UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO MATEMATICA

Frank-Wolfe variants for portfolio optimization

Optimization for Data Science

(AY 2022/2023)

Marinelli Andrea (ID: 2091700)

Rinaldi Giorgia (ID: 2092226)

September 18, 2023

Contents

1 Introduction

In the analysis presented below, we consider a constrained optimization problem known as the "Markowitz Portfolio Problem." In order to address this problem, three distinct algorithms are proposed. Following a theoretical exposition of their functioning and a comparative assessment, they are employed to evaluate two datasets: S&P 500 and Euro Stoxx 50. The final section of this paper elucidates the application of the algorithms to the datasets, the results obtained, and the conclusions that can be drawn from the analysis.

2 Markowitz portfolio problem

The Markowitz Portfolio Problem (Markowitz 1952) is a constrained optimization problem. It represents a mathematical approach to investment management aimed at identifying the set of assets that can maximize the portfolio's return while simultaneously minimizing the associated risk.
Let's consider n assets. The amount of money invested in the *i-th* asset is denoted as x_i and the return on the *i-th* asset is denoted as r_i. There are two constraints:

- $x_i \geq 0$: it indicates non - negativity for the variables. It means that selling assets not yet owned (short selling practice) is not allowed;
- $\sum_{i=1}^{n} x_i = B$: the budget constraint indicates that the total amount of money invested has to be equal to the budget (indicated with B and set to 1).

Considering a stochastic model where the return $r \in \mathbb{R}^n$ is a vector having mean $\bar{\mathrm{r}}$ and covariance Σ, then the Expected Return is: $\bar{\mathrm{r}}^\top x$, while Variance is: $x^\top \Sigma x$.
Markowitz Portfolio Problem can be written as a quadratic programming problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & \eta \cdot x^\top \Sigma x - \bar{\mathrm{r}}^\top x \\ \text{s.t.} \quad & \mathrm{e}^\top x = 1 \\ & x \geq 0 \end{aligned}$$

where $\eta > 0$ is the risk-aversion parameter and it indicates the extent of an investor's risk aversion:

- $0 < \eta < 1$: it characterizes a risk-seeking investor who is open to higher risks in pursuit of potentially greater returns.
- $\eta = 1$: it represents an investor adhering to the Markowitz mean-variance approach, aiming to optimize the balance between expected returns and risk by following the efficient frontier;
- $\eta > 1$: it indicates a risk-averse investor who is willing to forgo some return in exchange for risk reduction;

It is important to note that the choice of the risk-aversion parameter value depends on the investor's individual preferences and their specific objectives.
The gradient of the objective function can be expressed as:

$$\nabla f(x) = \eta \cdot (x^\top \Sigma + \Sigma x) - \bar{r}$$

In the following paragraphs three methods to solve this quadratic programming problem are shown.

3 Algorithms

We consider general constrained convex optimization problem:

$$\min_{x \in C} f(x) \qquad C = \text{conv}(\mathcal{A})$$

where $\mathcal{A} \subseteq \mathbb{R}^n$ is a set of vectors we call atoms.

In the specific problem of Markowitz portfolio optimization with a budget of 1, the set $\mathcal{A}$ consists of n vectors, each having only one non-zero component, which is 1, and all others are 0. These vectors are often denoted as $e_1, e_2, ..., e_i, ..., e_n$, where each e_i has a 1 in the i-th position and 0s elsewhere.
The convex set C is the n-dimensional unit simplex representing a set of feasible portfolios. The unit simplex is a geometric shape in n-dimensional space that consists of all possible portfolios where the weights sum up to 1 (hence, the budget of 1). This set is convex because if you have two portfolios inside it, any convex combination of them (a weighted average) will also be inside it.

We now present the Projected Gradient method, the Frank-Wolfe method, and its variants for solving this problem.

3.1 Projected Gradient

The Projected Gradient method is a variant of the Gradient Descent method commonly employed in constrained optimization problems. Due to the constraint imposed by a finite set of feasible points (C), it is not feasible to use the classical Gradient Descent method since it might suggest moving to a point that violates the problem's constraints.

3.1.1 Description

In the following paragraph, a brief description of the algorithm is provided.

