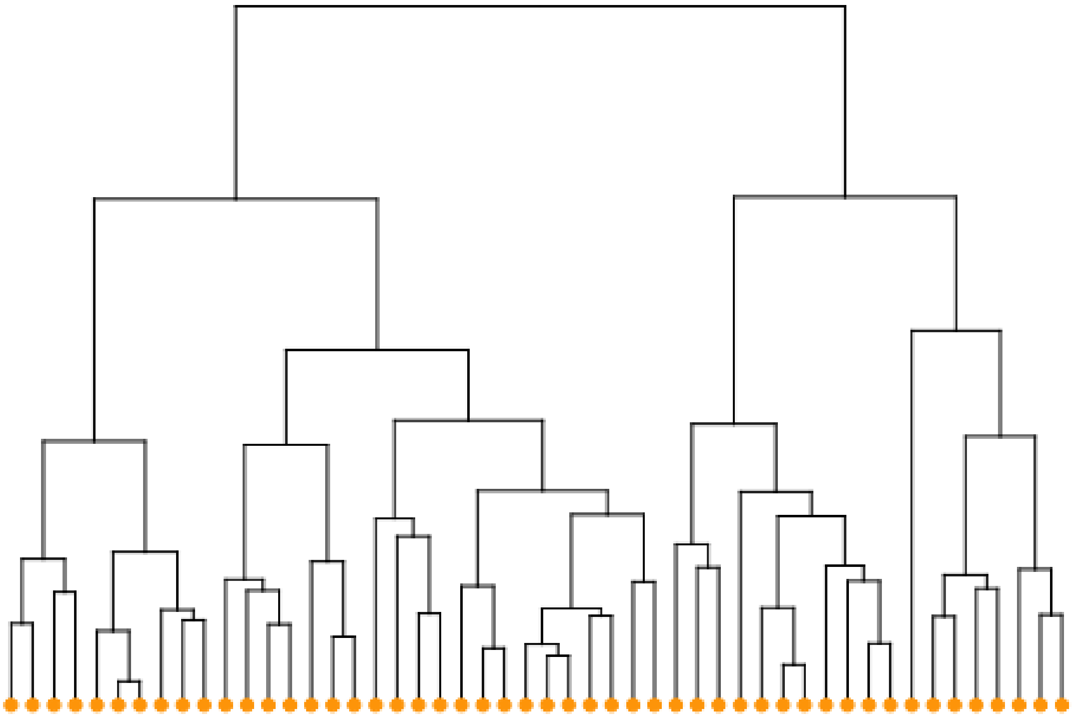


Hierarchical Portfolio Allocation in an Active Management Framework

Master's thesis in Engineering Mathematics and Computational Science

ADNAN DEUMIC

JAMES MEIJER

Mathematical Sciences - Engineering Mathematics and Computational Science

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2022

www.chalmers.se

Master's thesis 2022

Hierarchical Portfolio Allocation in an Active Management Framework

ADNAN DEUMIC

JAMES MEIJER

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
Gothenburg, Sweden 2022

Hierarchical Portfolio Allocation in an Active Management Framework
ADNAN DEUMIC
JAMES MEIJER

Supervisor: Holger Rootzén, Department of Mathematical Sciences
Supervisor: Jan Lennartsson, The Second Swedish National Pension Fund
Examiner: Serik Sagitov, Department of Mathematical Sciences

Master's Thesis 2022
Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Dendrogram of portfolio assets using Ward's Method as linkage criterion.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2022

Hierarchical Portfolio Allocation in an Active Management Framework
ADNAN DEUMIC
JAMES MEIJER
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

This master's thesis focuses on developing and evaluating a hierarchical portfolio allocation algorithm that combines hierarchical clustering and Markowitz Modern Portfolio Theory while being adapted to an active management framework. Sixteen different constellations were constructed and evaluated on equities return data from 01/03/1990 to 10/01/2022, using three different sets of observations as input and five different performance measures.

The results demonstrate that the combination of Equal Risk Contribution and Single Linkage generates the best outcomes. In general, the results also show that Tracking Error is significantly smaller when Equal Risk Contribution is used as a between-cluster allocation method. Moreover, the choice of linkage criteria is crucial for cluster size and the numerical stability of the associated sample covariance matrices. For instance, Single Linkage produces the smallest set of clusters, followed by Group-Average Linkage, Complete Linkage, and Ward's method. In addition, the ordering of the leaves in the hierarchical structure did not have a significant effect on the results. The suggested hierarchical portfolio allocation algorithm performs consistently and is able to capture the hierarchical structure between assets during different market conditions.

Keywords: hierarchical clustering, portfolio allocation, active management, minimum variance, modern portfolio theory, graph theory, covariance matrix, correlation, clustering.

Acknowledgements

First and foremost, we would like to express our sincerest appreciation to our supervisor Holger Rootzén at the Department of Mathematical Sciences. Thank you for making us see the full picture and focusing on the most important aspects of this thesis. Your invaluable expertise and advice have challenged us to make this project reach its full potential. We are forever grateful for the opportunity you have provided.

Further, we would like to extend our sincerest regards to Jan Lennartson at the Second Swedish National Pension Fund for the productive discussions and for guiding us through the world of active management. We always left our meetings with creative ideas and energy to continue to strive forwards.

We would also like to express our gratitude to our extended families and partners for their love and support during this thesis and our time at Chalmers.

Last but not least, we would like to thank our friends Joonas, Nils, Alex, Eric, Simon, Gustav, Johan, and Jonathan, as well as our respective partners Ebba and Helena whom all helped us by providing computing power so that the project could finish on time.

Adnan Deumic & James Meijer
Gothenburg, May 2022

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

CERC	Complete Linkage, Equal Risk Contribution
CEWE	Complete Linkage, Equal Weighting
COERC	Complete Linkage, Optimal Leaf, Equal Risk Contribution
COEW	Complete Linkage, Optimal Leaf, Equal Weighting
DR	Diversification Ratio
ERC	Equal Risk Contribution
FLAM	Fundamental Law of Active Management
GAERC	Group Average Linkage, Equal Risk Contribution
GAEW	Group Average Linkage, Equal Weighting
GAOERC	Group Average Linkage, Optimal Leaf, Equal Risk Contribution
GAOEW	Group Average Linkage, Optimal Leaf, Equal Weighting
GICS	Global Industry Classification Standard
GSI	Gap Statistic Index
HCAA	Hierarchical Clustering Based Asset Allocation
HERC	Hierarchical Equal Risk Contribution
HPAA	Hierarichal Portfolio Allocation Algorithm
HRP	Hierarchical Risk Parity
MDP	Maximum Diversification Portfolio
MPT	Modern Portfolio Theory
MST	Minnimal Spanning Tree
RC	Risk Contribution
RP	Risk Parity
SERC	Single Linkage, Equal Risk Contribution
SEWE	Single Linkage, Equal Weighting
SOERC	Single Linkage, Optimal Leaf Ordering, Equal Risk Contribution
SOEW	Single Linkage, Optimal Leaf Ordering, Equal Weighting
S&P 500	Standard & Poor's 500
TE	Tracking Error
WERC	Ward's Method, Equal Risk Contribution
WEWE	Ward's Method, Equal Weighting
WOERC	Ward's Method, Optimal Leaf Ordering, Equal Risk Contribution
WOEW	Ward's Method, Optimal Leaf Ordering, Equal Weighting

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i,j	Indices for assets
t	Time index

Sets

$\mathcal{D}$	Set of clusters indices for non-singleton clusters
$\mathcal{E}$	Set of cluster indices for singleton clusters
T	Set of time steps

Variables

p	Number of assets
α	Excess returns
$r_i^{(t)}$	Observed return of asset i between day t and $t + \Delta t$
$r_{z_i}^{(t)}$	Normalized return for asset i
$P_i^{(t)}$	Closing price of asset i at day t
$\boldsymbol{w}$	p-dimensional column vector of portfolio weights
$\boldsymbol{w}^b$	p-dimensional column vector of benchmark weights
$\boldsymbol{b}^{(t)}$	Bets at time t
$\boldsymbol{b}^{*,(t)}$	Optimal bets at time t
$r_P^{(t)}$	Portfolio return at time t
$\boldsymbol{r}_b$	Returns of benchmark

$\bar{\boldsymbol{r}}^{(t)}$	Sample mean vector of returns at time t
σ_i	Standard deviation of the returns of asset i
$\rho_{i,j}$	Correlation coefficient of the returns for asset i and j
$\boldsymbol{\Sigma}$	True covariance matrix
$\boldsymbol{C}$	True correlation matrix
$\hat{\boldsymbol{\Sigma}}$	Estimated covariance matrix
$\boldsymbol{S}$	Sample covariance matrix
$\hat{\boldsymbol{C}}$	Sample correlation matrix
$\boldsymbol{D}'$	Dissimilarity matrix
$\boldsymbol{D}$	Distance matrix
$\boldsymbol{S}'$	Similarity matrix
d	Distance measure
$\tilde{d}$	Euclidean distance between columns of distance matrix
c	Concentration ratio
κ	Condition number
μ_C	Mean vector of cluster
TO	Turnover Ratio
W_k	Within-cluster dissimilarity
k^*	Optimal number of clusters
k_{final}	Final number of clusters
Δk	Difference between final number of clusters and optimal number of clusters
$\boldsymbol{\theta}$	Scaling vector

Contents

List of Figures

List of Tables

1 Introduction

Portfolio optimization can be described as the process of allocating capital optimally amongst a set of assets. The objective and constraints of the optimization are determined by the preferences of the investor. Markowitz (1952) introduced Modern Portfolio Theory (MPT), which is foundational in the field of portfolio optimization and has been studied extensively. In essence, Markowitz's mean-variance framework yield the optimal portfolio which either maximizes the expected return given a specified level of risk, or minimizes the risk given a specified level of expected return. This framework provides investors with an intuitive and practical approach to diversification and allocation of capital, which explains its appeal among researchers and practitioners.

Even though MPT is optimal in theory, there are issues that arise when the theory is applied in practice (Kolm et al., 2010). Firstly, MPT requires the estimation of expected returns, and secondly, the estimation and inversion of the covariance matrix. Due to the inherently difficult challenges in estimating expected returns (Jagannathan and Ma, 2003), many researchers and practitioners choose to aim attention on estimating the covariance matrix. This has resulted in an increase in portfolio allocation strategies that are risk-based such as equal risk contribution (Maillard et. al, 2010) and maximum diversification (Choueifaty and Coignard, 2008). Nonetheless, there exist practical difficulties in estimating covariance matrices which can lead to negative consequences for the risk-based allocation methods. The practical difficulties are a result of the need to invert a potentially ill-conditioned estimated covariance matrix, which often is the case in a high-dimensional setting. The inversion operation of a numerically ill-conditioned covariance matrix augments the estimation errors within the matrix, which results in a portfolio allocation that may be far from optimal. DeMiguel et. al. (2009) argue that the naive approach of allocating capital evenly amongst assets outperforms more sophisticated portfolio optimization strategies. This may be a result of the negative effects of errors in covariance matrix estimation that erases the benefits of portfolio optimization.

Consequently, in order to enhance the effectiveness and applicability of portfolio optimization methods that rely on covariance matrix estimation, several different areas of research have attempted to improve robustness and decrease the instability of the estimation of the covariance matrix (Kolm et al., 2010). Such areas of research include shrinkage estimators (Ledoit and Wolf, 2003), denoising of the covariance matrix (López de Prado, 2020), multi-factor models (Fan et. al., 2008) and sparse estimators (Levina et. al., 2008), to name but a few. The intensity of research dedicated to this area highlights the significance of portfolio optimization.

López de Prado (2018) proposed a new portfolio allocation strategy based on graph theory and machine learning that circumvents the need to invert a covariance matrix, and thus circumvent the issues with traditional risk-based optimization methods. The method, referred to as Hierarchical Risk Parity, utilizes hierarchical clustering to find groups of assets with related characteristics, and allocate capital using a risk-based approach. Simon (1962) and Billio et. al. (2012) suggest that the financial markets can be viewed as a complex system with a structure that can be described as hierarchical with great interdependence and connectivity. Hierarchical Risk Parity attempts to capture this notion of hierarchy and, by extension, improve the diversification and robustness of portfolio allocation compared to other, more traditional optimization strategies.

These portfolio allocation methods have all been studied and examined in a passive setting, where returns and performance are measured in an absolute sense. For active management, this is not the case as their performance is always related to a benchmark. Additionally, different constraints on the portfolio allocation may exist which increases the level of difficulty for active fund managers. The amount of literature regarding portfolio optimization in an active management framework is rather small, especially concerning hierarchical methods. This thesis aims to expand and contribute to this area of research. It is written in collaboration with the Second Swedish National Pension Fund, which is an active management fund, and hence this thesis aims to construct and evaluate a hierarchical portfolio allocation method that is adapted to active management. In addition, this thesis attempts to develop a set of different variations of the proposed hierarchical portfolio allocation method to discover a favorable configuration of the original algorithm.

This thesis is organized as follows. In Chapter 2, definitions and theory concerning covariance matrices are provided. Further, reasons for covariance matrix instability are discussed as well as a description of active management is provided. In Chapter 3 the theory behind hierarchical clustering and its relevance for portfolio allocation is explained. Next, in Chapter 4, different portfolio allocation methods are presented. Drawing inspiration from these methods, Chapter 5 describes the hierarchical portfolio allocation algorithm proposed in this thesis. Chapter 5 also introduces the performance criteria used to evaluate the allocation algorithms, as well as a description of the data used in this thesis. The results obtained from the proposed method are presented in Chapter 6. The obtained results are then discussed and examined in Chapter 7. In chapter 8 the conclusions, remarks and future research are presented.

2

Theory

This chapter presents the theoretical frameworks relevant for this thesis. More specifically, definitions and theory regarding covariance matrices are presented, as well as a description of the active management framework outlined by the Second Swedish National Pension Fund. Finally, a discussion about the numerical instability issues that arise when estimating the covariance matrix is presented in the final section of this chapter.

2.1 Definitions

In this section, definitions of the variables used in this thesis are presented. First, let $\left\{R_i^{(t)}\right\}_{t\in T}$ be the stochastic process of the returns for asset i between time t and $t+\Delta t$ where

$$R_i^{(t)} = \frac{P_i^{(t+\Delta t)} - P_i^{(t)}}{P_i^{(t)}} \tag{2.1}$$

and $P_i^{(t)}$ denotes the price of the asset at time t. Asset prices are often modelled as cumulative sums of mean-independent increments $\epsilon_i^{(t)}$, meaning that $\mathbb{E}(\epsilon_i^{(t)}) = \mathbb{E}(\epsilon_i^{(t)}|\epsilon_i^{(t-\Delta t)} = \beta)$, for any $\beta \in \mathbb{R}$, such that

$$P_i^{(t)} = P_i^{(t-\Delta t)} + \epsilon_i^{(t)} = \cdots = \sum_{\tau=0}^{t} \epsilon_i^{(\tau)}. \tag{2.2}$$

Since the scale of the increments, $\epsilon_i^{(t)}$ grows with the level of $P_i^{(t-\Delta t)}$, $P_i^{(t)}$ is considered to be a non-stationary random variable. However, since

$$R_i^{(t)} = \frac{P_i^{(t+\Delta t)} - P_i^{(t)}}{P_i^{(t)}} = \frac{\sum_{\tau=0}^{t+\Delta t} \epsilon_i^{(\tau)} - \sum_{\tau=0}^{t} \epsilon_i^{(\tau)}}{P_i^{(t)}} = \frac{\epsilon_i^{(t+\Delta t)}}{P_i^{(t)}}, \tag{2.3}$$

the scale of $R_i^{(t)}$ does not grow with the levels of $P_i^{(t)}$ and, we assume that $\left\{R_i^{(t)}\right\}_{t\in T}$ can be approximated as a stationary stochastic process.

Furthermore, denote the standard deviation of the stochastic process $\left\{R_i^{(t)}\right\}_{t\in T}$ by σ_i and the covariance between $\left\{R_i^{(t)}\right\}_{t\in T}$ and $\left\{R_j^{(t)}\right\}_{t\in T}$ as σ_{ij}. Then, the true covariance matrix is defined as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \dots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \dots & \sigma_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \sigma_{p3} & \dots & \sigma_{pp} \end{bmatrix}. \tag{2.4}$$

Let $\rho_{i,j}$ be the correlation coefficient between the return time series $\left\{R_i^{(t)}\right\}_{t\in T}$ and $\left\{R_j^{(t)}\right\}_{t\in T}$ such that, $\rho_{i,j} = \frac{\sigma_{i,j}}{\sqrt{\sigma_i}\sqrt{\sigma_j}}$. Then, the true correlation matrix $\boldsymbol{C}$ is defined as

$$\boldsymbol{C} = \begin{bmatrix} \rho_{11} & \rho_{12} & \rho_{13} & \dots & \rho_{1p} \\ \rho_{21} & \rho_{22} & \rho_{23} & \dots & \rho_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{p1} & \rho_{p2} & \rho_{p3} & \dots & \rho_{pp} \end{bmatrix}. \tag{2.5}$$

A portfolio is defined as a p-dimensional column vector

$$\boldsymbol{w}^{(t)} := \left(w_1^{(t)}, w_2^{(t)}, \dots, w_p^{(t)}\right),^T \tag{2.6}$$

where p is the number of assets and $w_i^{(t)}$ represents the weight, or the proportion of the total capital, assigned to asset i at time t. Note that the portfolio weights must sum up to 1, i.e.

$$\sum_{i=1}^{p} w_i^{(t)} = \left(\boldsymbol{w}^{(t)}\right)^T \mathbf{1} = 1, \tag{2.7}$$

where $\mathbf{1} = (1, \dots, 1)^T$.

