COMP6212 - Computational Finance
Assignment I

Sebastian Gherhes - sag1g15

28018788

1 Markowitz Efficent Frontier

For this part, we have been given two assets with the following expected returns and covariance matrix

$$m = \begin{bmatrix} 0.10 & 0.10 \end{bmatrix}^t \tag{1}$$

$$C = \begin{bmatrix} 0.005 & 0.0 \\ 0.0 & 0.005 \end{bmatrix} \tag{2}$$

It is known that the sum of weights should add up to 1: $\sum_i^N \pi_i = 1$. In this case, we have 2 assets, which means the weight vector will look like $(\pi,\ 1-\pi)$. The return of the portfolio is thus given by $r = \sum_i^N r_i \pi_i$. The Markowitz optimal portfolio is given by optimization problem:

$$\min \quad \frac{1}{2} w^T \Sigma w \text{ subject to } \begin{cases} m^T \geqslant \mu_b \\ e^T w = 1 \end{cases} \tag{3}$$

Burke (2014) talks about Markowitz Portfolio in their paper and how it can be calculated from the KKT conditions. The conditions that need to be satisfied for this quadratic problem are the following:

$$\begin{aligned} \Sigma w - \lambda m - \gamma e = 0 \\ \mu_b \leqslant m^T w, e^T w = 1, 0 \geqslant \lambda \\ \lambda^T (m^T w - \mu_b) = 0; \end{aligned} \tag{4}$$

Thus, finding a solution for the KKT conditions results in finding the weight vector as a solution for our Markowitz portfolio. The case where $\mu_b < m^T w$ is taken into consideration. From Equation 4 we can say that $\lambda = 0$. We also know that $e^T w = 1$ and that e is a vector where each element is 1, which gives us the equation:

$$\Sigma w - \gamma e = 0 | * \Sigma^{-1} \implies w = \gamma \Sigma^{-1} e | * e \implies \gamma = (e^T \Sigma^{-1} e)^{-1} \tag{5}$$

Substituting γ back in the w equation gives us the value of our weight vector. The return is given by:

$$\mu_{min-var} = \frac{m^T \Sigma^{-1} e}{e^T \Sigma^{-1} e} \tag{6}$$

After demonstration, we then solve our equations with the mean and covariance given for our two assets.

$$w = \frac{\Sigma^{-1}e}{e^T\Sigma{-}1e} = \frac{\begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}}{\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}} = \frac{\begin{bmatrix} 200 \\ 200 \end{bmatrix}}{\begin{bmatrix} 200 & 200 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \tag{7}$$

$$\mu_{min-var} = \frac{m^T\Sigma^{-1}e}{e^T\Sigma{-}1e} = \frac{\begin{bmatrix} 0.1 & 0.1 \end{bmatrix}\begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}}{\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}} = 0.1 \tag{8}$$

$$\gamma = \frac{1}{e^T\Sigma{-}1e} = 0.0025 \tag{9}$$

The expected return will always be 0.10 in this case, and the variance will vary depending on the weights used. The weights used to reproduce the line varied between 0-1 between the two assets, always making sure the sum of them is 1. Figure 1 illustrates the outcome of calculating the efficient frontier.

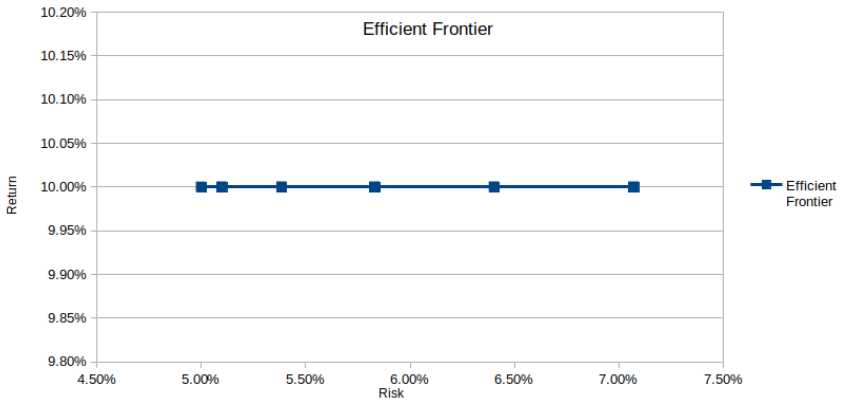


Figure 1: The Efficient Frontier of the portfolio of two assets with return and covariance matrix mentioned in 1 and 2. The frontier is linear.

