized day trading variant all had much higher trading volumes with the day trading and $\lambda = 25\%$ variants trading more than twice the amount of the directional formulation.

One can also see that as the penalty parameter is increased, the number of trades reduces as well, reinforcing λ as a value to control accepted risk and aggressiveness. Additionally, the significant reduction in trading costs once again highlights the directional formulation’s benefits of being a method to easily navigate a portfolio to a target while consistently minimizing lost value.

6.3 Runtime Analysis

As this is an approach designed for retail traders, in practice this would probably be delivered as a web-based application and as such, algorithmic runtime is important for both a seamless user experience and minimizing the cost of mounting the application on cloud platform that often charge on a usage basis.

policy	Mean	Std. Dev.	Median	Max	Min
Naive	10.43	4.37	10.31	24.11	3.22
Directional	22.26	13.05	20.17	166.39	7.77
DirP_500	33.87	16.17	30.70	104.45	11.67
DirP_75.0	40.75	27.14	33.91	293.39	11.63
DirP_50.0	42.28	29.70	35.23	295.14	11.51
DirP_25.0	38.44	24.47	32.93	222.33	11.48
DirP_0	33.80	14.56	31.63	96.73	12.67

Figure 5: Summary Statistics on Total Algorithm Runtime (s)

Figure 5 shows the total runtime taken by the optimization policies over the simulation. While in practice this optimization would be done over 30 days, these results are helpful for showcasing the performance of our policies.

While the total runtime across the simulation is significantly higher than the naive formulation, in practice these differences are negligible and on average the optimization would take 1-2 seconds per day with the directional policy taking even less time.

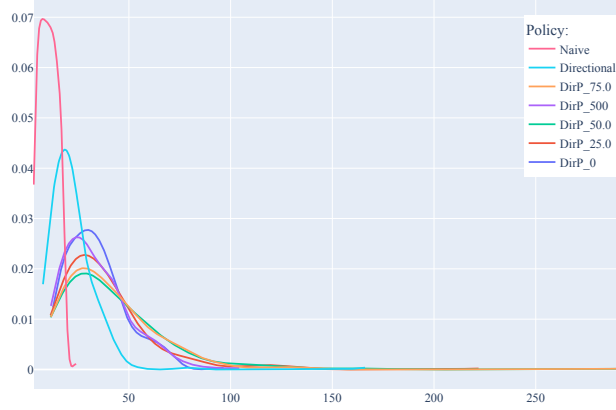


Figure 6: Probability Distribution of the runtimes

As seen in figure 6, the distribution of the runtimes has an extremely long tail and the difference between the median and the max runtimes for all our policies is extremely high. Additionally, while the median and mean runtimes for the penalty policies are similar, the length of the tails are drastically different.

An interesting observation is that the constrained directional policy ran in almost half the time as the penalized policies, despite the approach of the penalized policy being similar to taking a Lagrangian relaxation of the constraints 15 and 16. One potential reason for this runtime difference is that these two constraints trim a large portion of the feasible space in turn reducing the maximum possible size of the branch-and-bound tree. Additionally, the average runtime does not seem to correlate with penalty values as having a $\lambda = 0$ has an extremely similar runtime distribution to an extremely high penalty.

7 Column Generation

7.1 Introduction & Methodology

Column Generation is a method for solving complex IPs through feasible solution enumeration. It leverages the structure of a Dantzig reformulation to enumerate feasible solutions based on patterns in the problem constraints. In our problem, such patterns are present in the sell and buy behavior. We hope to lean on the enumeration idea of column generation to leverage the structure embedded in the trading pattern.

As a reminder, the problem we are solving involves a portfolio turnover from a current portfolio to a target portfolio in a defined number of days. The algorithm must determine the timing of the trades that would make a liquidation or purchase of positions most optimal, namely, by preserving portfolio value against transaction costs and market fluctuations. Since we have the target portfolio’s construction, the quantity bought or sold is assumed to be the quantity that satisfies the difference between the former and the initial portfolio - no more, no less. In other words, knowing how much is sold is partly irrelevant to the trader since it’s assumed that it’s the necessary quantity. In addition, the trader would not be asked to buy or sell the same asset more than once as that would defeat the goal of finding the “best” time to sell/buy. As such, we can apply a Dantzig reformulation, if we consider the enumerated feasible solutions to be the time period at which an action is taken for each asset in the portfolio (Formulation 42). We expose the Dantzig reformulation below.