The portfolio return $\left(\boldsymbol{w}^{(t)}\right)^T R^{(t)}$, between time t and $t + \Delta t$ is

$$\left(\boldsymbol{w}^{(t)}\right)^T R^{(t)} = \sum_{i=1}^{p} w_i R_i^{(t)}, \tag{2.8}$$

where $R^{(t)} = \left(R_1^{(t)}, R_2^{(t)}, \dots, R_p^{(t)}\right)^T$.

Furthermore, the variance of the portfolio return is

$$\mathrm{Var}\left[\left(\boldsymbol{w}^{(t)}\right)^T R^{(t)}\right] = \sum_{i,j} \rho_{i,j}\sigma_i\sigma_j w_i^{(t)} w_j^{(t)} = \left(\boldsymbol{w}^{(t)}\right)^T \boldsymbol{\Sigma}\boldsymbol{w}^{(t)}, \tag{2.9}$$

and the standard deviation is

$$\sigma\left[\left(\boldsymbol{w}^{(t)}\right)^T R^{(t)}\right] = \sqrt{\left(\boldsymbol{w}^{(t)}\right)^T \boldsymbol{\Sigma}\boldsymbol{w}^{(t)}}. \tag{2.10}$$

Now, let $\boldsymbol{r}_i = \left(r_i^{(1)}, r_i^{(2)}, \ldots, r_i^{(n)}\right)$ be the time series of observed realisations of $\left\{R_i^{(t)}\right\}_{t\in T}$. The Pearson sample correlation, $\rho\left(\boldsymbol{r}_i, \boldsymbol{r}_j\right)$, between $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ is

$$\hat{\rho_{ij}} = \rho\left(\boldsymbol{r}_i, \boldsymbol{r}_j\right) = \frac{\sum_t (r_i^{(t)} - \bar{r_i})(r_j^{(t)} - \bar{r_j})}{\sqrt{\sum_t (r_i^{(t)} - \bar{r_i})^2 \sum_t (r_j^{(t)} - \bar{r_j})^2}}, \tag{2.11}$$

and the sample standard deviation $s(\boldsymbol{r}_i)$ of $\boldsymbol{r}_i$ is

$$s(\boldsymbol{r}_i) = \sqrt{\frac{1}{n-1}\sum_{t=1}^{n}\left(r_i^{(t)} - \bar{r_i}\right)^2}. \tag{2.12}$$

where $\bar{r_i}$ is the sample mean of $\boldsymbol{r}_i$. The sample covariance matrix $\boldsymbol{S}$ of the returns, which is an estimation of $\boldsymbol{\Sigma}$, is

$$\begin{aligned} \boldsymbol{S} &= \frac{1}{n-1}\sum_{t=1}^{n}\left(\boldsymbol{r}^{(t)} - \bar{\boldsymbol{r}}\right)\left(\boldsymbol{r}^{(t)} - \bar{\boldsymbol{r}}\right)^T \\ &= \begin{bmatrix} s_{11} & s_{12} & s_{13} & \ldots & s_{1p} \\ s_{21} & s_{22} & s_{23} & \ldots & s_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{p1} & s_{p2} & s_{p3} & \ldots & s_{pp} \end{bmatrix}, \end{aligned} \tag{2.13}$$

where $\boldsymbol{r}^{(t)} = \left(r_1^{(t)}, r_2^{(t)}, \ldots, r_p^{(t)}\right)^T$. p is the number of assets and $\bar{\boldsymbol{r}}$ is the sample mean vector defined as

$$\bar{\boldsymbol{r}} = \frac{1}{n}\sum_{t=1}^{n}\boldsymbol{r}^{(t)} = \left(\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_p\right)^T. \tag{2.14}$$

Finally, the sample correlation matrix $\hat{\boldsymbol{C}}$ of the returns, which is an estimation of $\boldsymbol{C}$, is

$$\hat{\boldsymbol{C}} = \begin{bmatrix} \hat{\rho_{11}} & \hat{\rho_{12}} & \hat{\rho_{13}} & \ldots & \hat{\rho_{1p}} \\ \hat{\rho_{21}} & \hat{\rho_{22}} & \hat{\rho_{23}} & \ldots & \hat{\rho_{2p}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{\rho_{p1}} & \hat{\rho_{p2}} & \hat{\rho_{p3}} & \ldots & \hat{\rho_{pp}} \end{bmatrix}. \tag{2.15}$$

2.2 Active Management

In active management, the objective is slightly different compared to traditional Markowitz portfolio theory. The aim is to deliver higher returns for the investors compared to a designated benchmark with a similar risk level. Hence, the MPT and the other risk-based portfolio allocation methods can not be utilized without adapting the methods to this framework.

If one defines $\boldsymbol{w}_b^{(t)} \in \mathbb{R}^p \;:\; \left(\boldsymbol{w}_b^{(t)}\right)\mathbf{1} = 0$ as the weight vector of the assets in the benchmark at time t, then the return of the benchmark between time t and $t + \Delta t$, is

$$R_b^{(t)} = (\boldsymbol{w}_b^{(t)})^T R^{(t)}. \tag{2.16}$$

Define $\boldsymbol{w}^{(t)}$ to be the weight of the active portfolio at time t, such that the associated properties outlined in the section above holds. Then, the bet vector is

$$\boldsymbol{b}^{(t)} = \boldsymbol{w}^{(t)} - \boldsymbol{w}_b^{(t)}. \tag{2.17}$$

Since $\left(\boldsymbol{w}_p^{(t)}\right)^T \mathbf{1} = 1$ and $\left(\boldsymbol{w}_b^{(t)}\right)^T \mathbf{1} = 1$, it follows that $\left(\boldsymbol{b}^{(t)}\right)^T \mathbf{1} = 0$. An important note concerning this notation is that $b_i^{(t)} < -w_{b,i}^{(t)}$ represents taking a short position in asset i at time t, something that is allowed in some funds, but not in others.

Now, let $\left\{\alpha^{(t)}\right\}_{t\in T}$ be the stochastic process of the excess return s.t.

$$\alpha^{(t)} = \left(\boldsymbol{w}^{(t)}\right)^T R^{(t)} - \left(\boldsymbol{w}_b^{(t)}\right)^T R^{(t)} = \left(\boldsymbol{b}^{(t)}\right)^T R^{(t)}. \tag{2.18}$$

Hence, $\alpha^{(t)}$ is the difference between the portfolio return and the benchmark return between time t and $t + \Delta t$. Since $\left(\boldsymbol{b}^{(t)}\right)^T \mathbf{1} = 0$,

$$\begin{aligned}
\alpha^{(t)} &= \left(\boldsymbol{b}^{(t)}\right)^T R^{(t)} \\
&= \left(\boldsymbol{b}^{(t)}\right)^T R^{(t)} - \xi \left(\boldsymbol{b}^{(t)}\right)^T \mathbf{1} \\
&= \left(\boldsymbol{b}^{(t)}\right)^T \left(R^{(t)} - \xi\mathbf{1}\right), \quad \text{for any } \xi \in \mathbb{R}.
\end{aligned} \tag{2.19}$$

Setting $\xi = \bar{r}^{(t)} = \frac{\sum_{i=1}^{p} R_i^{(t)}}{p}$, where $\bar{r}^{(t)}$ is the average return of the asset between time t and $t + \Delta t$ and $\bar{R}^{(t)} = \bar{r}^{(t)}\mathbf{1}$, one obtains

$$\alpha^{(t)} = \left(\boldsymbol{b}^{(t)}\right)^T \left(R^{(t)} - \bar{R}^{(t)}\right) \tag{2.20}$$

which shows that $\alpha^{(t)}$ is a function of each asset's relative return, $(R_i^{(t)} - \bar{r}^{(t)})$. Hence, the variance of α is

$$\mathrm{Var}\left[\alpha^{(t)}\right] = \mathrm{Var}\left[\left(\boldsymbol{b}^{(t)}\right)^T R^{(t)}\right] = \left(\boldsymbol{b}^{(t)}\right)^T \boldsymbol{\Sigma}\boldsymbol{b}^{(t)} \tag{2.21}$$

Where $R^{(t)}$ is the random vector of the returns of all assets in the portfolio and $\boldsymbol{b}^{(t)}$ is the bets chosen by the investor. $\boldsymbol{\Sigma}$ now denotes the covariance matrix with entries

$$\boldsymbol{\Sigma}_{ij} = \sigma\left(\frac{R_i^{(t)} - \bar{R}^{(t)}}{\sigma_i^{(t)}}, \frac{R_j^{(t)} - \bar{R}^{(t)}}{\sigma_j^{(t)}}\right). \tag{2.22}$$

The reasons why the covariance between $\frac{R_i^{(t)} - \bar{R}^{(t)}}{\sigma_i}$ and $\frac{R_j^{(t)} - \bar{R}^{(t)}}{\sigma_j}$ is considered are twofold. Firstly, the excess return α is a function of the relative return, and thus it is natural to subtract the average return from each individual asset return. Secondly, since $\boldsymbol{b}^T\mathbf{1} = 0$, the scale of each element in $\boldsymbol{b}$ does not matter when considering the variance of α. In this case, the covariance matrix is scale-independent as well, and thus it is more naturally to consider normalized stock prices.

To properly assess active management one may utilize the Fundamental Law of Active Management (FLAM) (Grinold, 1989), which is an approach to measure an active fund's effectiveness and quality. This assessment framework consists of four distinct factors that determine the performance of an active fund: 1) an Information criterion $\rho(\boldsymbol{b}, \boldsymbol{r})$, 2) Market turbulence expressed as the sample standard deviation of the market return, $s(\boldsymbol{r})$, 3) Length of bets, expressed as the sample standard deviation of the bets, $s(\boldsymbol{b})$ and 4) Size of portfolio p. This leads to

$$\alpha = \boldsymbol{b}^T \boldsymbol{r} = (p-1)\rho(\boldsymbol{b}, \boldsymbol{r}) s(\boldsymbol{r}) s(\boldsymbol{b}) \tag{2.23}$$

Using this assessment framework, one can construct a measure or evaluation criterion which measures how well an active fund tracks a benchmark index without replicating it. This divergence between the price behavior of an active fund and a benchmark index is denoted as Tracking Error (TE), and thus represents the deviation from a benchmark index due to active bets taken by a fund. The only factor in Equation (2.23) that is dependent of $\boldsymbol{\Sigma}$ is $\rho(\boldsymbol{b}, \boldsymbol{r})$. Since one objective of this paper is to asses how well the proposed hierarchical portfolio allocation algorithm performs, the Tracking Error (TE) is expressed as:

$$\mathrm{TE} = s\left(\left\{\rho\left(\boldsymbol{r}^{(t,t+\Delta t)}, \boldsymbol{b}^{(t)}\right)\right\}_{i=1}^{N}\right) \tag{2.24}$$

where ρ is the sample correlation between the vectors and s denotes the sample standard deviation.

2.2.1 Investment Objectives and Constraints

Firstly, the objective of the active management framework, provided by the Second Swedish National Pension Fund, is to find the optimal bets $\boldsymbol{b}^*$ which minimize the active risk, i.e. the variance of the excess return (see Equation (2.21)). Hence, one has to solve the following optimization problem at each time t,

$$\begin{aligned} \boldsymbol{b}^{*,(t)} \in \arg\min \quad & \mathrm{Var}(\alpha^{(t)}) = \mathrm{Var}\left[\left(\boldsymbol{b}^{(t)}\right)^T R\right] = \left(\boldsymbol{b}^{(t)}\right)^T \boldsymbol{\Sigma} \boldsymbol{b}^{(t)} \\ \text{s.t.} \quad & \left(\boldsymbol{b}^{(t)}\right)^T \mathbf{1} = 0 \end{aligned} \tag{2.25}$$

However, this is minimized by the trivial solution $\boldsymbol{b}^{(t)} = \mathbf{0}$, which implies that one replicates the benchmark which is not the aim of an active fund. Thus, one needs to add another constraint to Problem (2.25) to overcome the issue of replication.

Indeed, one needs to specify in the optimization problem that $\boldsymbol{w}^{(t)} \neq \boldsymbol{w}_b^{(t)}$. In this thesis, this is achieved by equaling the bet on the asset q that yielded the highest return during the previous time period to one. More specifically, including this constraint to problem (2.25) yields

$$\begin{aligned} \boldsymbol{b}^{*,(t)} \in \arg\min \quad & \left(\boldsymbol{b}^{(t)}\right)^T \hat{\boldsymbol{\Sigma}} \boldsymbol{b}^{(t)} \\ \text{s.t.} \quad & \left(\boldsymbol{b}^{(t)}\right)^T \mathbf{1} = 0 \\ & b_q^{*,(t)} = 1. \end{aligned} \tag{2.26}$$

Often, there are other constraints added to the objective that one needs to consider when investing in an active setting. These constraints might include bet sizing, lower bounds on the expected excess return, and other allocation rules that the active manager needs to take into consideration.

The investment framework considered in this thesis consists of only one such constraint, namely that it is not allowed to hedge a certain bet using an asset of dissimilar characteristics, such as industry sector, geographical location, or any other type of characteristics that can determine that two assets are too different. In other words, one is only allowed to hedge a bet using assets that shares similar characteristics or behavior. In this thesis, the considered characteristic is the covariance of the excess returns between the assets. Highly correlated assets are assumed to be similar and vice versa.

If one groups the assets in the portfolio so that similar assets are placed together in group $G \in \{1, \ldots, n_G\}$, where n_G is the number of groups. Combining this constraint with Problem (2.26) results in:

$$\begin{aligned} \boldsymbol{b}_G^{*(t)} \in \arg\min \quad & \left(\boldsymbol{b}^{(t)}\right)_G^T \hat{\boldsymbol{\Sigma}}_G \boldsymbol{b}_G^{(t)} \\ \text{s.t.} \quad & \left(\boldsymbol{b}^{(t)}\right)_G^T \mathbf{1} = 0 \\ & b_{G_p}^{*(t)} = 1, \end{aligned} \tag{2.27}$$

where $\boldsymbol{b}_G^{*(t)}$ is the vector of optimal bets for the assets in group G. Using this notation, $\boldsymbol{b}^{*(t)}$ is the concatenation of $\boldsymbol{b}_G^{*(t)}$ $\forall\, G \in \{1, \ldots, n_G\}$, such that $b_1^{*(t)}, b_2^{*(t)}, \ldots b_p^{*(t)}$ corresponds to the bets taken in asset $1, 2, \ldots, p$, respectively.

As mentioned in the previous section, the scale of $\boldsymbol{b}^*$ does not matter when evaluating the variance of the excess return. Hence, there is no need to add constraints regarding short positions since the investor may just scale the bet vector if such positions are not allowed. With the same reasoning, the investor may also scale the bet vector if the constraint $b_{G_q}^* = 1$ is not satisfactory.

2.3 Covariance Instability

Several portfolio allocation methods use the covariance matrix as an input in order to determine portfolio weights. One has to carefully separate the *true* covariance matrix $\boldsymbol{\Sigma}$ from the estimates of the covariance matrix $\hat{\boldsymbol{\Sigma}}$. Since the true covariance matrix, $\boldsymbol{\Sigma}$, is unobservable, one has to use $\hat{\boldsymbol{\Sigma}}$, which is often set to be the sample covariance matrix, $\boldsymbol{S}$. Naturally, the quality of the estimated covariance matrix, $\hat{\boldsymbol{\Sigma}}$, will have a strong impact on the quality of the portfolio weights. Bun et. al (2017) show that in the context of portfolio allocation, the predicted risk underestimates the true, realized risk due to the induced errors in the estimation of $\boldsymbol{\Sigma}$.