2 Portfolio Optimization

2.1 Three Assets

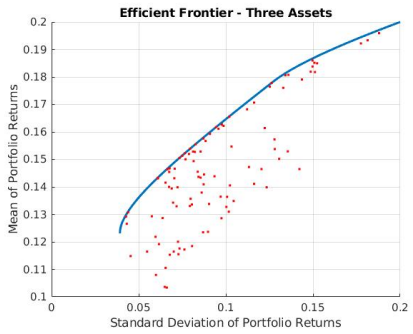
For the first part we were given three assets with their respective returns and covariances:

$$m = \begin{bmatrix} 0.10 & 0.20 & 0.15 \end{bmatrix}^t \tag{10}$$

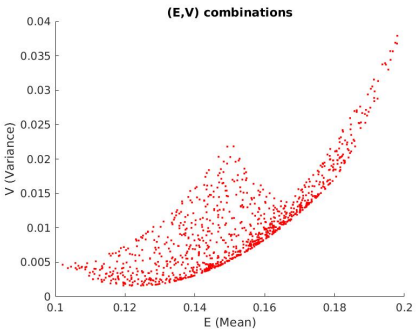
$$C = \begin{bmatrix} 0.005 & -0.010 & 0.004 \\ -0.010 & 0.040 & -0.002 \\ 0.004 & -0.002 & 0.023 \end{bmatrix} \tag{11}$$

A helper function called randfixedsum[1] was used to generate 100 random portfolios π_i. The entries on each row add up to 1, corresponding to the weights of each portfolio. Given the mean and covariance of the three assets, we then calculated the expected return and variance of each portfolio.

[1] https://www.mathworks.com/matlabcentral/fileexchange/9700-random-vectors-with-fixed-sum?focused=5064802&tab=function



(a) Efficient Frontier

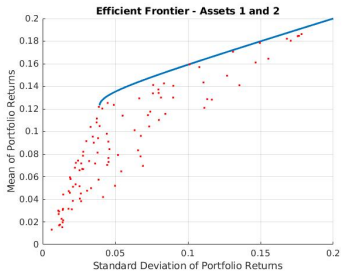


(b) E-V scatter diagram

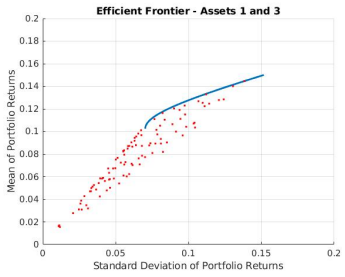
Figure 2: The Efficient Frontier for the three assets with 100 randomly generated portfolios. Diagram of 1000 portfolios scattered in the E-V space as illustrated in Markowitz paper. Initial requirement was to generate the scatter using only 100 portfolios, but 1000 were used for this graph to better illustrate the scatter of the data. (Markowitz 1952)

2.2 Two assets taken pair wise

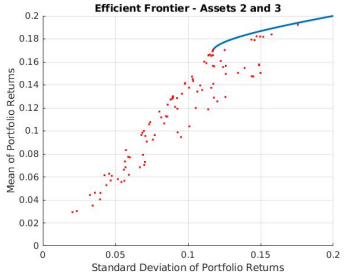
Looking at Figure 3 it can be noticed how portfolios fall outside the efficient frontier as opposed to Figure 2a. The Frontier calculated for all three assets is balanced compared to the other ones. The weights used for the pair-wise calculation are subsets of two taken from the same weights that were used for the three assets frontier.