1. Inizialization: start from a feasible point $x^{(0)}$;
2. Set $\hat{x}^{(k)} := \rho_C(x^{(k)} - s_k \nabla f(x^{(k)}))$, with $s_k > 0$, where $\rho_C(x)$ represents the projection of x over C and it is computed as
$$\arg\min_{\mathbf{z}\in C} \frac{1}{2}\|\mathbf{z} - \mathbf{x}\|_2^2$$
3. Check if $\hat{x}^{(k)}$ meets the stopping condition and if so, stop;
4. Otherwise compute the next iterate $x^{(k+1)} = x^{(k)} + \gamma_k(\hat{x}^{(k)} - x^{(k)})$, with $\gamma_k \in (0; 1]$;
5. Repeat steps 2 to 4 iteratively until a stopping criterion is met.

The condition used as stopping criterion is $\|\mathbf{g_C}(\mathbf{x^{(k)}})\|_2 \leq \varepsilon$, where $g_C(x^{(k)}) := x^{(k)} - \hat{x}^{(k)}$ is the gradient mapping and ε is a constant > 0. This condition is based on Proposition 5.9 Chapter 5 in [3].
In other words, the method stops when x is close enough to $\hat{x}^{(k)}$. This typically indicates that we are sufficiently close to the optimal solution.

3.1.2 Implementation

There are two ways to implement the Projected Gradient algorithm: fixing s_k and performing a line search on γ_k or vice versa. In the analysis presented in this paper, it was chosen to fix s_k at a value of 115.
As for the stopping condition, a value of ε equal to 10^{-5} was chosen, and a maximum limit on the number of iterations was set at 5000.
Finally, the projection of the point onto the convex set (in our case, the unit simplex) was computed using Algorithm 1 proposed in [4].

3.2 Frank-Wolfe

The Frank-Wolfe algorithm is an optimization method used to solve minimization problems. In recent years, it has attracted significant attention due to its utility in handling structured constraints in machine learning applications. Unlike the Projected Gradient method, it computes iterations in a less computationally expensive manner because it doesn't require a projection operator but instead relies on an oracle (Linear Minimization Oracle, LMO). However, it converges slowly when the optimal point is an extreme point of the constraint set. To address this issue, several variants have been introduced, such as Away-Step FW and Pairwise FW.
It is indeed possible to prove that certain variants of the Frank-Wolfe algorithm, such as Away-Step FW and Pairwise FW, converge linearly under the condition of strong convexity of the objective function.
In our analysis, we have implemented the Frank-Wolfe algorithm and the Pair-Wise Frank-Wolfe algorithm for the constraint represented by a unit simplex.

3.2.1 Description

In the following paragraph, a brief description of the algorithm is provided.

1. Inizialization: start from a feasible point $x^{(0)}$;
2. Set search atom $s_k := \mathrm{LMO}_C(\nabla f(x^{(k)}))$, where $\mathrm{LMO}_C(r) \in \arg\min_{\mathbf{x}\in C}\langle r, x\rangle$, for any given direction $r \in \mathbb{R}^n$;
3. Check if s_k meets the stopping condition and if so, stop;
4. Otherwise compute the next iterate as $x^{(k+1)} = x^{(k)} + \gamma_k(s_k - x^{(k)})$, with $\gamma_k \in (0; 1]$;
5. Repeat steps 2 to 4 iteratively until a stopping criterion is met.

Duality gap

The Frank-Wolfe method employs the concept of the duality gap as its stopping condition.

What is the duality gap?

The duality gap, denoted as $g(x)$, is a concept used in constrained convex optimization problems to assess how close our current solution is to the optimal solution. It

is defined as the maximum difference between the value of the current solution, $f(x)$, and the linear approximation of the objective function around the current solution, which is $f(x) + \langle s - x, \nabla f(x) \rangle$. Mathematically, it's expressed as:

$$g(x) := \max_{s \in C} \langle \nabla f(x), x - s \rangle = - \min_{s \in C} \langle \nabla f(x), s - x \rangle$$

In the Frank-Wolfe Method, 's' represents the search atom, which is the minimizer of the linearized problem solved at a certain step of the algorithm.

How does it provide a certificate for the current approximation quality?