For a non-singular covariance matrix $\boldsymbol{\Sigma}$, the sample covariance matrix $\boldsymbol{S}$ is positive definite a.s. with rank$(\boldsymbol{S}) = p$ when the concentration ratio $c := \frac{p}{n} \in (0, 1)$, i.e when the sample size n is greater than the number of assets in the portfolio p (Alfelt, 2021). In the limit, when $n \to \infty$, $\boldsymbol{S}$ converges to the true covariance matrix $\boldsymbol{\Sigma}$. On the contrary, the sample covariance matrix $\boldsymbol{S}$ is numerically ill-conditioned when the concentration $c = \frac{p}{n} \geq 1$, with $p, n \to \infty$, i.e. in a context of high-dimensional asymptotics (Bodnar et. al., 2017). This is due to the fact that one has to estimate $\frac{p(p-1)}{2}$ elements of the true covariance matrix, using a $n \times p$ matrix. It is not possible to accurately estimate p^2 parameters from $n \cdot p$ noisy observations without some structural assumptions on the true covariance matrix $\boldsymbol{\Sigma}$.

To quantify the stability of $\hat{\boldsymbol{\Sigma}}$, or in other words, quantify how much $\hat{\boldsymbol{\Sigma}}$ can change due to a small change in the input arguments, one can use the condition number κ (Belsley et. al., 1960). The condition number κ is defined as the product of the Euclidean-norm of the estimated covariance matrix $\hat{\boldsymbol{\Sigma}}$ and its inverse, $\hat{\boldsymbol{\Sigma}}^{-1}$, or

$$\kappa(\hat{\boldsymbol{\Sigma}}) = ||\hat{\boldsymbol{\Sigma}}|| \times ||\hat{\boldsymbol{\Sigma}}^{-1}|| \tag{2.28}$$

For normal matrices, i.e. matrices that commute with their conjugate transpose, this is equivalent to the condition number defined as the absolute value of the ratio between the maximum and minimum eigenvalues of a covariance matrix, i.e.

$$\kappa = |\frac{\lambda_{max}}{\lambda_{min}}| \tag{2.29}$$

A covariance matrix is said to be ill-conditioned if the condition number is large, meaning that $\hat{\boldsymbol{\Sigma}}$ can vary drastically because of small changes in the input arguments. A covariance matrix with a small condition number is called well-conditioned.

In an ideal case, the correlation between the asset returns is zero and the correlation matrix equals the identity matrix. Hence, the condition number κ equals one and the sample covariance matrix $\boldsymbol{S}$ is well-conditioned. Outside of this ideal case correlations will not equal zero and the condition number will be larger than one. As more correlated assets are added to the portfolio, the condition number of the associated $\hat{\boldsymbol{\Sigma}}$ typically grows, and the matrix becomes more ill-conditioned. Hence, the diversification benefits that arise when more assets are added, might be erased by the estimation errors. López de Prado (2020) referees this as signal-induced instability caused by the data structure.

On the contrary to the estimation errors inferred from sampling the covariance matrix $\boldsymbol{S}$, signal-induced instability can not be remedied by increasing the number of observations n. However, the assets in a portfolio may be grouped into subsets which has a higher correlation among themselves, compared to the rest of the portfolio. Such a subset is then subject to a common eigenvector, with eigenvalues that explain a greater amount of variance. Since the main diagonal of a correlation matrix $\boldsymbol{C}$ consists of ones, the trace of $\boldsymbol{C}$ is always n. This, by extension, implies that an eigenvalue can only increase if the other eigenvalues decreases within this subset of assets in the correlation matrix, resulting in a condition number $\kappa \geq 1$. Therefore, the condition number increases when the interdependence between grouped assets increases since the ratio between the largest- and smallest eigenvalue grows. In order to extract these subsets from the correlation matrix, López de Prado (2020) suggests hierarchical clustering, further discussed in Chapter 3.

3

Graph Theory and Hierarchical Clustering

This chapter introduces hierarchical clustering and the theory supporting it. It explores and describes the relations behind hierarchical clustering, graph theory and covariance matrices and how they can be used to construct hierarchical structures of assets. More specifically, graph theory and its ability to reduce the complexity of a correlation matrix is explored in Section 3.1, dendrograms are demonstrated in Section 3.2, hierarchical clustering algorithms are described in Section 3.3. The different linkage criteria considered in this thesis are introduced in Section 3.8, and methods supplementing hierarchical clustering are described in the final subsections.

As mentioned in the introduction, Billio et. al. (2012) describes the current financial system as a dynamic, complex system where the distinctions between actors in different markets have become blurred, driven by financial innovation and deregulation. Thus, the current setting of the financial markets is characterized by greater interdependence and connectivity where financial assets correlate to varying degrees. Simon (1962) argues that the structure of the financial markets can be described as hierarchical where the hierarchical structure of interactions among elements has a distinct influence on the dynamics of the financial markets. By using hierarchical clustering, one can utilize the current structure of the financial system and create clusters of assets based on similarity. These clusters may then be used to shrink the covariance matrix to deal with the problems that arise when estimating covariance matrices in a high-dimensional setting. López de Prado (2020) highlights the signal-induced instability of the covariance matrix (see Section 2.3), as an additional reason to use hierarchical clustering.

3.1 Graph Theory

A graph in which each node is connected to all the other nodes through a weighted edge is known as a complete weighted graph. Often, the most important information of a complete weighted graph can be found in just a few of its edges. A common approach to reducing the complexity of the graph, while keeping the most important information is to create a minimum spanning tree (MST). A minimum spanning tree is a subset of the edges of a complete graph that connects all the vertices together, without creating any cycles and with the minimum possible total edge weight. The complexity of the graph is hence reduced from $\frac{n}{2}(n-1)$ to $n-1$ edges, which is illustrated in Figure 3.1.

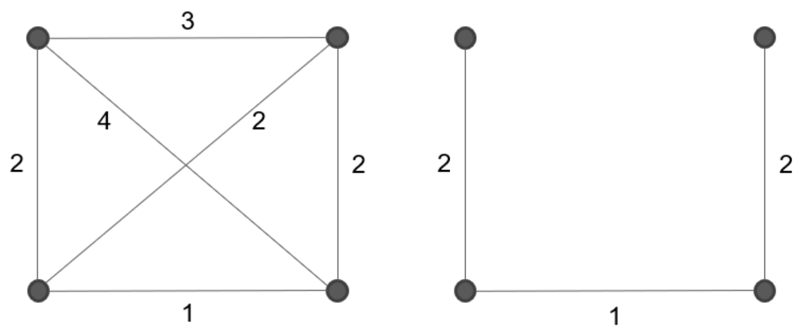


Figure 3.1: Example of a complete graph with 4 nodes and 6 edges (left) and a minimum spanning tree with 4 nodes and 3 edges (right).

The relationships between the returns of assets in a portfolio can be represented as a complete weighted graph. To achieve this, a dissimilarity measure d_{ij} between $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ $\forall\, i, j \in \{1 \dots p\}$ needs to be determined. A function $d\,:\,(\boldsymbol{r}_i, \boldsymbol{r}_j) \to [0, \infty]$ is a dissimilarity measure between $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ if the following holds:

1. $d(\boldsymbol{r}_i, \boldsymbol{r}_j) \geq 0 \quad \forall\, i, j$ and $d(\boldsymbol{r}_i, \boldsymbol{r}_j) = 0$ iff $i = j$
2. $d(\boldsymbol{r}_i, \boldsymbol{r}_j) = d(\boldsymbol{r}_j, \boldsymbol{r}_i) \quad \forall\, i, j$
3. $d(\boldsymbol{r}_i, \boldsymbol{r}_j) \leq d(\boldsymbol{r}_i, \boldsymbol{r}_k) + d(\boldsymbol{r}_k, \boldsymbol{r}_j) \quad \forall\, i, j, k \in \{1 \dots p\}$

This may be achieved in several ways, for example López de Prado (2018) suggests to transform the return matrix into a dissimilarity matrix $\boldsymbol{D}$ with entries

$$d_{ij} = d(\boldsymbol{r}_i, \boldsymbol{r}_j) = \sqrt{\frac{1}{2}(1 - \rho_{ij})} \tag{3.1}$$

where

$$\rho(\boldsymbol{r}_i, \boldsymbol{r}_j) = \frac{\sigma_{ij}}{\sqrt{\sigma_i}\sqrt{\sigma_j}} \tag{3.2}$$

is the correlation coefficient between the returns of asset i and j. The relationship among the assets in the portfolio can be represented as a complete weighted graph where the assets are the nodes and the entries d_{ij} are the weight of the edge between node i and j. Hence, the complexity of the relationship structure between the returns of the assets in a portfolio may be reduced, for example by creating a minimum spanning tree from the aforementioned complete graph.

3.2 Dendrograms

Another way to visualize relationships among data structures is through dendrograms. The advantage of dendrograms, compared to graphs discussed above, is that they can visualize hierarchical structures within the data. A dendrogram is a graph representing a binary merge tree. In the leaves of the dendrogram, all elements are considered to be single element clusters or singleton sets. As one transverses further up in the levels of the dendrogram, similar elements are merged into clusters until all elements are merged into one, single cluster at the highest level of the dendrogram. One can also transverse the tree using a top-down approach, starting with an all-encompassing cluster and creating sub-clusters by partition, ending with singleton sets. The greater the height of the branch in the dendrogram, the less similar the elements which are merged. In Figure 3.2, a dendrogram is illustrated. Evidently, the height of the branch between data points 5 and 7 is smaller than the branch between data points 0 and 1. Hence, the dissimilarity between data point 5 and 7 is smaller than the dissimilarity between data point 0 and 1.

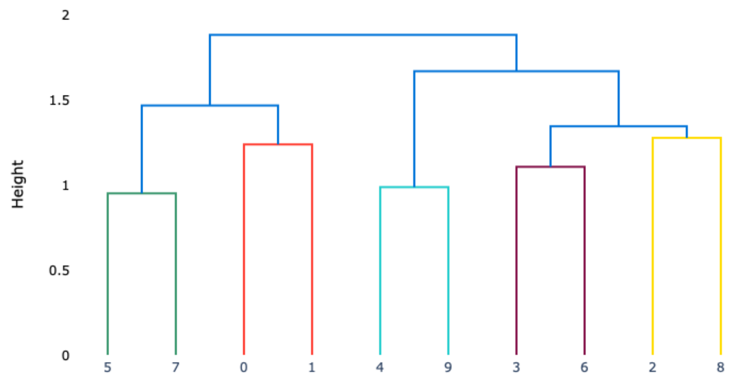


Figure 3.2: Example of a dendrogram

The hierarchical structures, represented by a dendrogram, can be obtained by applying a hierarchical clustering algorithm on a dataset, which is described in the following section.

3.3 Hierarchical Clustering Algorithms

Hierarchical clustering, which lies in the domain of unsupervised learning, is closely related to graph theory. There are two main types of hierarchical clustering algorithms, namely *agglomerative hierarchical clustering* and *divisive hierarchical clustering*. The objective of *divisive hierarchical clustering* is to maximize the inter-cluster dissimilarity. Initially, in divisive hierarchical clustering, all data points in the dataset belong to one large single cluster. The algorithm then partitions these clusters into sub-clusters in order to construct a binary merge tree, where the partition is performed recursively until all the data points are in single element clusters (Hastie et. al., 2001).

In contrast to divisive hierarchical clustering, the objectives of *agglomerative hierarchical clustering* is to minimize the inter-cluster dissimilarity. Initially, in agglomerative hierarchical clustering, all data points are considered as single element clusters or singleton sets. Then, clusters are merged, forming larger clusters at higher levels in the binary merge tree. This process is then performed iteratively until all clusters have been merged into one large cluster that contains all observations (Hastie et. al., 2001).

Even though the objectives and the method differ between agglomerative- and divisive- hierarchical clustering, both results in the same dendrogram if the same dissimilarity measures is used. Hastie et. al., (2001) concludes that agglomerative clustering is the most efficient of the two. Hence, this method will be the one further discussed and used in this thesis.

The key in cluster analysis is the dissimilarity measure. The criteria of which clusters to merge in each step of the merging process is determined by the intercluster dissimilarity between clusters. The intercluster dissimilarity can be determined using several different measures, referred to as linkage criteria. Some commonly used linkage criteria are presented in the following sections.

3.4 Single Linkage

In Single Linkage, the distance between two clusters is the minimum distance between members of the two clusters:

$$d_{SL}(A, B) = \min_{i \in A, i' \in B} d_{ii'}. \tag{3.3}$$

The Single Linkage method creates a minimum spanning tree between the clusters. Hence, in line with the arguments in Section 3.1, it is likely that this method selects the most relevant connections. However, a problem is that if the dissimilarity between one element in each cluster is small, the clusters are considered close. This can lead to a violation of the compactness property, which states that all observations within each cluster should be more similar to each other than to elements in other clusters and hence result in chaining (Raffinot, 2018). This effect is illustrated in Figure 3.3a.

3.5 Complete Linkage

In Complete Linkage, the distance between two clusters are defined as the maximum distance between two members of each cluster:

$$d_{CL}(A, B) = \max_{i \in A, i' \in B} d_{ii'}. \tag{3.4}$$

Thus, the distance between two clusters are determined by the distance between the two furthest members in the two clusters. In contrast to Single Linkage, two clusters are considered to be close if all observations in both clusters are relatively alike. As a result, Complete Linkage is sensitive to outliers but tends to form dense, similar-sized clusters. However, it can also create clusters with members which are closer to other clusters than to certain members of the same cluster (Hastie et al., 2001). A dendrogram using Complete Linkage is illustrated in Figure 3.3b, where it can be viewed that the structure is different from Single Linkage.

3.6 Group Average Linkage

In Group Average Linkage, the distance between two clusters, A and B, is defined as the average of all distances between members of the two clusters:

$$d_{GAL}(A,B) = \frac{1}{N_A N_B} \sum_{i \in A} \sum_{i' \in B} d_{ii'}, \tag{3.5}$$

where N_A, N_B is the number of observations in the clusters respectively. Group Average Linkage is a compromise between Single Linkage and Complete Linkage. The idea of Group Average Linkage is to construct clusters that on average are close together internally but far apart from other clusters (Hastie et al., 2001). The hierarchy given by Group Average Linkage is illustrated in Figure 3.3c.

3.7 Ward's Method

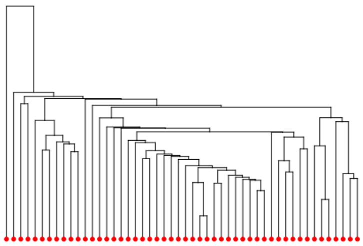
In Ward's Method (Ward, 1963) the distance is defined as the increase in the squared error that would occur if clusters A and B were merged.

$$d_{WM}(A,B) = \frac{N_A N_B}{N_A + N_B} \left\| \mu_A - \mu_B \right\|^2, \tag{3.6}$$

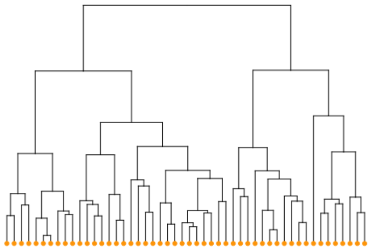
where μ_A and μ_B are the mean vectors of each cluster. A prevalent issue with Ward's Method is that it often creates larger clusters. On the contrary, it is more robust to noise and outliers than to the other linkage criteria (Raffinot, 2018). Figure 3.3d illustrates the created hierarchical structure obtained when Ward's method is used linkage criterion.

3.8 Linkage Criteria Discussion

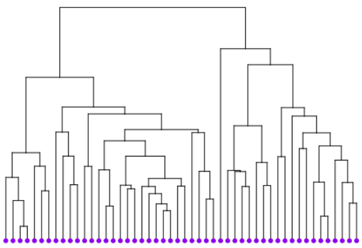
The different linkage criteria affect the dendrogram differently and result in different tree structures since the linkage criteria define the proximity of clusters. To mathematically establish the superiority of a linkage criteria over another is not feasible (Jain and Dubes, 1988). Hence, the user needs to inspect empirical experiments to see how the different linkage criteria construct the hierarchical structure. For example, Papenbrock (2011) argues that the Single Linkage algorithm creates both large and small clusters that are chained together, which is illustrated in Figure 3.3a. Thus, Single Linkage preserves the original structure as much as possible if one uses the perspective that elements that depart early in the structure can be viewed as different. This can lead to clusters that consist of outliers and are different from the rest of the elements. On the contrary, Complete Linkage creates more balanced and tight cluster where similar objects are grouped, see Figure 3.3b. Papenbrock (2011) argues that Group Average Linkage is a trade-off between Single- and Complete Linkage and that Ward's Method is capable of constructing very distinct clusters with clear separation. Figure 3.3 illustrates the hierarchical structures produced by the different linkage criteria. One can see that Complete Linkage and Ward's Method produce more symmetrical tree structures compared to Group Average- and Single Linkage.



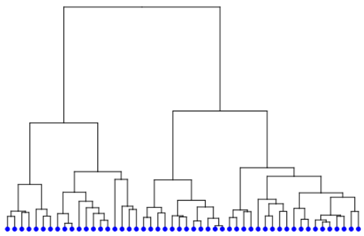
(a) Single Linkage



(b) Complete Linkage



(c) Group Average Linkage



(d) Ward's Method

Figure 3.3: Example of dendrograms using different linkage criteria.