(a) Assets 1 and 2



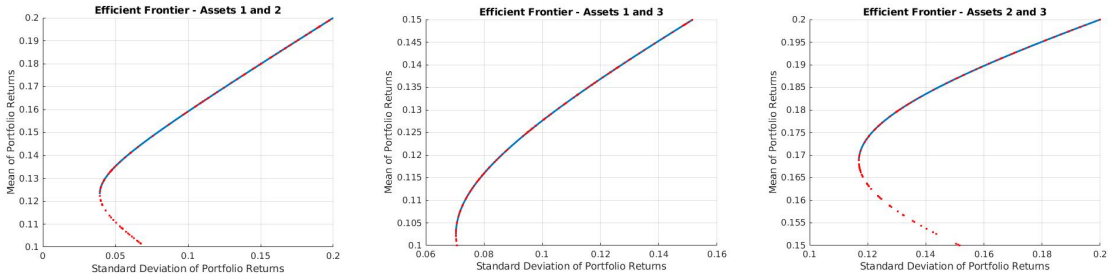
(b) Assets 1 and 3



(c) Assets 2 and 3

Figure 3: The three assets taken pairwise to calculate the efficient frontier and draw the portfolios on it. Weights were taken as subsets of 2 from the initial weights.

Furthermore, a different set of weights has been generated for the pair-wise calculation, the number of columns being the same as the number of assets, in this case 2. Analyzing Figure 4 one can notice how each portfolio lies on the Efficient Frontier, except for the ones that have very low return and high risk. The ones situated on the line are the most optimal ones in terms of return-risk ratio and they are both directly proportional to each other, i.e. when one increases, the other increases as well and vice versa.



(a) Assets 1 and 2 (b) Assets 1 and 3 (c) Assets 2 and 3

Figure 4: The three assets taken pairwise to calculate the efficient frontier and draw the portfolios on it. New weights were generated to draw the efficient frontier.

3 NaiveMV vs ConvexMV

For this part, the optimization steps in the NaiveMV function has been replaced using the CVX Convex Programming toolbox in MATLAB. The code written to reimplement the functions is shown below:

Table 1: Table with the code used to reimplement the optimization functions.

Linear Programming

$$\min_x f^T x \text{ such that } \begin{cases} Ax \leq b \\ A_{eq}x = b_{eq} \\ lb \leq x \leq ub. \end{cases} \quad (12)$$

```
cvx_begin quiet
    variable MaxReturnWeights(NAssets)
    maximize(ERet' * MaxReturnWeights)
    subject to
        V1 * MaxReturnWeights == 1
        V0 <= MaxReturnWeights
cvx_end
```

Quadratic Programming

$$\min_x x^T H x + f^T x \text{ such that } \begin{cases} Ax \leq 0 \\ A_{eq}x = b_{eq} \\ lb \leq x \leq ub. \end{cases} \quad (13)$$

```
cvx_begin quiet
    variable MinVarWeights(NAssets)
    minimize(MinVarWeights' * ECov *
        MinVarWeights + V0' *
        MinVarWeights)
    subject to
        V1 * MinVarWeights == 1
        V0 <= MinVarWeights
cvx_end
```

The optimization steps produced by the CVX toolbox show a similar result as the NaiveMV function. Figure 5 shows the two functions overlapping with each other. Calculating the Mean Squared Error between the two gives a small value of 1.7295e-14, from which one can conclude that they are identical.