The key insight here is that, due to the convexity of the function f, the linearization of the objective function always lies below the graph of the function f. This leads to the following inequality:

$$f(s) \geq f(x) + \langle \nabla f(x), s - x \rangle, \quad \forall s \in C$$

Minimizing on both sides of the inequality over C and rearranging, we get

$$f(x) - f(x^*) \leq - \min_{s \in C} \langle \nabla f(x), s - x \rangle = g(x)$$

So, if we want a primal gap $0 \leq f(x) - f(x^*) \leq \varepsilon$, we need:

$$f(x) - f(x^*) \leq g(x) \leq \varepsilon$$

Therefore, we can stop the optimization method when:

$$\langle \nabla f(x^{(k)}), (s_{(k)} - x^{(k)}) \rangle \geq -\varepsilon$$

Curvature coefficient

Another crucial concept extensively employed in the analysis of the convergence of the Frank-Wolfe algorithm and its variations is the curvature constant C_f.

What is the curvature constant C_f?

The curvature constant C_f is a metric used to quantify the level of "curvature" or "non-linearity" present in a function. A bounded C_f indicates that during one iteration of the algorithm, we expect the deviation between the new solution $f(x^{(k+1)})$ and the linear approximation of the objective function around the current solution $f(x^{(k)}) + \langle s_k - x^{(k)}, \nabla f(x^{(k)})\rangle$ to be less than C_f. Here, s_k represents the search atom selected by the algorithm. A formal definition of the curvature constant C_f of a convex and differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, with respect to a compact domain C is defined as:

$$C_f := \sup_{\substack{x,s\in C \\ \gamma\in[0,1] \\ y=x+\gamma(s-x)}} \frac{2}{\gamma^2}\left[f(y) - f(x) - \langle y - x, \nabla f(x)\rangle\right]$$

Relationship between Curvature Constant and Lipschitz Constant L

The Lipschitz constant L refers to how rapidly the gradient of a function changes across its domain. In simple terms, if a function has a low Lipschitz constant, it means that its gradient changes gradually, making the function relatively "linear" in small neighborhoods of points.
Theoretical insight into the connection between the curvature constant C_f and the Lipschitz constant L is presented in the following excerpt, extracted from Lemma 7 [1].

For any choice of norm $\|\cdot\|$, the curvature constant C_f can be upper bounded as follows:

Let f *be a convex and differentiable function with its gradient* ∇f *is Lipschitz-continuous w.r.t. some norm* $\|\cdot\|$ *over the domain* C *with Lipschitz constant* $L > 0$.

Then

$$C_f \leq \text{diam}_{\|\cdot\|}(C)^2 L$$

In summary, a low Lipschitz constant suggests a more "linear" function and limited curvature between points in its domain, whereas a high Lipschitz constant suggests a more "non-linear" function with higher curvature. Curvature and the Lipschitz constant are interconnected concepts that help us better understand the shape of a function and how we can efficiently optimize it.

Frank-Wolfe variants

In this paragraph, we present some alternative strategies that can be employed for certain steps of the classic Frank-Wolfe algorithm.

Approximate Linear Subproblems

When computing the exact value of the search atom s_k becomes prohibitively expensive, it is possible to approximate it using an error of $\xi = \frac{\delta \gamma C_f}{2}$, where $\delta \geq 0$ is an arbitrary fixed accuracy parameter. Thus, we choose an $s_k \in C$ such that

$$\langle s, \nabla f(x^{(k)}) \rangle \leq \langle s^*, \nabla f(x^{(k)}) \rangle + \xi$$

Alternative Step-size

It is possible to demonstrate that for a convex function f with a Lipschitz continuous gradient $L > 0$, by fixing a decreasing step size $\gamma = \frac{2}{k+2}$, the Frank-Wolfe method converges. Alternatively, one can choose the step size by performing a line search:

$$\gamma := \arg \min_{\gamma \in (0;1]} f(x^{(k)} + \gamma(s_k - x^{(k)})$$

Fully Corrective Frank-Wolfe

This variation of the Frank-Wolfe algorithm, upon selecting a new atom s, re-optimizes the objective function over the currently utilized atoms. This promotes improved sparsity of the solution. This algorithm encompasses various variants based on how the correction is performed and how atoms are maintained. It is advantageous to employ this approach when the correction is less costly than the linear oracle.