3.9 Selection of Number of Clusters

Hierarchical clustering subsets input data into clusters based on pairwise dissimilarities between observations, but does not find the optimal number of clusters. As opposed to other clustering methods, such as k-means (Lloyd, 1982), the number of clusters, k, is not needed as an input parameter for hierarchical methods. Indeed, hierarchical clustering methods find a cluster structure from 1 to p clusters among p assets which may induce potential overfitting. By randomly choosing the number of clusters k, one might end up with clusters that do not reflect reality very well (Raffinot, 2018). Hence, for correlation clustering problems, obtaining a suitable number of clusters k^* is an essential part of the problem. Cluster analysis is used to provide statistics for assigning an unknown number of natural clusters, defined as k^*. Usually, data-based methods examine the within-cluster dissimilarity W_k, as a function of the number of clusters $k \in \{1, 2, \dots p\}$

Hastie et. al. (2001) proposed the Gap Statistic Index (GSI), which now is a commonly used method to estimate k^*. The main idea of GSI is to compare the logarithm of the within-cluster dissimilarity, $\log[W_k]$, with uniformly distributed data with no apparent clusters. The within-cluster dissimilarity W_k is defined as the sum of the pairwise distances C_q for all points in a cluster q,

$$W_k = \sum_{q=1}^{k} \frac{1}{2n_q} D_q, \tag{3.7}$$

where $D_c = \sum_{i,i' \in C_q} d_{ii'}$, $d_{ii'}$ is the Euclidean distance between node i and i' and n_q is the number of points in C_q.

The idea behind GSI is to standardize the graph of $\log(W_k)$ by comparing it with the expectation under a null reference distribution of the data. Tibshirani et. al. (2001) stress the importance of an appropriate null model. The goal of the estimate is to provide the most information possible about a dataset compared to a reference distribution with no hierarchical structures or clusters, which corresponds to the value of k for which $\log[W_k]$ falls the farthest below this reference curve. Hence the measure is defined as follows:

$$Gap_n(k) = \mathbb{E}_n^*\{\log[W_k]\} - \log[W_k], \tag{3.8}$$

where $\mathbb{E}_n^*$ denotes the expectation under a sample of size n from the reference distribution.

To account for the dispersion of W_k, the optimization problem is adjusted using the standard deviation to account for the varying volatility of estimates, calculated as:

$$s_k = \sqrt{1 + (1/B)}\sigma(k), \tag{3.9}$$

where B is the number of performed simulations of the reference distribution to obtain $\sigma(k)$. The estimated number of clusters $\hat{k}$ by GSI can then be computed as follows,

$$\begin{aligned} \min_k \quad & Gap(k) \\ \text{s.t.} \quad & Gap(k) \geq Gap(k+1) + s_{k+1}. \end{aligned} \tag{3.10}$$

To find reliable estimations of the distribution one needs a large number of draws from the null distribution. Hence, there is a trade-off between computation time and accuracy.

Yue, Wang, and Wei (2008) proposed an alternative approach to compute the GSI in order to obtain the optimal number of clusters k^*. The authors argue that this method is not as computational expensive and provides more stable results compared to the previously presented method. Yue, Wang, and Wei (2008) substitute the two-order difference formation for the objective function in the original Gap Statistic Index. Hence, the new index for clustering optimality is characterized as,

$$\begin{aligned} k^* = \arg\max \quad & W_k - 2W_{k+1} + W_{k+2} \\ \text{s.t.} \quad & 0 \leq k \leq \sqrt{n}. \end{aligned} \tag{3.11}$$

3.10 Optimal Leaf Ordering

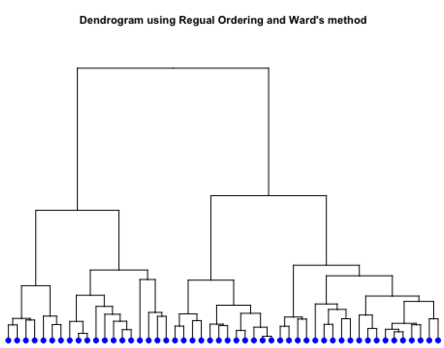
When performing hierarchical clustering on a set of data, the order of the leaves matter for the optimality of the clustering. Since there are 2^{p-1} orderings of a hierarchical tree with p leaves, hierarchical clustering algorithms depend on heuristic approaches based on local similarities or self-organizing maps to determine the global leaf ordering. Utilizing heuristic approaches may produce non-optimal leaf orderings, which in turn can yield sub-optimal results (Bar-Joseph et al. 2001). Bar-Joseph et al. (2001) present a leaf ordering algorithm, that maximizes the sum of similarities of adjacent elements in the leaf ordering. More specifically, the objective of the algorithm is to find a linear ordering amongst 2^{p-1} possible orderings that are optimal, and can be expressed as

$$D^{\phi}(T) = \sum_{i=1}^{p-1} \boldsymbol{S}'(z_{\phi_i}, z_{\phi_{i+1}}), \tag{3.12}$$

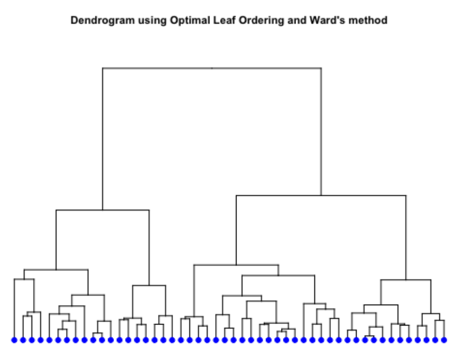
where $\boldsymbol{S}'$ is the similarity matrix, which is the opposite concept to the dissimilarity matrix discussed in Section 3.1, and where z_{ϕ_i} is the i^{th} leaf when a binary tree T is ordered according to ϕ. The objective of the optimization is to obtain the ordering or arrangement of arguments ϕ which maximizes $D^{\phi}(T)$, such that

$$\phi^* = \arg\max D^{\phi}(T), \tag{3.13}$$

The algorithm developed by Bar-Joseph et al. (2001) shows superior performance compared to hierarchical clustering approaches that depend on heuristics on a random-, artificial- and biological dataset. This in extension enables the user to establish meaningful cluster boundaries as well as identify the relationships between clusters in a superior manner. As can be seen in Figure 3.4, this results in more symmetrical structures.



(a) Random Leaf Ordering



(b) Optimal Leaf Ordering

Figure 3.4: Example of dendrograms using Ward's Method with Random Leaf Ordering (left) and Optimal Leaf Ordering (right).

3.11 Hierarchical Correlation Clustering

Since the interest of this thesis lies in clustering assets in order to shrink the covariance matrix, the return data needs to transformed into a distance matrix $\boldsymbol{D}$ with entries d_{ij}, as explored in Section 3.1. To obtain a satisfactory distance metric, the Euclidean distance between the pairwise column-vectors of the distance matrix $\boldsymbol{D}$ is then computed (López de Prado, 2016) as

$$\tilde{d}_{ij} = \tilde{d}[\boldsymbol{D}_i, \boldsymbol{D}_j] = \sqrt{\sum_{m=1}^{p} (d_{mi} - d_{mj})^2}. \tag{3.14}$$

The difference between d_{ij} and $\tilde{d}_{ij}$ is small. While d_{ij} measures the distances between the column vectors of the sample correlation matrix $\boldsymbol{C}$, $\tilde{d}_{ij}$ measures the distances between the column vectors of the $\boldsymbol{D}$, which in turn yields a distance of distances (López de Prado, 2016). This final distance matrix, $\tilde{\boldsymbol{D}}$, is then used to find correlation clustering within the asset portfolio through hierarchical clustering algorithms.

4
Portfolio Allocation Methods

This section presents different allocation methods stemming from Markowitz's Modern Portfolio Theory, and covers established risk-based methods as well as more novel techniques such as hierarchical allocation methods. The presented methods represent a source of knowledge used for developing the hierarchical portfolio allocation algorithm presented in the following chapter.

4.1 Equally-Weighted Portfolio

The Equally-Weighted portfolio is the naive approach to allocate capital evenly amongst assets within a portfolio. Hence, given p assets, the portfolio weights $\boldsymbol{w} = (w_1, \ldots, w_p)$ are given by

$$w_i = \frac{1}{p} \quad \forall i = 1, \ldots, p. \tag{4.1}$$

This simplistic approach has surprisingly performed well during different market regimes and has often out-performed more sophisticated asset allocation strategies (DeMiguel et. al, 2009; Ernst et. al., 2016; Kolm et. al, 2014). DeMiguel et. al (2009) highlight the estimation risk inherent in optimization portfolios as a cause for their inferiority compared to the naive approach of distributing capital evenly amongst assets. The need for estimation of returns and covariances, a necessity in several optimization methods, induce bias and error which results in portfolio weights that can be far from optimal.

4.2 Modern Portfolio Theory and Minimum Variance

Markowitz's (1952) work on Modern Portfolio Theory (MPT) revolutionized the field of portfolio construction, as it provided investors with a method to allocate capital amongst a set of assets efficiently. More specifically, Modern Portfolio Theory is a method for constructing portfolios with the objective to minimize the risk in the portfolio for a given level of expected returns, or vice versa (Markowitz, 1952). If an investor wants to invest in p risky assets with an expected return vector $\boldsymbol{\mu}$, and covariance matrix $\boldsymbol{\Sigma}$, using Markowitz's theory, the investor aims to obtain a $p \times 1$ weight vector $\boldsymbol{w}$. The expected return of the portfolio is $\boldsymbol{w^T \mu}$ and the variance of the portfolio is $\boldsymbol{w^T}\boldsymbol{\Sigma}\mathbf{w}$. Hence, if investors want to construct minimum-variance portfolios with a lower bound on the expected return β, the following quadratic optimization needs to be solved

$$\begin{aligned} \min \quad & \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w} \\ \text{s.t.} \quad & \boldsymbol{w}^T \boldsymbol{1_p} = 1 \\ & \boldsymbol{\mu}^T \boldsymbol{w} \geq \beta. \end{aligned} \tag{4.2}$$

The Efficient Frontier, illustrated in Figure 4.1, is the set of optimal portfolios that yield the lowest risk given a level of expected return, and vice versa. Typically, the risk is calculated using the standard deviation of the portfolio returns.

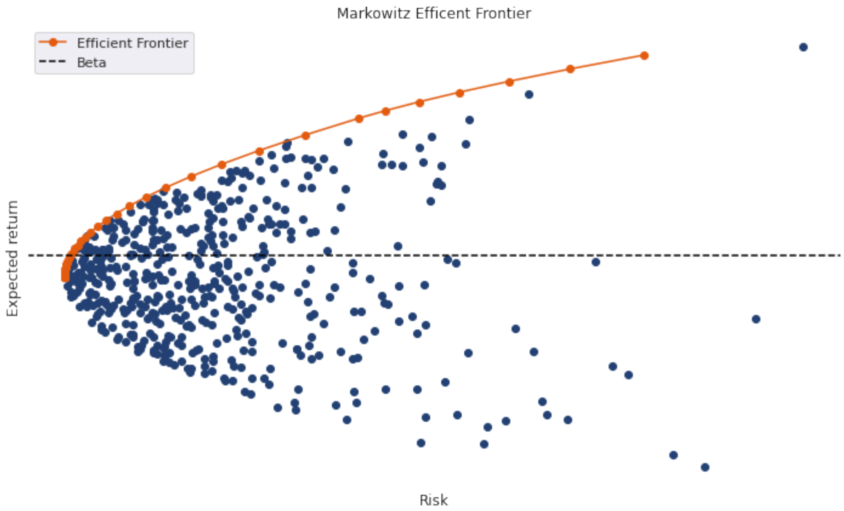


Figure 4.1: Example of Markowitz Efficient Frontier (red) using simulated data with four assets and 1000 observations. The dashed black line represents the lower bound on expected return β.

However, Markowitz's efficient solution to the problem of capital allocation has faced criticism when implemented in practice. Michaud (1989) highlights the inherent estimation risk within MPT and argues that the estimation procedure of the expected returns and the covariance matrix induces bias, which in turn leads to significant underestimation of an optimal portfolio's true level of risk. This paper also highlights the requirement of inversion of the estimated covariance matrix, which may not be possible to obtain in a high-dimensional setting. Thus, small changes in input parameters can lead to considerable changes in optimal solutions, due to the ill-conditioning and instability of the covariance matrix. Perrin and Roncalli (2020) emphasise that in order to generate acceptable solutions for the minimum-variance problem, appropriate weight constraints must be established. Hence, minimum-variance optimization is inherently a trial-and-error process, and not a systematic approach.

4.3 Risk Parity

In contrast to Markowitz's Modern Portfolio Theory, Risk Parity (RP) is an investment framework that has the objective of creating portfolios where each asset class contributes equally to the overall risk of the portfolio in order to obtain diversification (Qian, 2005a). Inherently, Risk Parity allocates risk instead of capital in a heuristic sense, in comparison to other traditional cross-asset portfolios such as market portfolios (where the weights are determined by market capitalization) or 60/40 portfolios of equities/bonds, in order to construct well-diversified portfolios. For instance, since equities are naturally riskier than bonds or other fixed-income products, the traditional 60/40 portfolio is not diversified from a Risk Parity perspective since equities contribute to an disproportional amount of risk to the overall portfolio compared to bonds, even though the capital between the two asset classes have been distributed rather evenly. Qian (2005b) stresses the importance of risk contribution and illustrates empirically that risk contribution approximates the expected loss contribution from the underlying components of the portfolio. Risk Parity has historically outperformed and yielded investors better returns than traditional portfolios such as the market- and 60/40 portfolio (Asness et. al., 2012).

Diversifying risk instead of capital amongst asset classes within a portfolio is the fundamental idea of Risk Parity and is in clear contrast to Markowitz and other type of portfolios. Extensions of the idea of Risk Parity is Equal Risk Contribution portfolios, which are described in more detail in the following section.

4.4 Equal Risk Contribution

Maillard et. al. (2010) present a different approach to the capital allocation problem which is an extension of the idea of Risk Parity (see Section 4.3). Every asset in a portfolio contributes with a varying level of risk to the overall portfolio, depending on the risk characteristics of the asset. Hence, the risk contribution from an asset i is the proportion of the overall portfolio risk related to that component. To obtain the risk contribution of asset i, one needs to calculate its marginal risk contribution which is defined as the change of the total risk of the portfolio as a result of an infinitesimal increase in weight of the asset i. More specifically, given a portfolio of p assets, the marginal risk contribution MRC_i, of asset i is defined such that

$$MRC_i = \frac{\partial \sigma(\boldsymbol{w})}{\partial w_i} \tag{4.3}$$

where $\sigma(\boldsymbol{w}) = \sqrt{\boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}}$ represents the risk of the portfolio. Furthermore, the risk contribution RC_i, of asset i is defined as the product of its marginal risk contribution and weight in the portfolio, i.e.

$$RC_i = w_i \cdot MRC_i = w_i \frac{\partial \sigma(\boldsymbol{w})}{\partial w_i} = w_i \frac{(\boldsymbol{\Sigma w})_i}{\sqrt{\boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}}} \tag{4.4}$$

Maillard et. al. (2010) shows that the risk contributions are additive by Euler's theorem for homogeneous functions, and thus is the total risk of the portfolio the sum of the risk contributions from its different assets, namely $\sigma(\boldsymbol{w}) = \sum_{i=1}^{p} RC_i$.

The Equal Risk Contribution (ERC) portfolio is the portfolio where the risk contributions from the different assets are equalized. Naturally, different assets contribute with varying degrees of risk to the overall portfolio, which is adjusted by the allocation of capital to the different assets so that the risk contribution amongst the assets in the portfolio is equal. Hence, the ERC portfolio, where short positions are prohibited, is defined as

$$\boldsymbol{w} = \{w_i \geq 0 : \boldsymbol{w}^T \mathbf{1} = 1, w_i(\boldsymbol{\Sigma w})_i = w_j(\boldsymbol{\Sigma w})_j \quad \forall\, i, j = 1, \ldots, p\}. \tag{4.5}$$

4.5 Hierarchical Risk Parity

López de Prado (2018) uses hierarchical clustering to develop a novel approach for portfolio optimization that is not dependent on the inversion of the covariance matrix, named Hierarchical Risk Parity (HRP). According to this paper, the method decreases the instability and the portfolio weight concentration prevalent in other portfolio optimization methods. As mentioned before, hierarchical clustering methods such as HPR can construct efficiently allocated portfolios on ill-conditioned and even singular covariance matrices since the methods do not rely on the inversion of the covariance matrices. HRP uses the information in the covariance matrix without requiring it to be invertible or positive-definite. The algorithm consists of three stages: *hierarchical clustering*, *quasi-diagonalization*, and *naive recursive bisection*, which are described in the following sections.

4.5.1 Hierarchical Clustering

The objective in this primary stage of the HRP method is to cluster p assets into a hierarchical structure using agglomerative clustering so that the asset allocations can flow downstream through a tree. López de Prado (2018), applies the Single Linkage criterion, which as previously mentioned, produces the same result as that of the Minimum Spanning Tree.