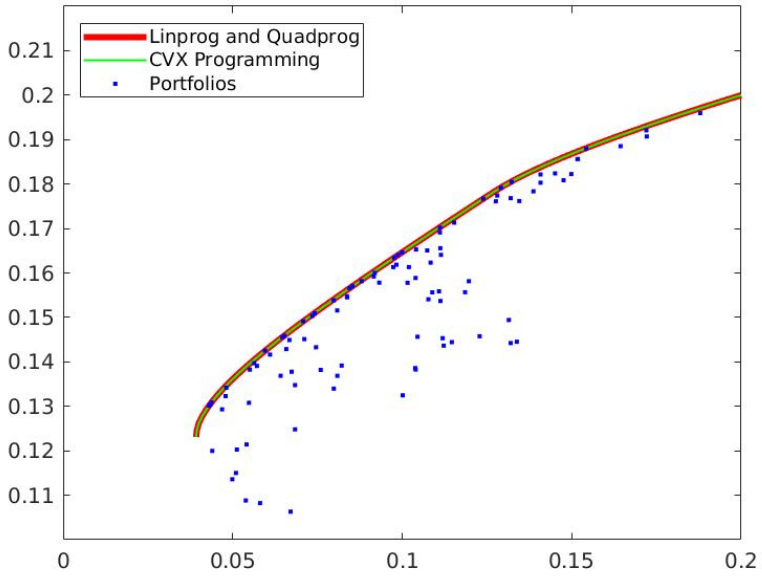


Figure 5

4 FTSE 100 data

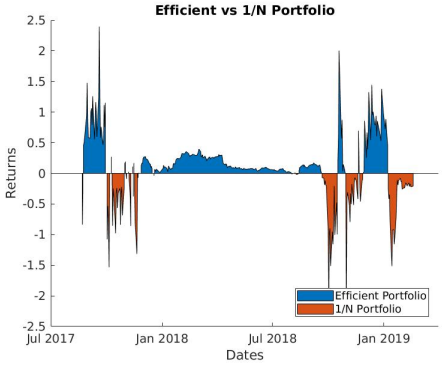
For this part, one had to download data for the daily FTSE 100 for the past three years and data for 30 companies that are to be found in the FTSE index. The stocks information has been downloaded from Yahoo Finance[2]. The data was selected for the period of 19 February 2016 to 19 February 2019. Before selecting the three random stocks that the author will use for this exercise, one had to sanitize the obtained data in order to ensure consistency and eliminate anomalies. First, missing values had to be calculated and included in the stocks. For this, Python's[3] library called Numpy[4] has been used. This package includes a linspace function that returns numbers from a uniform distribution over a specified interval. This calculated the missing values and added them to the spreadsheet of each stock.

After data has been cleaned, three stocks have been randomly selected and their expected returns and covariances from their first half years have been calculated using the formulas shown below:

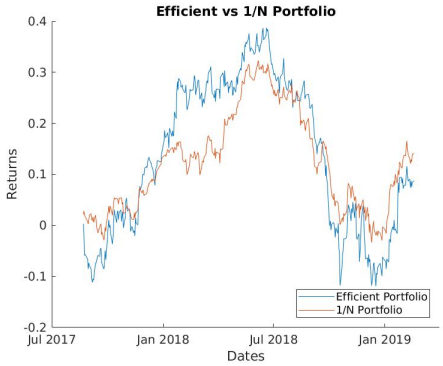
$$\begin{aligned} \mu &= \frac{1}{N}\sum_i r_i \\ \Sigma &= \frac{1}{N}\sum_i (r_i - \mu)(r_i - \mu)^T \end{aligned} \tag{14}$$

We would refer to this as our training data for the three assets. Sharpe (1994) introduces the Sharpe Ratio, $\frac{R-r_f}{V}$, which has been used to get the most Efficient Portfolio. After which, both portfolios were tested on the remaining half of the data set, our testing data. The performance is similar to the one presented by DeMiguel et al. (2007) in their paper, showing that the optimal portfolio performs slightly worse than the $\frac{1}{N}$, which was also the case in this situation.

[2]https://uk.finance.yahoo.com/quote/%5EFTSE/components?p=%5EFTSE
[3]https://www.python.org/
[4]http://www.numpy.org/



(a) $\log_{10}$



(b) Cumulative sums

Figure 6: Graph with the returns of the efficient and $\frac{1}{N}$ portfolio. The first one is a $\log_{10}$ of the returns to illustrate which stock was on top at given dates and the second one is a cumulative sum of the returns to show the fluctuations of the returns on each date. The assets used to produce these graphs are: TSCO, EZJ and ANTO.

5 Shortsale-constrained portfolio

DeMiguel et al. (2007) express the maximisation of a portfolio with the following equation:

$$\max_x x^T \mu - \frac{\gamma}{2} x^T \Sigma x. \tag{15}$$

This equation offers a solution in terms of return and covariance, which can be further derived to calculate the weights of the portfolio. Equation 15 is not constrained by short sale. The author mentions the Lagrangian multiplier which is expressed in terms of λ and is added to the previous maximization which yields the following equation:

$$\mathcal{L} = \text{return} - \text{variance} + x^T \lambda. \tag{16}$$

This constraint for the short sales just means that the weights have to be $\geqslant 0$ and positive. This can be expressed as an optimization problem such as the one similar to the previous implementation in Section 3:

$$\min \quad x^T \mu - \frac{1}{2} x^T \Sigma x \text{ subject to } \begin{cases} \sum_i w_i = 1 \\ w \geqslant 0 \end{cases} \tag{17}$$

6 Greedy Selection and Sparse Index Tracking

For this part, the same 30 companies from Section 4 and the FTSE index have been looked over. After, a comparison between the two index tracking methods has been performed.