Away-Steps Frank-Wolfe

Unlike the classic Frank-Wolfe algorithm, this variant enables movement away from an active atom. This method addresses the zig-zag issue, enhances the sparsity of iterates, and, in certain scenarios, leads to accelerated linear convergence.

Pairwise Frank-Wolfe

Pairwise Frank-Wolfe transfers mass weight exclusively between two atoms in each step. More precisely, it shifts weight from the away atom v_k to the search atom s_k while leaving all other weights unchanged.
In the next section this method will be discussed more accurately.

3.2.2 Implementation

In the implementation of the algorithm, we computed the search atom s_k as e_{i_k}, with $i_k = \text{argmin}_i \nabla_i f(x^{(k)})$, given that our constraint is the unit simplex. This simplifies the computation since there is no need to invoke the Linear Minimization Oracle (LMO). The computation has a linear cost because, if the constraint is a polytope (in our case, the unit simplex), by the fundamental theorem of linear programming, the solution to the linear problem is one of the vertices of the polytope.
As stopping criteria in our research, we employed the condition $g(x) \leq \varepsilon$, with ε fixed at 10^{-5}. Additionally, we set a maximum iteration limit of $\frac{1}{\varepsilon}$ due to the Theorem 2 [1] that states that a small duality gap is guaranteed after $\mathcal{O}(\frac{1}{\varepsilon})$ iterations.

3.3 Pairwise Frank-Wolfe

The Frank-Wolfe Pairwise algorithm is an advanced variant of the Frank-Wolfe algorithm, used to solve constrained optimization problems. This technique stands

out for its specific approach to finding the descent direction within the constraint domain. While the classic Frank-Wolfe focuses on finding a single point within the domain that minimizes the objective function, the Frank-Wolfe Pairwise adopts a "pairwise" approach that considers two points within the domain at a time. This distinctive feature makes it particularly suitable for problems where the objective function consists of a combination of linear terms involving pairs of variables.

3.3.1 Description

In the following paragraph, a brief description of the algorithm is provided.

1. Inizialization: start from a feasible point $x^{(0)}$;
2. Let $s_k := \text{LMO}_{\mathcal{A}}(\nabla f(x^{(k)})$;
3. Let $v_k := \arg\max_{v \in S^{(k)}} \langle \nabla f(x^{(k)}, v - x^{(k)} \rangle$;
4. Set $d_k := d_k^{\text{PFW}} = s_k - v_k$ and $\gamma_{\max} = \alpha_{v_k}$;
5. Check if s_k meets stopping condition and if so, stop;
6. Otherwise set $x^{(k+1)} = x^{(k)} + \gamma_k d_k$, with $\gamma_k \in (0, \gamma_{\max}]$;
7. Compute $S^{(k+1)}$ set of currently used vertices;
8. Repeat steps 2 to 7 iteratively until a stopping criterion is met.

Here α_v represents the weight associated to atom v.
Note that $\gamma_{\max}$ represents the maximum step size we can take along the direction d_k from $x^{(k)}$ without violating the problem's constraints.
Finally the set of currently used vertex $S^{(k)}$ is updated. Notice that a pair-wise FW step can be interpreted as a swap of mass from the away atom v_k to FW atom s_k. When γ is set to $\gamma_{\max}$ we call this step "drop step" as it fully removes the atom v_k from the active set $S^{(k)}$. Additionally, when a new atom s_k is discovered, it is included in the active set $S^{(k+1)}$.

3.3.2 Implementation

In the implementation of the method, we computed s_k as in the previous algorithm. Meanwhile, we calculated v_k as $\arg\max_{v \in S^{(k)}} \langle \nabla f(x^{(k)}, v - x^{(k)} \rangle$. We set $\gamma_{\max}$ as the weight associated with v_k, representing the maximum allowed mass swap between v_k and s_k. The stopping criteria are the same as those in the classic Frank-Wolfe

algorithm that are $g_k^{FW} := \langle -\nabla f(x^{(k)}), s_k - x^{(k)} \rangle \leq \varepsilon$ and imposing a maximum number of iterations of $\frac{1}{\varepsilon}$ as in the classical Frank-Wolfe.

4 Implementation and testing

In this section, the three algorithms previously described are employed to determine the optimal portfolio for an investor with a risk aversion level of 1. Two sets of assets were considered: S&P 500 and EuroStoxx 50. Following a brief explanation of these datasets, the results obtained are presented with the aim of conducting a performance comparison of the three algorithms.