The original method suggested by López de Prado (2018) is similar to the general clustering algorithm detailed in Section 3.3. First, compute a $p \times p$ correlation matrix $\boldsymbol{C}$ with entries $\rho = \{\rho_{i,j}\}_{i,j=1,\ldots,p}$ where $\rho_{ij} = \rho[r_i, r_j]$ is the Pearson correlation coefficient between the return time series of asset i and j. Next, the distance measure is defined as:

$$d_{i,j} = d(r_i, r_j) = \sqrt{\frac{1}{2}(1 - \rho_{i,j})} \tag{4.6}$$

López de Prado (2018) verifies that Equation (4.6) is a dissimilarity measure, where perfectly correlated assets with $\rho_{i,j} = 1$ has a distance $d_{i,j} = 0$ and conversely, perfectly negatively correlated assets with $\rho_{i,j} = -1$ has a distance $d_{i,j} = 1$. Next, the distance matrix $\boldsymbol{D} = \{d_{i,j}\}_{i,j=1,\ldots,p}$ is computed.

Then, the Euclidean distance between any two column vectors of $\boldsymbol{D}$ is computed as

$$\tilde{d}_{i,j} = \tilde{d}[D_i, D_j] = \sqrt{\sum_{n=1}^{N} (d_{n,i} + d_{n,j})^2} \tag{4.7}$$

Hence, $\tilde{d}_{i,j}$ is a distance of distances. Next step is to cluster the pair columns (i^*, j^*) with Single Linkage, such that $(i^*, j^*) = \arg\min(i, j)_{i \neq j}\{\tilde{d}_{i,j}\}$. Then, the distance between the newly formed cluster and the remaining clusters is defined by the Single Linkage criteria, discussed in Section 3.3. These steps are then executed recursively until there is only one cluster that contains all assets. The produced hierarchical structure is the output from this primary stage of the HRP method.

4.5.2 Matrix Reordering

The next step of the HRP method reorders the rows and columns of the covariance matrix to obtain a quasi-diagonal covariance matrix where assets with high correlation are placed adjacently and close to the matrix diagonal, while assets with low correlation are placed apart. Figure 4.2 illustrates the effect of quasi-diagonalization of a covariance matrix using Single Linkage. The quasi-diagonalization of the covariance matrix reflects the hierarchical structure that was found in the hierarchical clustering step of the method and is shown in Figure 4.2. López de Prado (2018) proves that the inverse-variance allocation, which is used in the next stage of the HRP method, is in fact optimal when the covariance matrix is diagonal.

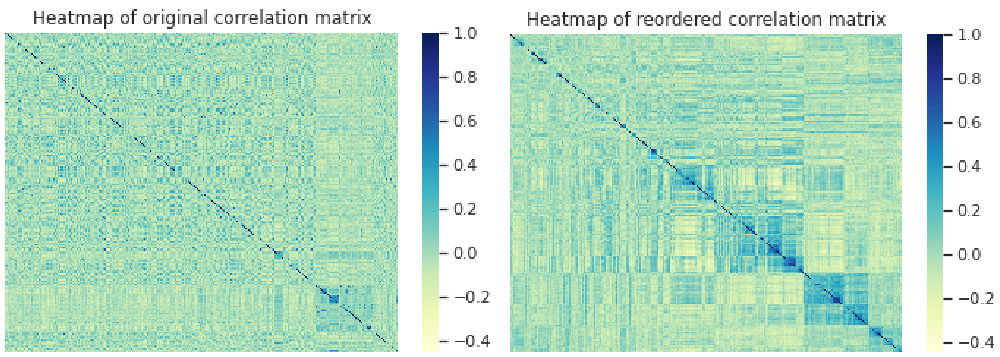


Figure 4.2: *Example of reordered correlation matrix using Single Linkage*

4.5.3 Naive Recursive Bisection

The previous step of the HRP method yielded a quasi-diagonal matrix, which is used in this final stage to allocate weights to the assets in the portfolio. The asset weights are calculated recursively by bisecting the covariance matrix into subsets of equal size until there are only singletons sets left, and where the weight of each asset is determined by inverse-variance allocation. In detail, this stage of the HRP algorithm can be described as follows:

Initialize a list of clusters of assets in the portfolio, denoted $L = \{L_0\}$ with $L_0 = \{i\}_{i=1,\dots,p}$ and initialize a between-cluster weight vector of unit weights:

$$w_i = 1 \quad \forall\, i \in [1, \dots, p]. \tag{4.8}$$

Then, for each cluster $L_i \in L$ with more than one asset (non-singleton set), bisect L_i into two equally-sized subsets $L_i^{(1)}$ and $L_i^{(2)}$ such that $L_i = L_i^{(1)} \cup L_i^{(2)}$. The bisection of L_i preserves the order.

The next step is to calculate the variance of each bisection $L_i^{(j)}$, $j = 1, 2$ as

$$\mathrm{Var}_i^{(j)} = \tilde{w}_i^{(j)T} \mathbf{\Sigma}_i^{(j)} \tilde{w}_i^{(j)} \quad j = 1, 2 \quad , \tag{4.9}$$

where $\mathbf{\Sigma}_i^{(j)}$ is the covariance matrix of the assets in cluster j and $\tilde{w}_i^{(j)}$ is defined such as

$$\tilde{w}_i^{(j)} = \frac{\mathrm{diag}[\mathbf{\Sigma}_i^{(j)}]^{-1}}{\mathrm{Tr}[\mathrm{diag}[\mathbf{\Sigma}_i^{(j)}]^{-1}]} \quad j = 1, 2 \quad , \tag{4.10}$$

where diag[$\cdot$] and Tr[$\cdot$] is the diag- and trace operator, respectively.

The weight allocation to clusters is computed using a split factor θ_i which is defined as the inverse-variance allocation between clusters, i.e.

$$\theta_i = 1 - \frac{\mathrm{Var}_i^{(j)}}{\sum_j \mathrm{Var}_i^{(j)}} \quad j = 1, 2 \quad . \tag{4.11}$$

Finally, re-scale and update the between-cluster weights for the sub-clusters such as:

$$\begin{aligned} w_i &:= \theta_i \times w_i \quad \forall i \in L_i^{(1)} \\ w_i &:= (1 - \theta_i) \times w_i \quad \forall i \in L_i^{(2)} \quad . \end{aligned} \tag{4.12}$$

This process iterates until there are only singleton sets left, i.e. when each asset constitutes a cluster. Hence, this stage of the HRP method uses a top-down approach to assign cluster weights.

Moreover, López de Prado (2018) allocates the HRP weights in a naive manner since the algorithm does not incorporate the hierarchical structure from the primary stage. Since the quasi-diagonal matrix provides the ordering of the assets, which is stored in a list and then bisected into subsets of equal size resulting in a binary tree that is distinctly different from the hierarchical structure produced in stage 1, which consecutively leads to distinctly different clusters as well. This entails that the order of assets in the input data is highly important since the naive recursive bisection stage of the HRP algorithm creates clusters, and thus cluster weights, that are dependent of the order on assets. In a portfolio optimization context, investors would desire an allocation method that is independent of the ordering structure of the input data.

4.6 Hierarchical Clustering-Based Asset Allocation

Raffinot (2017) builds on López de Prado's (2018) notion of Hierarchical Risk Parity and proposes a method referred to as Hierarchical Clustering-Based Asset Allocation

(HCAA) that attempts to mitigate some of the issues that are prevalent in the HRP method. HCAA consists of four major stages, namely:

(I) Hierarchical tree clustering

(II) Determining optimal number of clusters

(III) Allocation of capital across clusters

(IV) Allocation of capital within clusters

Using the HCAA algorithm, Raffinot (2017) conducted experiments on several datasets of varying characteristics, such as multi-asset- and individual stock datasets as well as on a S&P 500 sector dataset to assess the performance of HCAA compared to more traditional portfolio optimization methods. The out-of-sample performance of the hierarchical clustering portfolios shows that they are able to be well-diversified while achieving statistically superior risk-adjusted returns compared to traditional risk-based allocation strategies.

4.6.1 Hierarchical Tree Clustering

The hierarchical tree clustering stage in HCAA follows the initial stage of HRP and the theory described in Section 3.3, where the hierarchical structure is built using an agglomerative approach. However, a different distance measure is used to encode the correlation matrix to a distance matrix $\boldsymbol{D}$ with entries $d_{i,j}$ (Mantegna, 1999),

$$d_{i,j} = \sqrt{2(1 - \rho_{i,j})}. \tag{4.13}$$

Also, Raffinot (2017) considers a wider set of linkage criteria in the hierarchical clustering of the assets compared to HRP, namely Ward's method as well as Single, Group Average, and Complete Linkage.

4.6.2 Determining the Optimal Number of Clusters

The second stage of HCAA consists of finding the optimal number of clusters, which is achieved using Gap Statistic Index, described in Section 3.9. The use of Gap Statistic Index to find the optimal number of clusters reduces the computation time of the algorithm as the full tree does not need to be constructed. Additionally, prohibiting the tree from growing to maximum depth can mitigate overfitting issues that can lead to estimation errors in portfolio weights.

4.6.3 Allocation of Capital Across and Within Clusters

Steps three and four of HCAA incorporate the same reasoning regarding capital allocation, namely equal weighting of capital between clusters as well as between assets within clusters. Raffinot (2017) focuses on capital allocation simplicity and efficiency, so that a large number of correlated assets receive the same overall allocation as a single uncorrelated asset. This contrasts with López de Prado's (2018) original algorithm which instead uses inverse-variance allocation.

4.7 Hierarchical Equal-Risk Contribution

Raffinot (2018) developed yet another hierarchical clustering allocation method, referred to as Hierarchical Equal Risk Contribution Portfolio (HERC). HERC aims to combine the best of HRP and HCAA, respectively, resulting in a portfolio allocation method with the objective to diversify capital- and risk allocation. HERC follows the initial steps of HCAA and HRP, however, it is in the latter stages of HERC that diverges from the other hierarchical methods. HERC uses a top-down recursive division approach for allocating assets, similarly to HRP's recursive bisection step, but the method employs equal risk contribution (ERC) to calculate the scaling factor θ_i. This enables the user to implement a larger scope of risk metrics, such as conditional drawdown at risk (CDaR) and conditional value at risk (CVaR), to calculate the allocation of portfolio weights.

Using Ward's Method as linkage criteria for the agglomerative clustering in the HERC method, Raffinot (2018) conducted comparison studies using HERC, HCAA, and HRP on two separate empirical datasets consisting of asset classes and individual stocks, respectively. The conclusion was that HERC and HCAA are very similar in their out-of-sample performance and generated comparable risk-adjusted returns. There was also a difference in the out-of-sample performance of the HERC method for different risk measures, were using conditional drawdown at risk as a risk measure yielded the largest outperformance. Finally, both HERC and HCAA were able to deliver statistically higher risk-adjusted returns and better diversification compared to HRP on the primary dataset consisting of asset classes. This was not true for the second dataset consisting of individual stocks where HRP outperformed the HCAA model but unperformed the HERC model.

We now provide more details about HERC. It can be summarized in four distinct stages:

(I) Hierarchical tree clustering

(II) Determining the optimal number of clusters

(III) Top-down recursive bisection

(IV) Naive risk parity within the clusters

Stages one and two are identical to the two first stages of HCAA (see Section 4.6.1 and Section 4.6.2), where agglomerative clustering is used to construct a binary tree with an optimal number of clusters determined by Gap Statistics Index using Ward's Method as linkage criteria. The top-down recursive bisection and the naive risk parity within clusters stages of the HERC algorithm are described in the following subsections, respectively.

4.7.1 Hierarchical Recursive Bisection

Similarly to HRP, the clustering of assets is used to determine the allocation of weights using recursive bisection. Nonetheless, as previously detailed, HRP does not utilize the hierarchical structure of the produced binary tree - only the ordering of assets. HERC captures on the contrary this hierarchical structure by bisecting clusters into two sub-clusters at each node by traversing the binary tree using a top-down approach. Note that HERC does not require an equal size of the sub-clusters, which is not true for HRP. The bisection process terminates when the optimal number of clusters k^*, given by Gap Statistic Index in the preceding stage, is reached. The scaling factor θ_i is determined by using ERC portfolio allocation, in particular

$$\theta_i = 1 - \frac{\mathrm{RC}_i^{(j)}}{\sum_j \mathrm{RC}_i^{(j)}} \quad j = 1, 2 \tag{4.14}$$

where RC_i is the risk contribution from each cluster. Here, one can consider a range of different risk measures to calculate RC_i. For instance, one can calculate cluster variance using (4.10). Consequently, one can then calculate the cluster weights as

$$\begin{aligned} w_n &\coloneqq \theta_i \times w_n \\ w_n &\coloneqq (1 - \theta_i) \times w_n \end{aligned} \tag{4.15}$$

4.7.2 Within-Cluster Weight Allocation

Furthermore, the last and final step of HERC consists of computing the asset weights within a cluster. Consider the set of assets $X \coloneqq \{x_1, x_2, \ldots, x_n\} \in C_i$, where $C_i \in [1, \ldots, k^*]$ is an arbitrary cluster of n assets. Then, using inverse-risk allocation to determine the weights for X in C_i one obtains the *naive risk parity weights*:

$$w_{NRP}^{C_i} = \frac{\frac{1}{RC_i^{C_i}}}{\sum_{k=1}^{n} \frac{1}{RC_k}} \quad \forall i \in [1, \ldots, k^*] \tag{4.16}$$

Where the variance of cluster i is the most common risk metric RC_i. Using w_{NRP}^{C} for a cluster C_i, one can compute the asset weight in cluster C_i by taking the product of the cluster weights and the naive risk parity weights, i.e.

$$w_{HERC} = w_n \times w_{NRP}^{C_i} \quad \forall\, C_i \tag{4.17}$$

5

Method

The methodology used in this thesis is constituted by three main stages: 1) Literature study, 2) Development of hierarchical portfolio allocation algorithms and 3) Evaluation using empirical data. In the primary stage, a study of previous research in the field of portfolio allocation and hierarchical clustering was conducted. The study primarily focused on the works of Markowitz (1952), López de Prado (2018), and Raffinot (2017; 2018), as well as on some supplementary research regarding hierarchical clustering. This primary stage of the methodology is detailed in Chapter 2. Secondly, based on the conducted literature study, a hierarchical portfolio allocation algorithm (HPAA) was developed in order to adapt the methods of López de Prado (2018) and Raffinot (2017; 2018) to an active management framework. Several different constellations of the algorithm were constructed for comparison purposes. Thirdly, all of the developed allocation algorithms were tested and evaluated using empirical data. All numerical calculations have been performed using Python. Finally, the latter two stages are explained in the following sections in this chapter.

5.1 Description of Evaluated Portfolio Allocation Algorithms

The main objective of the proposed hierarchical portfolio allocation algorithm is to place a set of bets on certain assets and hedge those bets so that the variance of the total excess return is minimized. As mentioned in Section 2.2, there are constraints that needs to be taken into consideration to be able to hedge the placed set of bets, since the proposed hierarchical portfolio allocation algorithm operates in an active management framework. For instance, it is not allowed to hedge a certain bet using an asset of dissimilar characteristic. In this thesis, the considered characteristic is the covariance of the excess returns between the assets. Highly correlated assets are assumed to be similar and vice versa.

Based on the previously detailed objective and constraint, the different constellations of HPAA that has been developed in this study encompasses four stages:

(I) Hierarchical clustering of assets

(II) Cluster selection

(III) Allocation of bets within clusters

(IV) Scaling of bets across clusters

The first step performs hierarchical clustering on assets to obtain a hierarchical structure. The second step applies a modified version of Gap Statistic Index presented by Yue, Wang, and Wei (2008) to determine the initial number of clusters considered. This number is then used as the first level in the hierarchical structure to start the recursive search for the final number of clusters where all clusters are invertible. In the third step, the bets within clusters are calculated based on Markowitz's minimum variance. Finally, in step four the bet scaling between clusters is determined. These four stages are the foundation of each portfolio construction method developed in this thesis. However, every construction method is unique since the primary and the final step can be changed using different modifications. The following sections describe each step of the hierarchical portfolio allocation algorithm, the rationale behind every step as well as the implemented modifications.

5.1.1 Hierarchical Clustering

As previously stated, all of the evaluated allocation algorithms consists of a primary hierarchical clustering stage which follows the approach described in Section 4.5, where the sample correlation matrix $\hat{\boldsymbol{C}}$ is computed using the normalized returns. Using the sample correlation matrix $\hat{\boldsymbol{C}}$, a distance matrix $\boldsymbol{D} = \{d_{i,j}\}_{i,j=1}^{p}$ is calculated using Equation (4.6). From the distance matrix $\boldsymbol{D}$, a hierarchical structure of the assets is constructed in an agglomerative fashion using a different set of linkage criteria, such as Single-, Complete- and Group Average Linkage as well as Ward's Method.