6.1 Greedy Forward Selection

This algorithm uses a greedy strategy that seeks to find an optimal portfolio that selects a fifth of the total number of stocks. The immse[5] Matlab function has been used to calculate the minimal error between the returns of each stock and the FTSE index. At each increment, the errors were stored in an array that

[5]https://uk.mathworks.com/help/images/ref/immse.html

was later sorted and the index for the 6 best stocks for the portfolio have been selected. Figure 7 showcases the returns of each stock compared to the FTSE index. While the algorithm is not an optimal choice of selection, it had a reasonable performance on the data set. The Mean Squared Deviation of this selection is 0.0022.

An improved method for the greedy algorithm would be to find all the combinations of each pair of 6 stocks and to calculate the weights for each pair. After which, each pair will be compared to each other and the optimal portfolio would be found. The maximum possible combinations are of the form $\binom{30}{6}$, which yields us 593,775 possible combinations of portfolios.

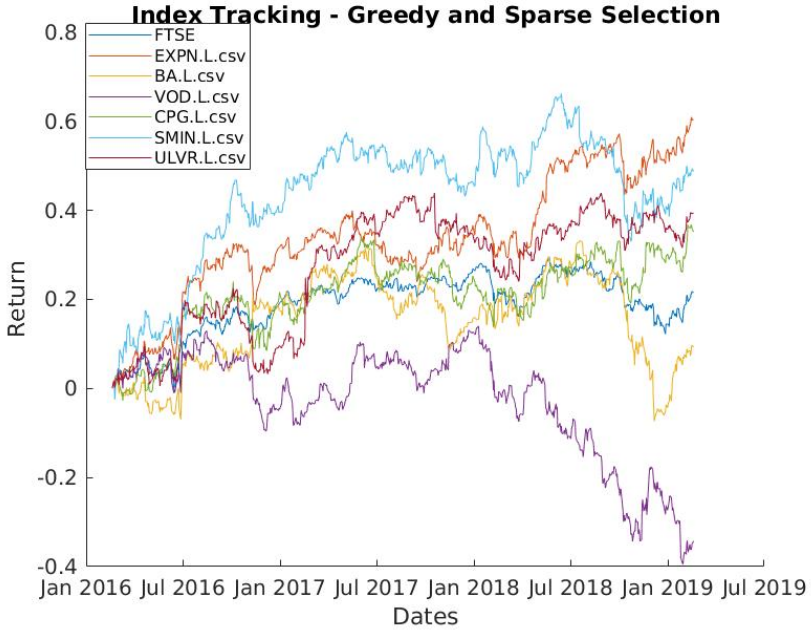


Figure 7: The 6 stocks selected by our greedy selection compared to the FTSE index. As one can notice, their values added together get pretty close to the FTSE index. Selected stocks can be seen in the legend of the graph. The indexes for the stocks are [11;4;30;9;22;29].

6.2 Sparse Index Tracking

The Sparse Index Tracking seeks to minimize the optimization problem by introducing a *l1*-penalty to our optimization. The following term is added to our equation:

$$\min_{w} \quad \|\rho 1_T - Rw\|_2^2 + \tau \|w\|_1 \text{ such that } \begin{cases} w^T \mu = \rho \\ w^T 1_N = 1 \end{cases} \tag{18}$$

with τ being our regularization parameter that acts on the norm of our weights, R are the stocks and w the parameter that is minimized so that our return/risk ratio stays as high as possible. (Brodie et al. 2009)

The aim for the Sparse Index Tracking was to tune the $l1$ parameter so that our sparse tracking produces a similar result as our greedy selection. The author has tried different values for τ, ranging from positive values to negative values and during the implementation process the optimal τ that generated the same outcome is 0.06. The Mean Squared Error for the Sparse selection is 0.0529. The selected indexes for the Sparse Tracking are [1;2;5;17;22;27].

The Greedy Forward Selection performed better than the Sparse method in terms of the errors produced by the two.