4.1 Datasets: S&P500 and EuroStoxx50

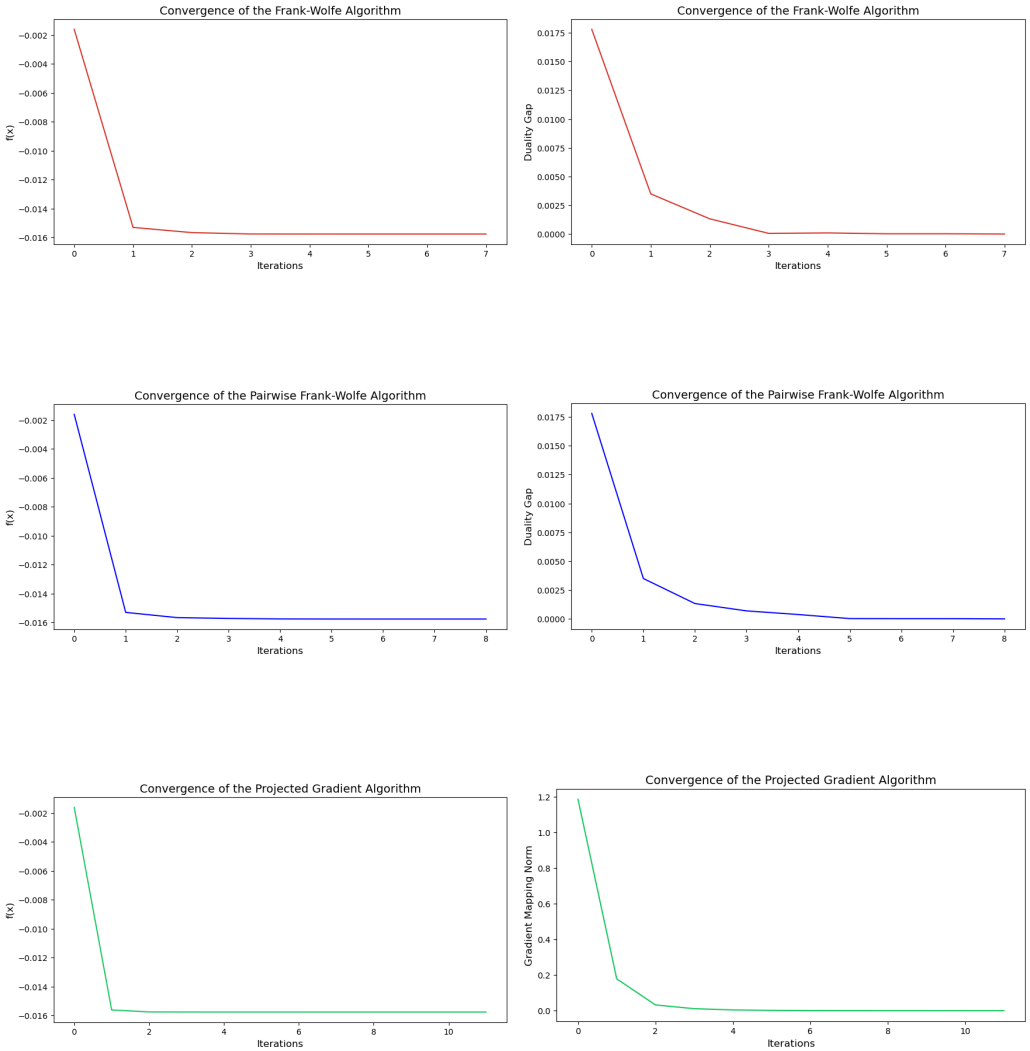
Two datasets have been used in the analysis. The first one, S&P 500 dataset, is a collection of data that tracks the performance of the S&P 500 index, which is one of the most closely monitored and representative indices of the U.S. stock market. This index comprises the stocks of the top 500 publicly traded companies in the United States and is widely employed by investors and analysts to assess the overall health of the U.S. stock market and make investment decisions. The dataset used contains information on 476 assets collected over the five-year period from March 2003 to March 2008.