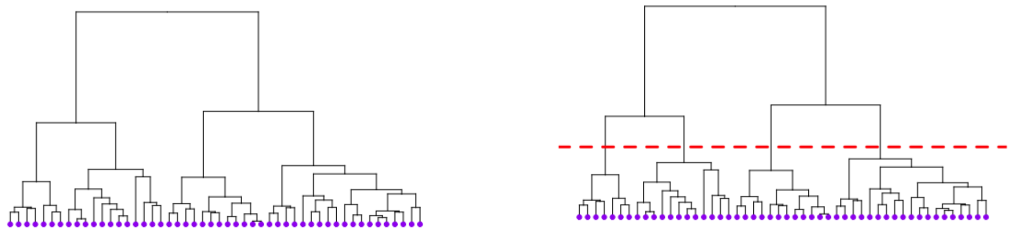
5.1.1.1 Optimal Leaf Ordering

To complement the primary step 5.1.1, Optimal Leaf Ordering was implemented (see Section 3.10) to optimally reorder the leaves in the hierarchical structure to examine the algorithm's impact on performance.

5.1.2 Cluster Selection

In order to obtain the clusters used for calculating the optimal set of bets, not only is the hierarchical structure of the assets necessary but also the number of clusters is needed. A natural approach to find the optimal number of clusters k^* is to apply the Gap Statistic Index (GSI), described in Section 3.9. Using a modified version of this approach for this study (see Equation (3.11)), the number of clusters obtained k^* provide an optimal middle ground between overfitting and information loss.

However, one problem with using Gap Statistic Index is to determine the cluster selection. In order to place bets in the next stage, the sample covariance matrix $\boldsymbol{S}_c$ associated with each clusters must be invertible. This is due to the fact that the optimal set of bets for every cluster $\boldsymbol{b}_c^*$ is a function of $\boldsymbol{S}_c^{-1}$ (See Equation (2.26)). Therefore, the optimal number of clusters k^*, given by GSI, is not equal to the final number of clusters considered in the latter stages of HPAA. Instead, k^* represents the level in the hierarchical structure from where to prune the binary tree from a top-down perspective and start the recursive search for invertible clusters. This is illustrated in the following Figure 5.1:



(a) Hierarchical structure before pruning

(b) Hierarchical structure after pruning

Figure 5.1: Illustration of before and after pruning of the hierarchical structure using k^*. The red dashed line represents where the number of clusters equals k^*.

Then, starting with k^* clusters, the hierarchical algorithm transverses through the binary tree using a top-down approach and checks the invertibility of the covariance matrix $\boldsymbol{S}_c$ for each cluster c. If $\boldsymbol{S}_c$ is not invertible, the associated cluster c is partitioned into two new sub-clusters with respect to the hierarchical structure. If $\boldsymbol{S}_c$ is invertible, the algorithm will not partition c into sub-clusters. This procedure is then applied recursively until all the covariance matrices associated with a cluster are invertible and there are k_{final} clusters. Note that this second stage of the hierarchical algorithm is identical for all of the different portfolio construction methods.

5.1.3 Within-Cluster Bet Selection

Using the clusters obtained from the previous step, the optimal set of bets is constructed. This is achieved by computing the sample covariance matrix $\boldsymbol{S}_c$, associated with each cluster c. Using $\boldsymbol{S}_c$, the optimal set of bets $\boldsymbol{b}_c^*$ for every cluster $c \in \{1, \ldots, k_{final}\}$ is obtained by solving the following optimization problem:

$$\begin{aligned} \boldsymbol{b}_c^* \in \arg\min \quad & \boldsymbol{b}_c^T \boldsymbol{S}_c \boldsymbol{b}_c \\ \text{s.t.} \quad & \boldsymbol{b}_c^T \mathbf{1} = 0 \\ & b_{c_q} = 1. \end{aligned} \tag{5.1}$$

Since the objective of the evaluated allocation algorithms is to place bets on assets and hedge those bets so that the variance of the total excess return is minimized, the selection of which bet that is needed to be hedged is necessary. This is achieved by equalling b_{c_q} to one in the last constraint of Problem (5.1). q is the index of the asset in the cluster that has had the highest normalized return from the previous week. One can choose amongst several different criteria to obtain q, however, the previously detailed approach was chosen in this thesis. Additionally, for clusters only containing one asset the bet is set to zero, since in order to hedge a bet, there must be another asset to hedge it against. Finally, one obtains an optimal set of bets $\boldsymbol{b}^*$, which is the concatenation of $\boldsymbol{b}_c^* \; \forall c \in \{1, \ldots, k_{final}\}$. This stage is also the same for all portfolio construction methods.

5.1.4 Between-Cluster Allocation

The last step of the proposed allocation method is to scale bets between clusters. In this thesis, the equally-weighted portfolio (see Section 4.1) and a modified version of equal risk contribution (see Section 4.4) are used and evaluated.

In the original equal risk contribution method proposed by Maillard et. al. (2010), the risk contribution from asset i is defined as

$$RC_i = w_i \frac{(\boldsymbol{\Sigma w})_i}{\sqrt{\boldsymbol{w}^T \boldsymbol{\Sigma w}}} \tag{5.2}$$

However, to adapt Equation (4.4) to active management where the interest lies in the minimization of the variance of the excess return rather than the variance of the portfolio, a different approach is utilized to calculate the risk contribution RC_c for every cluster c, namely:

$$RC_c = \frac{\boldsymbol{b}_c^T \boldsymbol{\Sigma}_c \boldsymbol{b}_c}{\sum_{c=1}^{k_{final}} \boldsymbol{b}_c^T \boldsymbol{\Sigma}_c \boldsymbol{b}_c} \tag{5.3}$$

In short, the heuristics behind equal risk contribution is that each asset in a portfolio should contribute to an equal amount of risk to the overall risk of the portfolio (see Section 4.4). In this case, clusters are considered instead of individual assets and thus should every cluster have an equal risk contribution. To achieve this, the bets $\boldsymbol{b}_c^*$ for every cluster c are scaled by a scaling factor θ_c. In the case of clusters with only one asset, the associated θ_c is set to zero, since the risk contribution for those clusters is equal to zero. Finally, to obtain the scaling vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_{k_{final}}]$ such that the risk contribution from every non-singleton cluster is equal, the algorithm solves the following optimization problem:

$$\begin{aligned} \min \quad & \boldsymbol{\theta}^T \boldsymbol{RC} \\ \text{s.t.} \quad & \sum_{i=1}^{m} \theta_i = 1 \\ & \theta_i RC_i = \theta_j RC_j \quad \forall \quad i, j \in \mathcal{D} \\ & \theta_i = 0 \quad \forall \quad i \in \mathcal{E} \end{aligned} \tag{5.4}$$

where $\boldsymbol{RC} = [RC_1, RC_2, \ldots, RC_{k_{final}}]$, $\mathcal{D}$ is the set of cluster indices for the clusters containing more than one asset, and $\mathcal{E}$ is the set of cluster indices for the clusters containing only one asset.

Furthermore, when using an equal-weighted portfolio approach to determine the between-cluster allocation (see Section 4.1), the scaling vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_{k_{final}}]$ is calculated by:

$$\begin{aligned} \theta_i &= \frac{1}{\|\mathcal{D}\|} \quad \forall \quad i \in \mathcal{D} \\ \theta_j &= 0 \quad \forall \quad j \in \mathcal{E} \end{aligned} \tag{5.5}$$

where $\mathcal{D}$ is the set of cluster indices for the clusters containing more than one asset, and $\mathcal{E}$ is the set of cluster indices for the clusters containing only one asset.

Once $\boldsymbol{\theta}$ has been calculated, $\boldsymbol{b}_c^*$ is scaled:

$$\boldsymbol{b}_c^{scaled} = \theta_c \cdot \boldsymbol{b}_c^* \tag{5.6}$$

Finally, one obtains a vector of scaled bets $\boldsymbol{b}^{scaled}$, which is the concatenation of $\boldsymbol{b}_c^{scaled} \; \forall \; c \in \{1, \ldots, k_{final}\}$.

5.2 Summary of Evaluated Allocation Methods

To summarise this section, there are sixteen different constellations of HPAA which are evaluated in this thesis. All of the methods incorporate similar cluster selection and within-cluster bet selection. The clustering methods, the use of optimal leaf ordering and between-cluster allocation methods vary between the different portfolio allocation methods. This is visualized in Table 5.1 below (see List of Acronyms for portfolio names).

Portfolio	Linkage Criteria				Optimal Leaf Ordering		Between-Cluster Allocation	
	Single	Group Average	Complete	Ward's Method	Yes	No	EW	ERC
SOERC	x				x			x
SOEW	x				x		x	
SERC	x					x		x
SEW	x					x	x	
GAOERC		x			x			x
GAOEW		x			x		x	
GAERC		x				x		x
GAEW		x				x	x	
COERC			x		x			x
COEW			x		x		x	
CERC			x			x		x
CEW			x			x	x	
WOERC				x	x			x
WOEW				x	x		x	
WERC				x		x		x
WEW				x		x	x	

Table 5.1: Overview of allocation algorithms. The names of the different allocation algorithms are acronyms based on their linkage criteria, usage of optimal leaf ordering and between-cluster allocation. For example, SOERC incorporates Single Linkage, Optimal Leaf Ordering and Equal Risk Contribution.

5.3 Performance Measures

The different portfolio construction methods were evaluated both on their ability to replicate the returns of a benchmark with a similar risk level, and the stability of the calculated bets over time. The ability to replicate the returns of a benchmark is assessed by using Tracking Error as a performance measure. In addition, the stability of the calculated bets is determined by the Turnover Ratio. To expand the performance analysis of the proposed hierarchical portfolio allocation algorithm, the number of clusters given by GSI and the final number of clusters were stored. In addition, the condition number for each sample covariance matrix for every cluster in the final number of clusters was computed in order to analyze the numerical stability of the estimations. Finally, the cluster size, meaning the average number of assets in a cluster was computed for comparison purposes. These measures are defined and described in the following subsections.

5.3.1 Tracking Error

Tracking Error is the divergence between the return of a portfolio and the return of a related benchmark, with a similar level of risk. In this thesis, the risk is defined as the variance of the excess return. In a relative setting like this, Tracking Error is a commonly used metric to gauge how well an investment in a portfolio is performing. Most portfolios behave differently compared to the benchmark and the Tracking Error is used to quantify this difference. In this thesis, the Tracking Error is defined in line with the one used by The Second Swedish National Pension Fund

$$\text{TE} = s\left(\left\{\rho\left(\boldsymbol{r}^{(t,t+\Delta t)}, \boldsymbol{b}^{*(t)}\right)\right\}_{t=1}^{N}\right), \tag{5.7}$$

where $\boldsymbol{b}^{*(t)}$ are the bets placed at time t, $\boldsymbol{r}^{(t,t+\Delta t)}$ is the return vector between time t and $t+\Delta t$, $\rho(\boldsymbol{x}, \boldsymbol{y})$ is the Pearson sample correlation between a vector $\boldsymbol{x}$ and a vector $\boldsymbol{y}$, and $s(\boldsymbol{x})$ is the sample standard deviation of a vector $\boldsymbol{x}$. This definition measures how much the correlation between the bets and the returns fluctuates over time.

5.3.2 Turnover

To measure the stability of the bets as well as to quantify how much $\boldsymbol{b}^*$ changes over time, a turnover measure was implemented. The mean and the standard deviation of the turnover measure were studied. The Turnover between time t and $t+\Delta t$, $TO^{(t)}$ was defined as follows:

$$TO^{(t)} = \left\|\boldsymbol{b}^{*(t)} - \boldsymbol{b}^{*(t+\Delta t)}\right\|. \tag{5.8}$$

The sample mean of the Turnover is

$$\overline{TO} = \sum_{t=1}^{n-1} \frac{TO^{(t)}}{n-1},. \tag{5.9}$$

and its sample standard deviation is

$$s_{TO} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(TO^{(i)} - \overline{TO})^2}. \tag{5.10}$$

The goal of the hierarchical portfolio allocation algorithms is to have $\overline{TO}$ and s_{TO} as small as possible since it is typically undesirable to change positions frequently.

5.3.3 Condition Number

To evaluate the numerical stability of the sample covariance matrices associated with the clusters (see Section 5.1.3), the condition number of each sample covariance matrix was computed and stored. The condition number was calculated using Equation (2.28). Since there typically are several clusters, and thus several sample covariance matrices, only the minimum $\kappa_{Min}^{(t)}$, maximum $\kappa_{Max}^{(t)}$, average $\mu_{\kappa}^{(t)}$ and standard deviation $\sigma_\kappa^{(t)}$ of the condition numbers were stored at each time t. What is presented below is the averages of these measures,

$$\bar{\kappa}_{Min} = \frac{\sum_{i=1}^{n} \kappa_{Min}^{(i)}}{n}, \tag{5.11}$$

$$\bar{\kappa}_{Max} = \frac{\sum_{i=1}^{n} \kappa_{Max}^{(i)}}{n}, \tag{5.12}$$

$$\overline{\mu_\kappa} = \frac{\sum_{i=1}^{n} \mu_\kappa^{(i)}}{n}, \tag{5.13}$$

and

$$\overline{\sigma_\kappa} = \frac{\sum_{i=1}^{n} \sigma\kappa^{(i)}}{n}. \tag{5.14}$$

5.3.4 Number of Clusters

In order to evaluate the potential overfitting of the clusters, the number of formed clusters $k_{final}^{(t)}$ is compared to the number of clusters suggested by the Gap Statistics Index, $k^{*,(t)}$, resulting in the value $\Delta k^{(t)} = |k_{final}^{(t)} - k^{*,(t)}|$, for each time t. The measures presented are the mean and standard deviation of $\Delta k^{(t)}$ over time,

$$\overline{\Delta k} = \frac{\sum_{i=1}^{n} \Delta k^{(i)}}{n} \tag{5.15}$$

and

$$s_{\Delta k} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (\Delta k^{(i)} - \overline{\Delta k})^2}. \tag{5.16}$$

5.3.5 Cluster Size

The size of each of the clusters, i.e. the number of assets in each cluster, for every allocation algorithm was stored to inspect how the cluster size varies based on the parameter configuration for each portfolio. The average cluster size and standard deviation of the cluster sizes generated by every allocation algorithm were computed at each time t. The presented results are the averages of these measures over time such that,

$$\overline{n_C} = \frac{\sum_{i=1}^{n} n_C^{(i)}}{n} \tag{5.17}$$

and

$$s_{n_C} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (n_C^{(i)} - \overline{n_C})^2}. \tag{5.18}$$

Note that n denotes the number of time steps considered and n_C denotes the number of assets in a cluster, i.e. the cluster size.

5.4 Empirical Walk-Forward Backtest

To construct the sample covariance matrices, the normalized returns for the asset are essential. As previously discussed in Section 2.3, the number of observations n used to construct the sample covariance matrix $\boldsymbol{S}$ is a parameter of great importance that has a considerable influence on the quality of $\boldsymbol{S}$. As previously stated, when $n \to \infty$ with p fixed, the sample covariance matrix $\boldsymbol{S}$ converges to the true covariance matrix $\boldsymbol{\Sigma}$. However, in practice investors desire to use as recent return data as possible to reflect current market conditions. Therefore, a middle ground between a too narrow and too wide time window is required. To evaluate the influence of n on the quality of the purposed hierarchical portfolio allocation algorithm, three different time windows are considered for constructing the sample covariance matrix $\boldsymbol{S}$: 100-, 300- and 500 days.

Since the return data is in the form of time series, a rolling window analysis is performed. Denote the total number of days in the dataset as $\|T\|$. The sample covariance matrix $\boldsymbol{S}$ is then estimated on $m := \frac{\|T\|}{n}$ subsets of normalized returns for all p assets in the dataset, where $n \in \{100, 300, 500\}$. Using $\boldsymbol{S}$, the optimal set of bets is computed using Equation (2.26) for every subset m. The estimation procedure and bets computation are performed iteratively by starting from the beginning of the dataset and incrementing m by one until the complete dataset has been iterated through.

5.5 Data

This section includes a presentation and description of the empirical data used in this study, as well as a presentation of the data manipulations performed.

5.5.1 Original Dataset

The empirical data considered in this study consists of weekly equity returns for 5815 stocks in the time period between 01/03/1990 and 10/01/2022. Note that not all assets have available weekly returns for the whole period. All data used in this project were provided by The Second Swedish National Pension Fund.

The data provided by The Second Swedish National Pension Fund consisted of weekly returns for each asset, the currency that the asset is denoted in, the country listing of the asset as well as the Global Industry Classification Standard (GICS) developed by MSCI and Standard & Poor's, geographical market of the asset. The weekly equity returns $r_i = r_i^{(t,t+\Delta t)} \in \mathbb{R}^p$ for asset i are defined as

$$r_i^{(t,t+\Delta t)} = \frac{P_i^{(t+\Delta t)} - P_i^{(t)} + d_i^{(t)}}{P_i^{(t)}}, \tag{5.19}$$

where $P_i^{(t)}$ is the closing price at time t for asset i and $d_i^{(t)}$ is the dividend payed for asset i between t and $t + \Delta t$. Since the dataset consist of weekly returns, Δt

corresponds to one week, i.e. five trading days. All returns are given in Swedish currency SEK.