First, we have:

$$\underset{z, w}{\text{maximize}} \quad C_{\mathbf{T}} + Y_{\mathbf{T}}^T P_{\mathbf{T}} - \sum_{\tau=t}^{\mathbf{T}} \sum_{a \in A} F(w_{\tau,a}^+ + w_{\tau,a}^-) \quad (28)$$

$$\text{subject to} \quad P_{\tau} = P_{\tau-1} + z_{\tau}^+ - z_{\tau}^- \quad \forall \tau = t+1, \dots, \mathbf{T} \quad (29)$$

$$C_{\tau} = C_{\tau-1} - Y_{\tau}^T (z_{\tau}^+ - z_{\tau}^-) - \sum_{a \in A} F(w_{\tau,a}^+ + w_{\tau,a}^-) \quad \forall \tau = t+1, \dots, \mathbf{T} \quad (30)$$

$$z_{\tau}^+ \leq M_{\tau} w_{l,\tau}^+ \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (31)$$

$$z_{\tau}^- \leq M_{\tau} w_{k,\tau}^- \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (32)$$

$$P_{\tau} \geq \vec{0} \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (33)$$

$$C_{\tau} \geq 0 \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (34)$$

$$P_{\mathbf{T}} \geq \mathcal{T} \quad (35)$$

$$z_{\tau}^+, z_{\tau}^- \in \mathbb{Z}_+^{|A|} \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (36)$$

$$w_{l,\tau}^+ \in W_{\tau}^+ \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (37)$$

$$w_{k,\tau}^- \in W_{\tau}^- \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (38)$$

$$(39)$$

With,

$$W^+ = \{w_l^+ \in \{0, 1\}^{[T \times N]} : w_l^+ + w_l^- \leq 1, \sum_{\tau} w_{l,\tau}^+ \leq 1\} \quad (40)$$

$$W^- = \{w_k^- \in \{0, 1\}^{[T \times N]} : w_k^+ + w_k^- \leq 1, \sum_{\tau} w_{k,\tau}^- \leq 1\} \quad (41)$$

Which gives us the following Dantzig formulation:

$$\underset{z, \lambda, \gamma}{\text{maximize}} \quad C_{\mathbf{T}} + Y_{\mathbf{T}}^T P_{\mathbf{T}} - \sum_{\tau=t}^{\mathbf{T}} \sum_{a \in A} F(w_{\tau,a}^+ + w_{\tau,a}^-) \quad (42)$$

$$\text{subject to} \quad P_{\tau} = P_{\tau-1} + z_{\tau}^+ - z_{\tau}^- \quad \forall \tau = t+1, \dots, \mathbf{T} \quad (43)$$

$$C_{\tau} = C_{\tau-1} - Y_{\tau}^T (z_{\tau}^+ - z_{\tau}^-) - \sum_{a \in A} F(w_{\tau,a}^+ + w_{\tau,a}^-) \quad \forall \tau = t+1, \dots, \mathbf{T} \quad (44)$$

$$z_{\tau}^+ \leq M_{\tau} w_{l,\tau}^+ \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (45)$$

$$z_{\tau}^- \leq M_{\tau} w_{k,\tau}^- \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (46)$$

$$P_{\tau} \geq \vec{0} \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (47)$$

$$C_{\tau} \geq 0 \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (48)$$

$$P_{\mathbf{T}} \geq \mathcal{T} \quad (49)$$

$$w_{\tau}^+ = \lambda_l \cdot w_{l,\tau}^+ \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (50)$$

$$w_{\tau}^- = \gamma_k \cdot w_{k,\tau}^- \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (51)$$

$$\sum_{l=1}^L \lambda_l = 1 \quad (52)$$

$$\sum_{k=1}^K \gamma_k = 1 \quad (53)$$

$$z_{\tau}^+, z_{\tau}^- \in \mathbb{Z}_+^{|A|} \quad \forall \tau = t, t+1, \dots, \mathbf{T} \quad (54)$$