The second one, Euro Stoxx 50, consists of data that monitors the Euro Stoxx 50 index's performance. This index is a prominent and indicative benchmark for the European stock market, encompassing the stocks of the 50 leading publicly traded companies in the Eurozone. It is extensively utilized by investors and analysts to gauge the overall condition of the European stock market and to inform their investment choices. The dataset used contains information on 48 assets collected over the five-year period from March 2003 to March 2008.

4.2 Analysis and results

S&P500 dataset

Here are the graphs illustrating the convergence of the Frank-Wolfe, Pairwise Frank-Wolfe, and Projected Gradient algorithms for the S&P500 dataset with the risk aversion parameter $\eta = 1$.



The graphs depict the convergence of the three algorithms, with Frank-Wolfe represented in red, Pairwise Frank-Wolfe in blue, and Projected Gradient in green.

1. **Frank-Wolfe Algorithm**: The classical Frank-Wolfe method converges swiftly, requiring only 8 iterations to reach a solution.
2. **Pairwise Frank-Wolfe Algorithm**: Pairwise Frank-Wolfe also converges promptly, taking just 9 iterations to reach convergence.
3. **Projected Gradient Method**: The Projected Gradient method also converges rapidly, requiring 11 iterations.

The three algorithms on this dataset exhibit excellent performance, which is similar among them, achieving convergence in approximately 10 iterations.

Below are two graphs: Figure 1 illustrates the convergence of the three algorithms,

while Figure 2 provides a comparison between the classical Frank-Wolfe method and the Pairwise Frank-Wolfe. As can be observed from the nearly overlapping curves, there are no significant differences among the three algorithms.