The different geographical markets and the number of assets associated to each market is shown in Table 5.2:

Geographical Market	**Number of Assets**
Asia	1115
Eastern Europe	154
Europe	1392
Japan	618
Latin America	257
Middle East / Africa	213
North America	1650
Pacific / Japan	471

Table 5.2: The geographical distribution of assets.

5.5.2 Data Manipulation

Since the interest of this thesis lies in the excess return, all weekly equity returns have been normalized by the subtraction of the average return $\bar{r} \in \mathbb{R}$ between t and Δt, and division of the sample standard deviation of the market return $s(r)$. Here, $r \in \mathbb{R}^p$ is a vector of the returns of the assets in the portfolio between time t and Δt, such that the normalized weekly return $r_{z_i} \in \mathbb{R}$ for assets i is given by

$$r_{z_i} = \frac{r_i - \bar{r}}{s(r)} \quad \forall i = 1, \ldots, p \tag{5.20}$$

Evidently, $\bar{r}_{z_i} = 0,\ s\left(r_{z_i}\right) = 1\ \ \forall i = 1, \ldots, p$ at every time $t > 0$.

To handle the fact that weekly return data is not available for each asset during the whole time period, the data set was divided into four subsets of an equal number of weekly returns. These four subsets consist of the data associated with the assets with consistent weekly returns between 1990-03-01 and 1998-02-13, 1998-02-16 and 2006-01-31, 2006-02-01 and 2014-01-16, 2014-01-17 and 2022-01-03, respectively. This resulted in the geographical market distribution illustrated in Table 5.3.

	Periods			
Geographical Market Belonging	Period 1	Period 2	Period 3	Period 4
Asia	23	0	224	231
Eastern Europe	0	0	29	13
Europe	362	221	258	246
Japan	266	180	198	147
Latin America	0	0	49	51
Middle East / Africa	0	0	25	20
North America	356	221	385	365
Pacific / Japan	148	51	90	68
Total	**1155**	**673**	**1256**	**1141**

Table 5.3: The geographical distribution of assets with consistent weekly returns for different time periods.

Table 5.3 illustrates that the number of assets decreased from partitioning the data using the aforementioned approach. On the other hand, one can argue that the total number of assets is still large enough in each period for the portfolios to be considered high-dimensional and thus, it can be used to achieve the aims of this thesis.

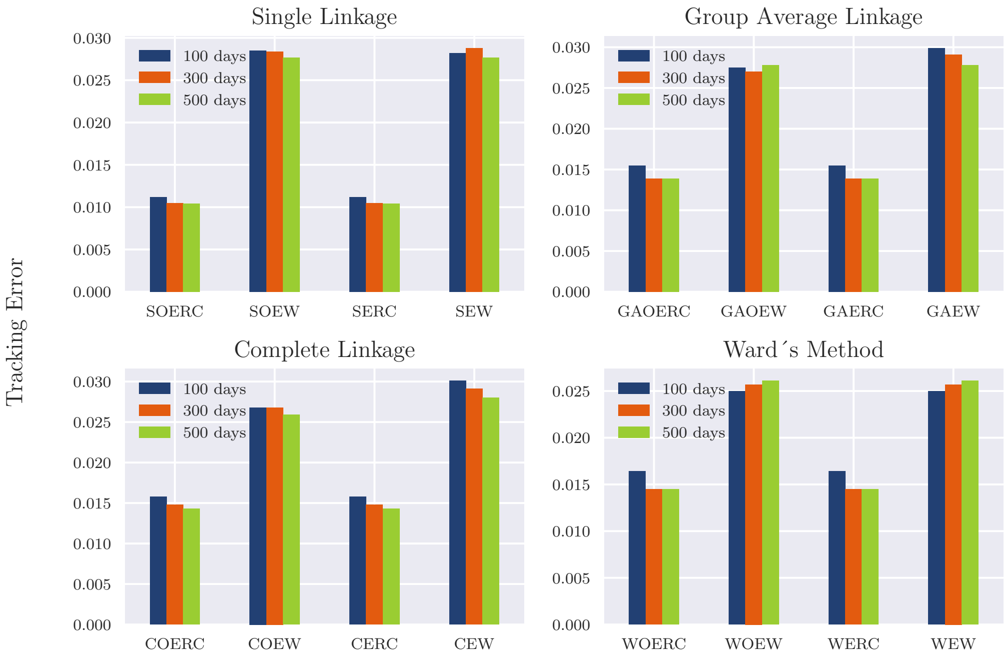
6

Results

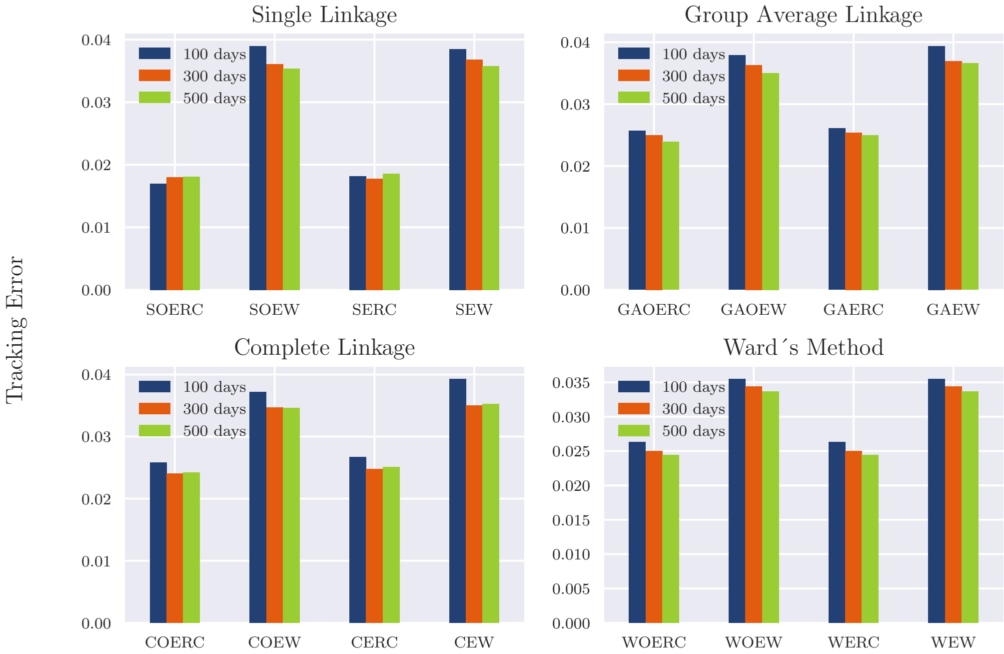
In this chapter, the empirical results are presented. This chapter is divided into sections associated with each of the performance measures presented in Section 5.3, namely Tracking Error, Turnover, condition number, number of clusters and cluster size. The results includes the investigated time periods, see Section 5.5.2, and were obtained following the methodology described in the previous chapter.

6.1 Tracking Error

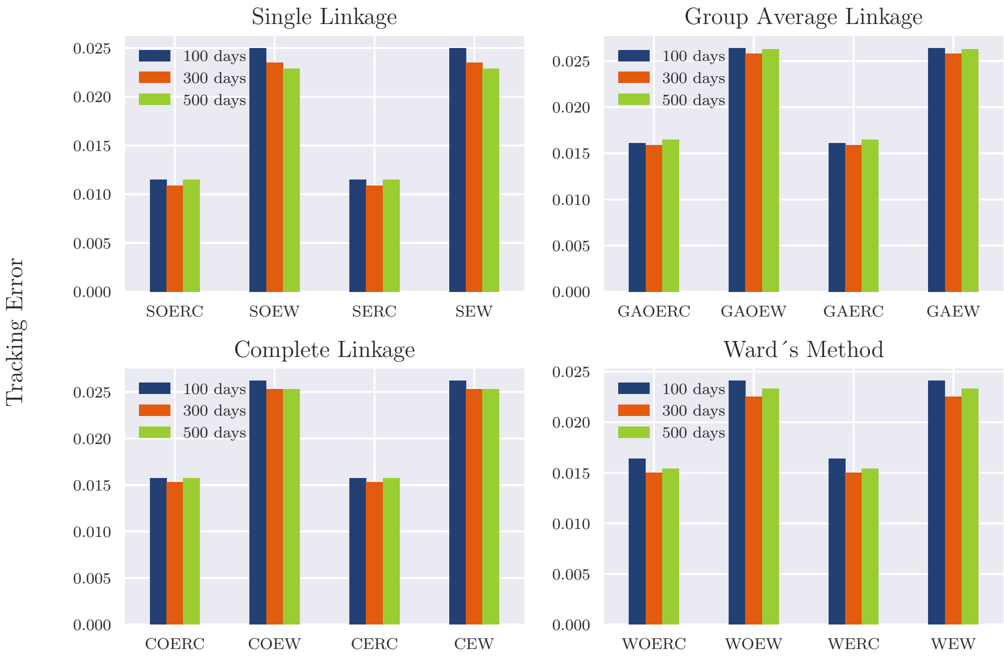
From the sub-figures in Figure 6.1, it is evident that the Tracking Error produced by the different portfolio construction methods do not change considerably over time since the results are rather similar irrespective of which time period is considered. This implies that the proposed hierarchical clustering allocation algorithm is robust over time and for different market regimes. Note that the Tracking Error in period two is larger for all allocation algorithms compared to the other periods. This indicates that all the allocation algorithms perform better when more assets are added to the portfolio. The combination of Single Linkage and ERC outperformed the rest of the portfolios for all considered time periods. The implementation of ERC as a between-cluster allocation method seems to produce superior portfolios compared to EW, which produces relatively high TE. In addition, optimal leaf ordering seems to have a little or no effect on the produced TE across all linkage criteria. Similarly, the size of the time windows used for the construction of covariance and correlation matrices does not enhance the TE remarkably. However, larger time windows do usually produce smaller TE in general, even if the improvement is small.



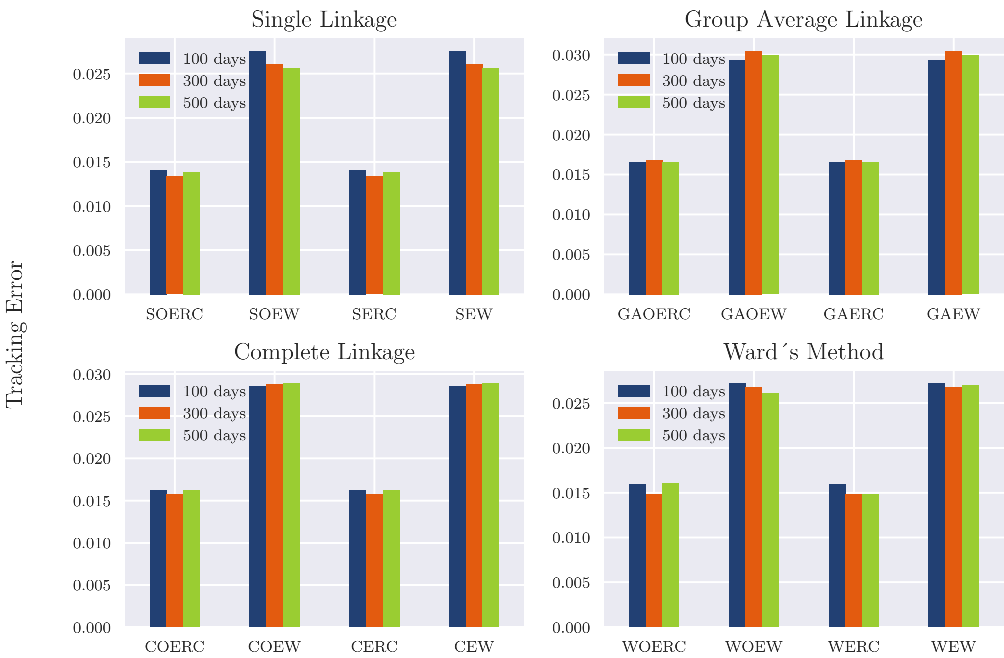
(a) Average TE for different portfolio construction methods in Period 1, corresponding to the period between 01/03/1990 and 13/02/1998.



(b) Average TE for different portfolio construction methods in Period 2, corresponding to the period between 16/02/1998 and 31/01/2006.



(c) Average TE for different portfolio construction methods in Period 3, corresponding to the period between 01/02/2006 and 16/01/2014.

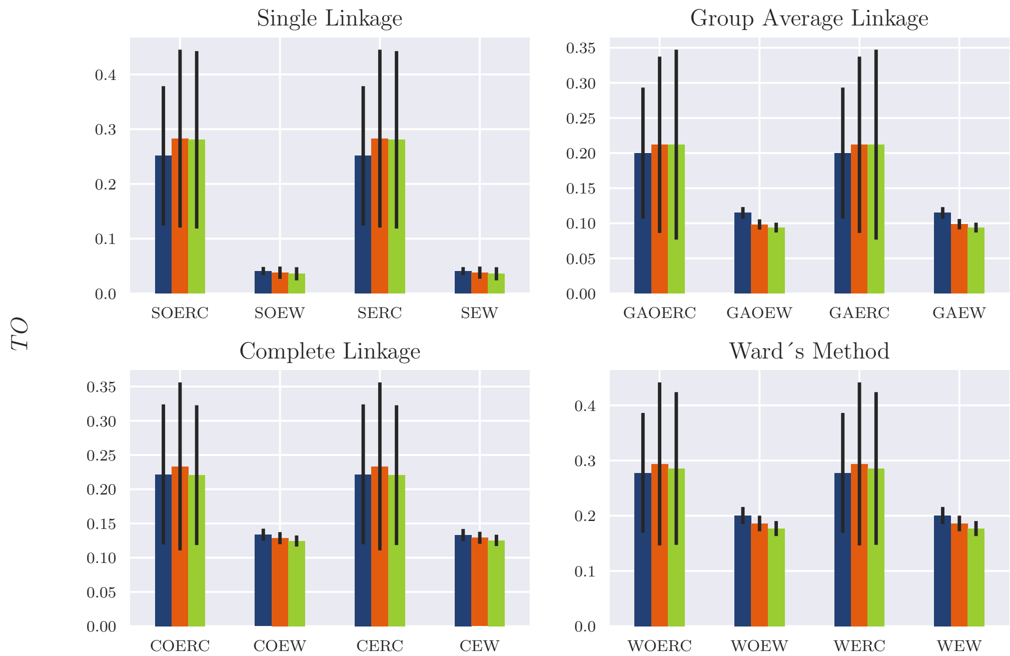


(d) Average TE for different portfolio construction methods in Period 4, corresponding to the period between 17/01/2014 and 03/01/2022.

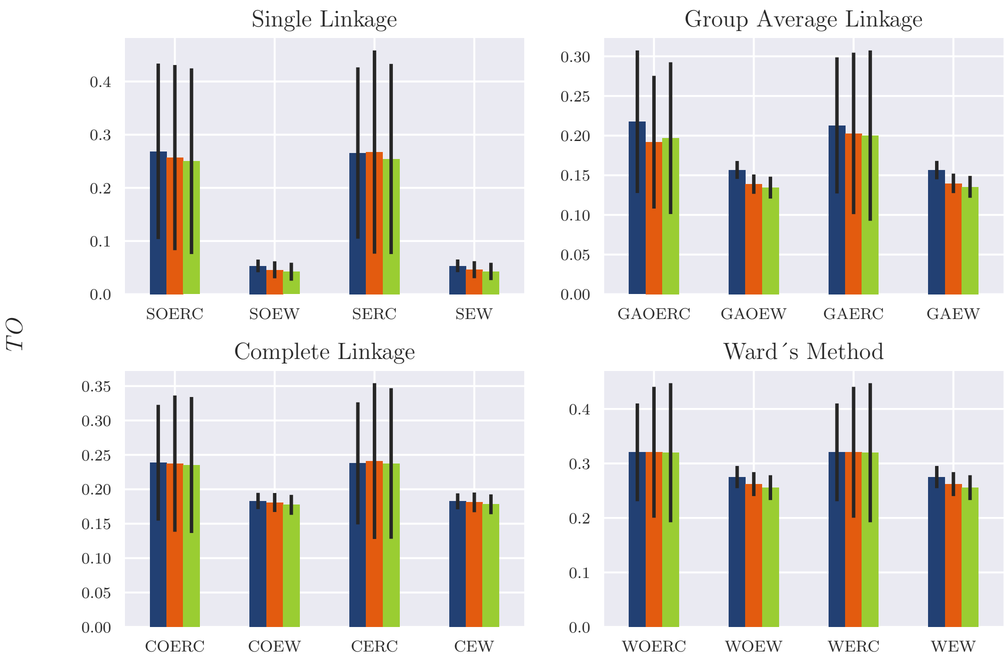
Figure 6.1: Average TE for different portfolio construction methods during different time periods.