$$\lambda_l \in \{0, 1\} \quad \forall l = 1, \dots, L \quad (55)$$

$$\gamma_k \in \{0, 1\} \quad \forall k = 1, \dots, K \quad (56)$$

In plain terms, the sets W^+ , W^- , are both respective enumerations of all the buy actions and sell actions (when to buy and when to sell). For an optimization with 3 assets, 30 lookahead days and 2 assets needing to be liquidated, the W^- set will have (including: “no sell action as a possible action”) $30 * 30 + 2 * 30 + 1 = 961$ different possible combinations of sell actions. This value grows exponentially with the number of assets and lookahead periods. The key insight is that the number of possible options dramatically decreases once an asset reaches its target state. In fact, if this occurs before the trading horizon is complete, no more feasible solution will be generated for this asset, i.e., $\gamma = \vec{0}$.

7.2 Results

We implemented two versions of our column generation (CG) algorithm. The first one (ColGen_False) only enumerated the possible buy strategies and maintained the sell actions as defined in formulation 1. The second version (ColGen_True) includes the enumeration of both buy and sell strategies. Both were compared for the sake of strategic and numerical performance.

policy	Mean	Std. Dev.	Median	Max	Min
Naive	3.16	6.37	5.46	8.98	-7.19
Directional	3.08	6.31	5.46	8.99	-7.19
DirP_25.0	3.08	6.31	5.46	8.99	-7.19
DirP_0	8.43	12.56	8.73	19.27	-12.11
ColGen_True	3.11	6.34	5.46	8.99	-7.20
ColGen_False	3.11	6.33	5.46	8.98	-7.19

Table 1: Summary Statistics on the Percent Change (%) in Portfolio Value

policy	Amount Traded					Trading Cost Incurred				
	Mean	Std. Dev.	Median	Max	Min	Mean	Std. Dev.	Median	Max	Min
ColGen_False	2.0	1.41	3.0	3.0	0.0	7.0	4.64	9.0	12.0	0.0
ColGen_True	2.2	1.30	3.0	3.0	0.0	8.0	4.64	9.0	12.0	0.0
DirP_0	14.0	7.68	18.0	21.0	3.0	55.8	34.77	57.0	108.0	15.0
DirP_25.0	2.6	1.14	3.0	4.0	1.0	10.2	4.09	9.0	16.0	5.0
Directional	1.8	1.30	2.0	3.0	0.0	6.4	4.51	6.0	12.0	0.0
Naive	2.4	0.89	3.0	3.0	1.0	9.4	2.88	9.0	12.0	5.0

Table 2: Summary Statistics on Trading Behavior

policy	Mean	Std. Dev.	Median	Max	Min
Naive	3.39	0.12	3.38	3.58	3.27
Directional	9.55	0.17	9.60	9.71	9.32
DirP_25.0	10.00	0.17	9.96	10.28	9.85
DirP_0	11.26	0.63	11.24	12.02	10.64
ColGen_True	9.32	2.79	9.35	13.79	6.57
ColGen_False	9.55	0.17	9.53	9.75	9.31

Table 3: Summary Statistics on Total Algorithm Runtime (s)

The CG strategies performed almost identically on all metrics (Table 1). The algorithm achieves a better mean return than the strict directional algorithm with minimal additional variance. Further, similar performance is observed when we consider the number of trades and the amount of trading costs incurred (Table 2). On all accounts, the CG methods, place themselves between the performance of the Naive method and the strict directional method.

We note that the solve times of the CG enumeration methods are on average less than their directional counterpart, but, they suffer from large variances. This is the direct result of large number of enumerations at the beginning of the trading horizon (Table 3).

7.3 Discussion

In general, the performance of the enumeration method is not particularly advantageous though it does beat the directional approach. Where this method becomes beneficial is when trades are completed before the end of the trading horizon. Then, the optimization is systematically constrained to stop considering possible trades for a given asset.