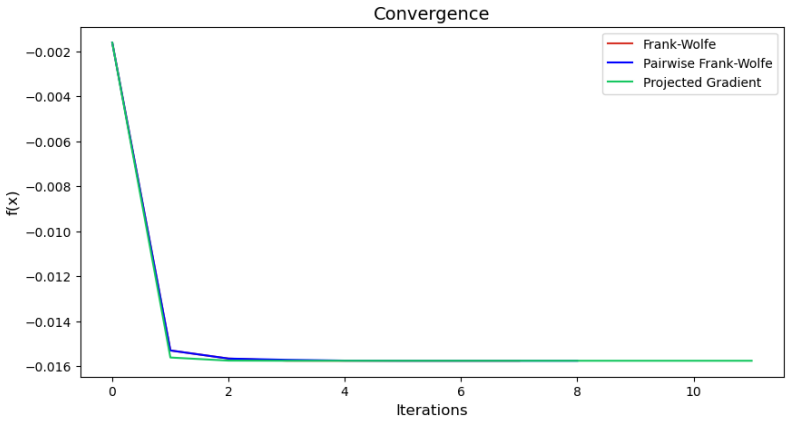


Figure 1: Comparison of the three algorithms

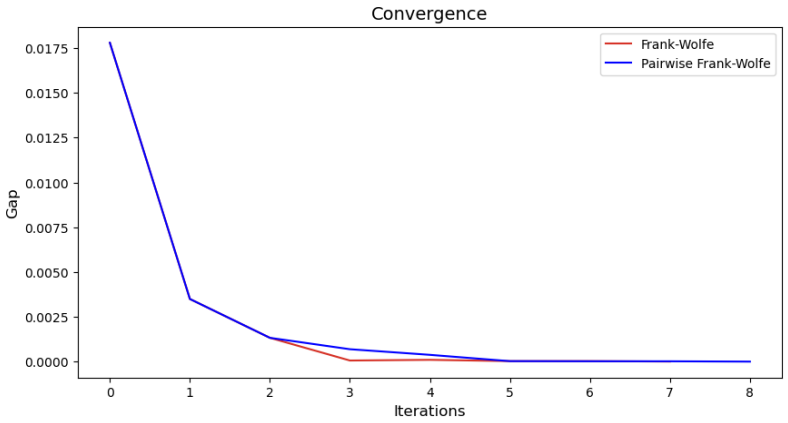


Figure 2: FW vs PFW

EuroStoxx50 dataset

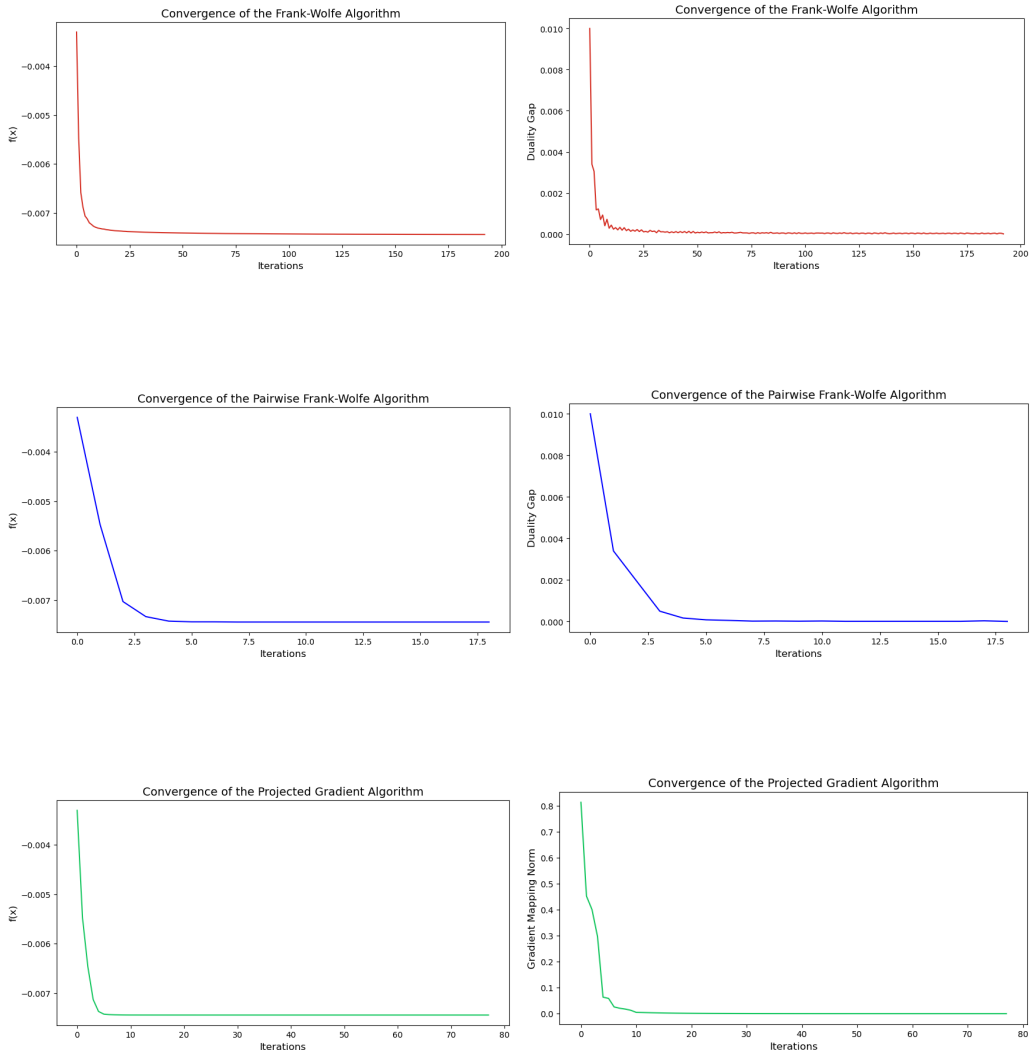
The following figures show the convergence behavior of the three distinct algorithms applied to the EuroStoxx50 dataset with a risk aversion parameter of $\eta = 1$. The primary objective is to assess their convergence behaviors and relative performance.

1. **Frank-Wolfe Algorithm**: This algorithm exhibits moderate convergence, requiring approximately 200 iterations to reach a solution.

2. **Pairwise Frank-Wolfe Algorithm**: This alternative approach demonstrates

remarkable efficiency, converging notably faster after a mere 19 iterations. Such rapid convergence is advantageous in reducing computational effort.

3. **Projected Gradient Method**: The Projected Gradient method also exhibits good performance and achieves convergence in approximately 80 iterations.



The chart in Figure 3 compares the convergence of the three algorithms and confirms what was previously observed. Unlike the classical Frank-Wolfe method, the other two algorithms exhibit a more pronounced decrease. In Figure 4, the classical Frank-Wolfe algorithm is compared with the Pairwise Frank-Wolfe variant, with the latter proving to be the most efficient, converging more rapidly.