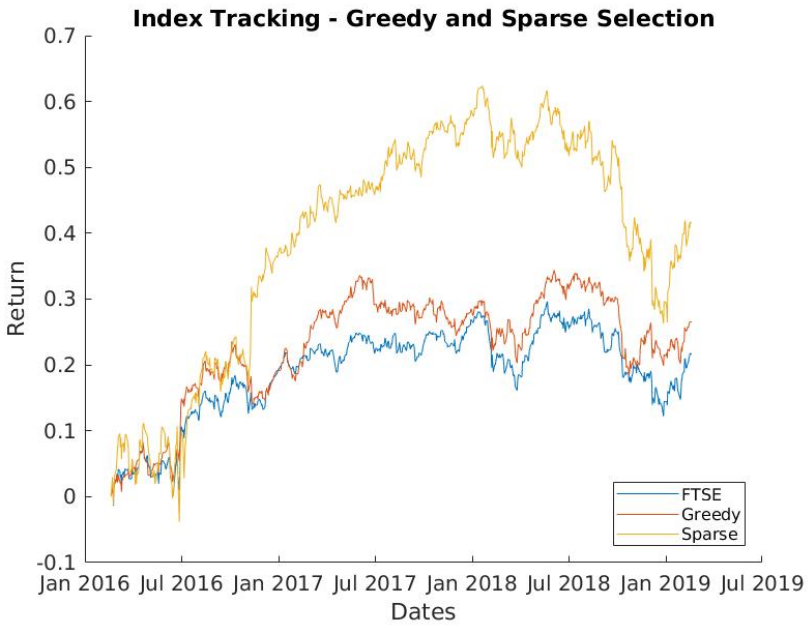


Figure 8: The cumulative sums of the Greedy Selection, FTSE index and Sparse selection. The graph illustrates the poor performance of the lasso method compared to the Greedy selection.

7 Transaction costs

The transaction cost constraint is introduced as an inequality equation that is used in the maximization method of the portfolio selection. This constraint is defined as a cost function which has to be minimized in order to get the minimum transaction costs for the portfolio. The inequality constraint as well as the minimization problem is given below:

$$0 \geqslant 1^T x + \phi, (x) \qquad \min \quad \phi(x) \text{ subject to } \begin{cases} a^T(w+x) \geqslant r_{min} \\ w + x \in S. \end{cases} \tag{19}$$

Lobo et al. (2007) paper proposes a return over probability graph of a cumulative function taken from a Gaussian distribution. It is a return function of an efficient portfolio. Because it is a Gaussian, the expected return lies at probability 0.5, the return being the true mean of the Gaussian function. The authors impose two limits which show the probabilities of a bad return to happen in regards to the distribution. The chances of having a bad return are relatively small.

The authors then introduce another optimization problem for a portfolio, where the combination of convex function and constraints yield a convex optimization problem. Their objective is to maximize the function $r^T(w + x^+ - x^-)$ by taking into account the transaction costs and the given constraints. This function is maximized with respect to our weights and the result will be the most optimal portfolio on the efficient frontier.

The implementation of this convex problem would be done similarly to how the CVX Programming has been achieved in Section 3. We would seek to maximize the weights by adding the constraints and using the parameters presented in the paper.

References

Brodie, J., Daubechies, I., De Mol, C., Giannone, D. & Loris, I. (2009), 'Sparse and stable markowitz portfolios', *Proceedings of the National Academy of Sciences* **106**(30), 12267–12272.

Burke, J. (2014), 'Markowitz mean-variance portfolio theory'.
URL: *https://sites.math.washington.edu/ burke/crs/408/fin-proj/mark1.pdf*

DeMiguel, V., Garlappi, L. & Uppal, R. (2007), 'Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy?', *The review of Financial studies* **22**(5), 1915–1953.

Lobo, M. S., Fazel, M. & Boyd, S. (2007), 'Portfolio optimization with linear and fixed transaction costs', *Annals of Operations Research* **152**(1), 341–365.

Markowitz, H. (1952), 'Portfolio selection, journal of finance 7', *Markowitz HM—1952.*— pp. 77–91.

Sharpe, W. F. (1994), 'The sharpe ratio', *Journal of portfolio management* **21**(1), 49–58.