We realize that our algorithm could benefit from a complete implementation of the CG algorithm. Instead of generating and optimizing over all possible trade patterns we could generate subsets of feasible solutions to create interesting bounds to be utilized in a branch-and-bound algorithm. Unfortunately, with limited time, we will have to leave this out of the scope of this project. In the same vein, we believe there is space to explore intelligent generation of subsets that would allow the BnB algorithm to quickly identify a tight bound to the problem which could then serve to easily solve the MIP problem. For example, one could divide subsets based on chunks of days. More work needs to be undertaken to develop such an approach.

8 Conclusion

In this work, we present a multi-period receding horizon framework along with two different integer programming based optimization policies that allow retail traders to efficiently transition the holdings in their current portfolio to one that satisfies their investment targets. The first policy, also known as the directional trading policy, limits trades to only ones that make progress in the transition and leverages price volatility to identify trading moments that minimize the trades needed. The other policy, also known as the directional penalty policy, allows the user to specify their acceptable risk and suggests trades that maximize the end portfolio value. Experimental results show that the directional trading policy significantly reduces the amount of money wasted on trading fees and reduces losses to portfolio value in bear markets when compared to a single-period transition. The directional penalty policy also shows significant potential and allows the user to achieve significant returns in bull markets when users are willing to accept higher risk and moderate gains in portfolio value when traders want lower risks. Additionally the computational load of all our policies is relatively low, allowing our methodologies to be easily implemented in practice, potentially on low-cost cloud computing platforms. These promising results show the potential of integer programming as a tool to help retail investors better manage their portfolios. One shortcoming of this work is the assumptions that the forecast will have reasonable error.

Black swan events can occur and can result in extremely inaccurate forecasts, a case we do not explicitly handle. While a receding horizon framework slightly mitigates these risks, including some explicit risk-handling measures could greatly improve our framework. Additionally, our framework would obviously benefit from high quality forecasts therefore anyone aiming to implement the methodologies described in this paper for their own use should develop a strong forecasting algorithm they can trust. On a similar note, future directions of work for this problem include modifying the prediction models used to generate price estimates to one that generates probabilistic forecasts and leverage multi-stage stochastic programs as the optimization policy. While this would increase the computational load, approaching the portfolio transition problem through this lens can allow for safer, more profitable decision making.

Another potential avenue for exploration is splitting up the problem into two-stages, one to decide when to trade and another to decide how much to trade on the days decided. This reformulation combined with the multi-stage receding horizon framework could allow for significant runtime improvements by not solving the second-stage problem if the number of trades decided for a given day is zero, as only the trade recommendations for the current day is accepted in our framework.

The nature of the portfolio transition problem also lends itself nicely to a Smart "Predict, then Optimize" approach such as the one proposed by Elmachtoub and Grigas [2022]. Instead of having our forecasting and optimization as two separate components in our work flow, recent work has made it possible to integrate optimization models directly into the prediction workflow and have a neural network directly predict the optimization output [Tang and Khalil, 2022].

Along with new techniques, further improvements could be made to our work by including other components of trading such as dynamic trade pricing, dividends, bid-ask spreads, and trade latency.

References

Harry Markowitz. Portfolio Selection. *The Journal of Finance*, 7(1):77–91, 1952. ISSN 0022-1082. doi:10.2307/2975974. URL <https://www.jstor.org/stable/2975974>. Publisher: [American Finance Association, Wiley].