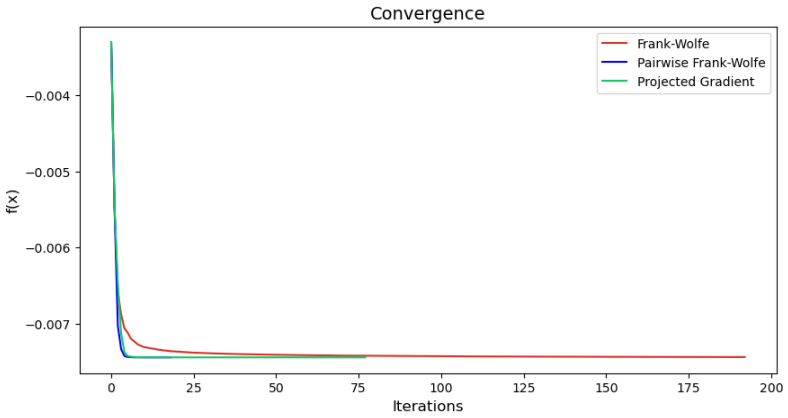


Figure 3: Comparison of the three algorithms

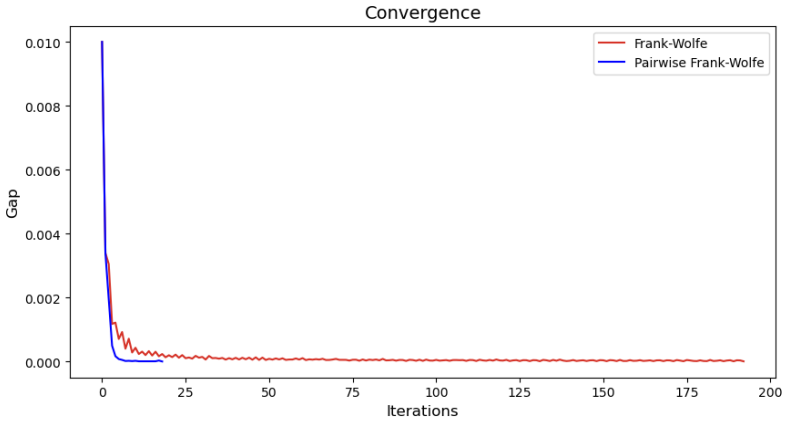


Figure 4: FW vs PFW

5 Conclusions

In conclusion, our analysis of the Markowitz Portfolio Problem involved a thorough examination of three optimization algorithms: Frank-Wolfe, Pairwise Frank-Wolfe, and Projected Gradient. We investigated their convergence behaviors and efficiency when applied to two datasets, the S&P 500 and Euro Stoxx 50, with a particular focus on their performance in terms of convergence speed.

In the case of the S&P 500 dataset, we observed that the classical Frank-Wolfe, Pairwise Frank-Wolfe, and Projected Gradient methods all exhibited rapid convergence, with minimal difference between them.

However, in the case of the EuroStoxx50 dataset, the differences were more pronounced. Pairwise Frank-Wolfe and Projected Gradient algorithms showed remarkable efficiency, converging faster than the classical Frank-Wolfe method. In par-

ticular, the Pairwise Frank-Wolfe algorithm emerges as the most efficient in terms of cost-effectiveness, as it conducts iterations with lower computational overhead compared to the Projected Gradient method, which necessitates the use of the projection operator. Furthermore, in contrast to the classical Frank-Wolfe algorithm, Pairwise Frank-Wolfe appears to be better suited for addressing this problem of optimal resource allocation. This is attributed to its ability to promote sparsity, enabling a departure from less favorable assets in the portfolio allocation process.

Bibliography

[1] Jaggi, Martin (2013), *Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization*, CMAP, Ecole Polytechnique, Palaiseau, France.

[2] Lacoste-Julien, Simon and Jaggi, Martin (2015), *On the Global Linear Convergence of Frank-Wolfe Optimization Variants*

[3] *Optimization for Data Science course material* (2023), `https://stem.elearning.unipd.it/course/view.php?id=5040`

[4] Condat, Laurent (2016), *Fast projection onto the simplex and the* l_1 *ball*, Mathematical Programming volume 158, pages 575–585 (2016), `https://doi.org/10.1007/s10107-015-0946-6`