- William F. Sharpe. CAPITAL ASSET PRICES: A THEORY OF MARKET EQUILIBRIUM UNDER CONDITIONS OF RISK*. *The Journal of Finance*, 19(3):425–442, September 1964. ISSN 0022-1082. doi:10.1111/j.1540-6261.1964.tb02865.x. URL <http://dx.doi.org/10.1111/j.1540-6261.1964.tb02865.x>. Publisher: Wiley.
- R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3): 21–41, 2000. ISSN 14651211. doi:10.21314/JOR.2000.038. URL <http://www.risk.net/journal-of-risk/technical-paper/2161159/optimization-conditional-value-risk>.
- Renata Mansini, Włodzimierz Ogryczak, and M. Grazia Speranza. *Linear and Mixed Integer Programming for Portfolio Optimization*. EURO Advanced Tutorials on Operational Research. Springer International Publishing, Cham, 2015. ISBN 978-3-319-18481-4 978-3-319-18482-1. doi:10.1007/978-3-319-18482-1. URL <https://link.springer.com/10.1007/978-3-319-18482-1>.
- Miguel Sousa Lobo, Maryam Fazel, and Stephen Boyd. Portfolio optimization with linear and fixed transaction costs. *Annals of Operations Research*, 152(1):341–365, March 2007. ISSN 0254-5330, 1572-9338. doi:10.1007/s10479-006-0145-1. URL <http://link.springer.com/10.1007/s10479-006-0145-1>.
- Stephen Boyd. Multi-Period Trading via Convex Optimization. *Foundations and Trends® in Optimization*, 3(1): 1–76, 2017. ISSN 2167-3888, 2167-3918. doi:10.1561/24000000023. URL <http://www.nowpublishers.com/article/Details/OPT-023>.
- Joseph Wickremasinghe and Abhishek Gupta. Breaking Down the Direct Indexing Investment Process, July 2022. URL <https://www.msci.com/www/research-report/breaking-down-the-direct/03310986438>.
- Keith V. Smith. A Transition Model for Portfolio Revision. *The Journal of Finance*, 22(3):425–439, 1967. ISSN 0022-1082. doi:10.2307/2978895. URL <https://www.jstor.org/stable/2978895>. Publisher: [American Finance Association, Wiley].
- Pierre Collin-Dufresne, Kent Daniel, and Mehmet Sağlam. Liquidity regimes and optimal dynamic asset allocation. *Journal of Financial Economics*, 136(2):379–406, May 2020. ISSN 0304-405X. doi:10.1016/j.jfineco.2019.09.011. URL <http://dx.doi.org/10.1016/j.jfineco.2019.09.011>. Publisher: Elsevier BV.
- Nicolae Gârleanu and Lasse Heje Pedersen. Dynamic Trading with Predictable Returns and Transaction Costs: Dynamic Trading with Predictable Returns and Transaction Costs. *The Journal of Finance*, 68(6):2309–2340, December 2013. ISSN 00221082. doi:10.1111/jofi.12080. URL <https://onlinelibrary.wiley.com/doi/10.1111/jofi.12080>.
- Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler, and Artur Dubrawski. N-hits: Neural hierarchical interpolation for time series forecasting. *CoRR*, abs/2201.12886, 2022. URL <https://arxiv.org/abs/2201.12886>.
- Julien Herzen, Francesco LÃ¤ssig, Samuele Giuliano Piazzetta, Thomas Neuer, LÃ©o Tafti, Guillaume Raille, Tomas Van Pottelbergh, Marek Pasieka, Andrzej Skrodzki, Nicolas Huguenin, Maxime Dumonal, Jan KoÅcisz, Dennis Bader, FrÃ©dÃ©rick Gusset, Mounir Benheddi, Camila Williamson, Michal Kosinski, Matej Petrik, and GaÅl Grosch. Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, 23(124):1–6, 2022. URL <http://jmlr.org/papers/v23/21-1177.html>.
- Dattatray P. Gandhmal and K. Kumar. Systematic analysis and review of stock market prediction techniques. *Computer Science Review*, 34:100190, 2019. ISSN 1574-0137. doi:<https://doi.org/10.1016/j.cosrev.2019.08.001>. URL <https://www.sciencedirect.com/science/article/pii/S157401371930084X>.
- Ashish Sharma, Dinesh Bhuriya, and Upendra Singh. Survey of stock market prediction using machine learning approach. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, volume 2, pages 506–509, 2017. doi:10.1109/ICECA.2017.8212715.
- Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL <https://www.gurobi.com>.
- Adam N Elmachtoub and Paul Grigas. Smart “predict, then optimize”. *Management Science*, 68(1):9–26, 2022.
- Bo Tang and Elias B Khalil. Pyepo: A pytorch-based end-to-end predict-then-optimize library for linear and integer programming. *arXiv preprint arXiv:2206.14234*, 2022.