6.2 Turnover

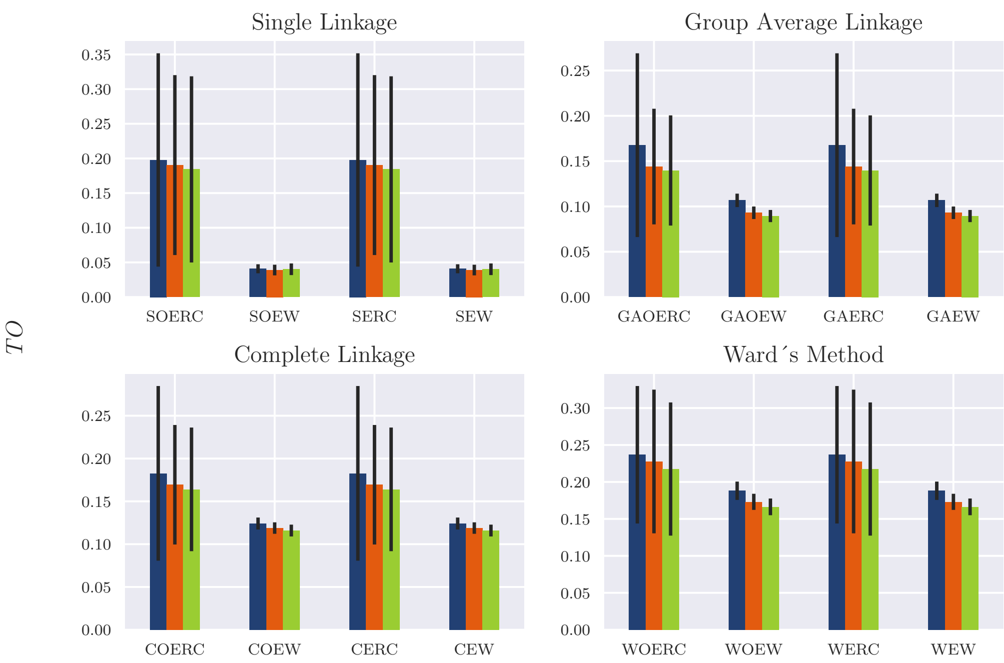
The Turnover TO is illustrated in Figure 6.2. Similar to Tracking Error, the overall patterns are similar across different time periods, and exists clear differences in the performance between different portfolio construction methods that are consistent over time. Portfolios incorporating EW produce better results since the Turnover of bets is considerably smaller than for portfolios constructed using ERC. Additionally, the set of bets produced using EW does not fluctuate frequently since the standard deviation is significantly smaller in comparison to ERC. This difference is most distinguishable for portfolios using Single Linkage as linkage criteria. It is evident that the combination of Single Linkage and EW (SOEW and SEW) yields very stable bets over time with little variation. Moreover, Single Linkage seems to be the favorable linkage criteria with respect to TO. However, it is difficult to establish any substantial differences between the Turnover results of the remaining allocation algorithms, barring the portfolios created using Ward's method that experience a slightly higher turnover of bets. Additionally, optimal leaf ordering and the size of the time windows do not seem to have a significant impact on the turnover of bets over time.



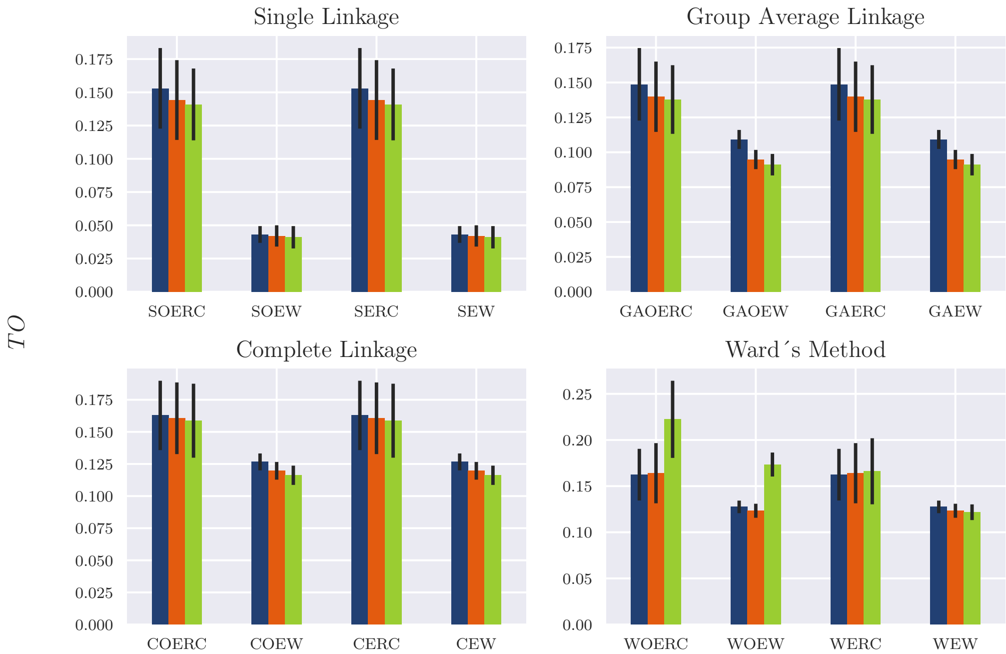
(a) Average TO and s_{TO} for different portfolio construction methods in Period 1, corresponding to the period between 01/03/1990 and 13/02/1998.



(b) Average TO and s_{TO} for different portfolio construction methods in Period 2, corresponding to the period between 16/02/1998 and 31/01/2006.



(c) Average TO and s_{TO} for different portfolio construction methods in Period 3, corresponding to the period between 01/02/2006 and 16/01/2014.



(d) Average TO and s_{TO} for different portfolio construction methods in Period 4, corresponding to the period between 17/01/2014 and 03/01/2022.

Figure 6.2: Average TO for different allocation algorithms during different time periods. The black line represents the standard deviation of the TO for every portfolio constellations. The blue, orange and green bars represents the results using time windows of size 100-, 300- and 500 days, respectively.

6.3 Condition Number

Table 6.1 shows as expected, that the condition number for the different portfolio construction methods decrease as one increases the number of observations used for constructing the covariance and correlation matrices. This is true for all time periods and portfolio construction methods considered, and is in line with the discussion in Section 2.3. However, the effect diminishes when going from $n = 300$ to $n = 500$ in comparison to when going from $n = 100$ to $n = 300$. This is as expected, since the standard deviation of the estimates decrease with a factor $\frac{1}{\sqrt{n}}$. Single Linkage portfolios outperformed the other portfolios with respect to κ, and also exhibiting the lowest variability σ_κ. Concerning condition number, Single Linkage portfolios performed best and were followed by Group Average Linkage, Complete Linkage, and lastly, constellations constructed using Ward's method. The discrepancy between the condition numbers by Single Linkage and Ward's method is large, Ward's method yield on average 4.5 times larger condition numbers across the different time windows. There exists no difference in the condition numbers obtained when implementing EW or ERC, as expected. This is due to the fact the obtained covariance matrices are the same, irrespective of which between-cluster allocation method is used. Interestingly, leaf order does not have any impact on the condition numbers. The condition numbers are in general smaller in period two compared to the other periods, which might depend on the fact that this period contains fewer assets.

$n = 100$				
	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	3.21	663.46	27.88	70.05
SOEW	3.21	663.46	27.88	70.05
SERC	3.21	663.46	27.88	70.05
SEW	3.21	663.46	27.88	70.05
GAOERC	2.58	1019.20	45.55	101.87
GAOEW	2.58	1019.20	45.55	101.87
GAERC	2.58	1019.20	45.55	101.87
GAEW	2.58	1019.20	45.55	101.87
COERC	3.26	945.69	54.45	103.96
COEW	3.26	945.69	54.45	103.96
CERC	3.26	945.69	54.45	103.96
CEW	3.26	945.69	54.45	103.96
WOERC	6.17	1588.70	134.91	217.72
WOEW	6.17	1588.70	134.91	217.72
WERC	6.17	1588.70	134.91	217.72
WEW	6.17	1588.70	134.91	217.72

$n = 300$				
	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	2.23	532.42	22.53	59.31
SOEW	2.23	532.42	22.53	59.31
SERC	2.23	532.42	22.53	59.31
SEW	2.23	532.42	22.53	59.31
GAOERC	1.77	767.90	25.48	73.42
GAOEW	1.77	767.90	25.48	73.42
GAERC	1.77	767.90	25.48	73.42
GAEW	1.77	767.90	25.48	73.42
COERC	2.20	722.07	37.97	84.26
COEW	2.20	722.07	37.97	84.26
CERC	2.20	722.07	37.97	84.26
CEW	2.20	722.07	37.97	84.26
WOERC	4.99	1029.08	84.17	150.33
WOEW	4.99	1029.08	84.17	150.33
WERC	4.99	1029.08	84.17	150.33
WEW	4.99	1029.08	84.17	150.33

$n = 500$				
	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	2.02	488.52	22.30	57.27
SOEW	2.02	488.52	22.30	57.27
SERC	2.02	488.52	22.30	57.27
SEW	2.02	488.52	22.30	57.27
GAOERC	1.58	692.83	22.19	66.98
GAOEW	1.58	692.83	22.19	66.98
GAERC	1.58	692.83	22.19	66.98
GAEW	1.58	692.83	22.19	66.98
COERC	1.93	702.91	34.13	81.08
COEW	1.93	702.91	34.13	81.08
CERC	1.93	702.91	34.13	81.08
CEW	1.93	702.91	34.13	81.08
WOERC	4.41	888.83	72.03	131.25
WOEW	4.41	888.83	72.03	131.25
WERC	4.41	888.83	72.03	131.25
WEW	4.41	888.83	72.03	131.25

(a) Condition number κ all allocation algorithms in Period 1, corresponding to the period between 01/03/1990 and 13/02/1998.

$n = 100$

	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	3.01	361.47	21.38	46.91
SOEW	3.01	361.47	21.38	46.91
SERC	3.01	361.47	21.38	46.91
SEW	3.01	361.47	21.38	46.91
GAOERC	2.67	599.90	39.11	73.42
GAOEW	2.67	599.90	39.11	73.42
GAERC	2.67	599.90	39.11	73.42
GAEW	2.67	599.90	39.11	73.42
COERC	4.07	605.67	47.56	78.78
COEW	4.07	605.67	47.56	78.78
CERC	4.07	605.67	47.56	78.78
CEW	4.07	605.67	47.56	78.78
WOERC	13.64	1033.30	119.70	167.58
WOEW	13.64	1033.30	119.70	167.58
WERC	13.64	1033.30	119.70	167.58
WEW	13.64	1033.30	119.70	167.58

$n = 300$

	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	2.17	243.01	15.17	33.94
SOEW	2.17	243.01	15.17	33.94
SERC	2.17	243.01	15.17	33.94
SEW	2.17	243.01	15.17	33.94
GAOERC	1.78	400.89	22.12	50.03
GAOEW	1.78	400.89	22.12	50.03
GAERC	1.78	400.89	22.12	50.03
GAEW	1.78	400.89	22.12	50.03
COERC	2.91	448.69	33.46	63.89
COEW	2.91	448.69	33.46	63.89
CERC	2.91	448.69	33.46	63.89
CEW	2.91	448.69	33.46	63.89
WOERC	11.06	638.48	75.21	114.46
WOEW	11.06	638.48	75.21	114.46
WERC	11.06	638.48	75.21	114.46
WEW	11.06	638.48	75.21	114.46

$n = 500$

	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	1.97	195.60	13.53	28.22
SOEW	1.97	195.60	13.53	28.22
SERC	1.97	195.60	13.53	28.22
SEW	1.97	195.60	13.53	28.22
GAOERC	1.59	338.50	19.46	44.40
GAOEW	1.59	338.50	19.46	44.40
GAERC	1.59	338.50	19.46	44.40
GAEW	1.59	338.50	19.46	44.40
COERC	2.52	399.33	30.40	59.14
COEW	2.52	399.33	30.40	59.14
CERC	2.52	399.33	30.40	59.14
CEW	2.52	399.33	30.40	59.14
WOERC	9.94	563.73	68.37	105.30
WOEW	9.94	563.73	68.37	105.30
WERC	9.94	563.73	68.37	105.30
WEW	9.94	563.73	68.37	105.30

(b) Condition number κ all allocation algorithms in Period 2, corresponding to the period between 16/02/1998 and 31/01/2006.

$n = 100$

	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	3.23	593.08	27.09	62.28
SOEW	3.23	593.08	27.09	62.28
SERC	3.23	593.08	27.09	62.28
SEW	3.23	593.08	27.09	62.28
GAOERC	2.59	760.74	41.84	76.06
GAOEW	2.59	760.74	41.84	76.06
GAERC	2.59	760.74	41.84	76.06
GAEW	2.59	760.74	41.84	76.06
COERC	3.33	747.45	50.34	79.87
COEW	3.33	747.45	50.34	79.87
CERC	3.33	747.45	50.34	79.87
CEW	3.33	747.45	50.34	79.87
WOERC	10.72	1445.82	134.22	189.76
WOEW	10.72	1445.82	134.22	189.76
WERC	10.72	1445.82	134.22	189.76
WEW	10.72	1445.82	134.22	189.76

$n = 300$

	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	2.31	340.05	18.98	39.70
SOEW	2.31	340.05	18.98	39.70
SERC	2.31	340.05	18.98	39.70
SEW	2.31	340.05	18.98	39.70
GAOERC	1.76	437.39	24.08	47.17
GAOEW	1.76	437.39	24.08	47.17
GAERC	1.76	437.39	24.08	47.17
GAEW	1.76	437.39	24.08	47.17
COERC	2.27	478.67	34.52	56.53
COEW	2.27	478.67	34.52	56.53
CERC	2.27	478.67	34.52	56.53
CEW	2.27	478.67	34.52	56.53
WOERC	7.39	774.50	82.17	113.04
WOEW	7.39	774.50	82.17	113.04
WERC	7.39	774.50	82.17	113.04
WEW	7.39	774.50	82.17	113.04

$n = 500$

	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	2.13	288.56	17.49	35.01
SOEW	2.13	288.56	17.49	35.01
SERC	2.13	288.56	17.49	35.01
SEW	2.13	288.56	17.49	35.01
GAOERC	1.59	343.23	21.47	40.54
GAOEW	1.59	343.23	21.47	40.54
GAERC	1.59	343.23	21.47	40.54
GAEW	1.59	343.23	21.47	40.54
COERC	2.00	455.69	31.68	54.30
COEW	2.00	455.69	31.68	54.30
CERC	2.00	455.69	31.68	54.30
CEW	2.00	455.69	31.68	54.30
WOERC	6.78	603.80	70.91	90.79
WOEW	6.78	603.80	70.91	90.79
WERC	6.78	603.80	70.91	90.79
WEW	6.78	603.80	70.91	90.79

(c) Condition number κ all allocation algorithms in Period 3, corresponding to the period between 01/02/2006 and 16/01/2014.

$n = 100$

	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	3.22	800.44	28.47	73.76
SOEW	3.22	800.44	28.47	73.76
SERC	3.22	800.44	28.47	73.76
SEW	3.22	800.44	28.47	73.76
GAOERC	2.62	841.59	43.51	80.95
GAOEW	2.62	841.59	43.51	80.95
GAERC	2.62	841.59	43.51	80.95
GAEW	2.62	841.59	43.51	80.95
COERC	3.26	764.29	51.27	80.06
COEW	3.26	764.29	51.27	80.06
CERC	3.26	764.29	51.27	80.06
CEW	3.26	764.29	51.27	80.06
WOERC	4.40	746.83	55.02	79.24
WOEW	4.40	746.83	55.02	79.24
WERC	4.40	746.83	55.02	79.24
WEW	4.40	746.83	55.02	79.24

$n = 300$

	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	2.27	486.74	19.85	46.33
SOEW	2.27	486.74	19.85	46.33
SERC	2.27	486.74	19.85	46.33
SEW	2.27	486.74	19.85	46.33
GAOERC	1.84	509.95	24.75	48.69
GAOEW	1.84	509.95	24.75	48.69
GAERC	1.84	509.95	24.75	48.69
GAEW	1.84	509.95	24.75	48.69
COERC	2.29	512.73	34.47	56.61
COEW	2.29	512.73	34.47	56.61
CERC	2.29	512.73	34.47	56.61
CEW	2.29	512.73	34.47	56.61
WOERC	9.57	656.64	84.59	102.37
WOEW	9.57	656.64	84.59	102.37
WERC	9.57	656.64	84.59	102.37
WEW	9.57	656.64	84.59	102.37

$n = 500$

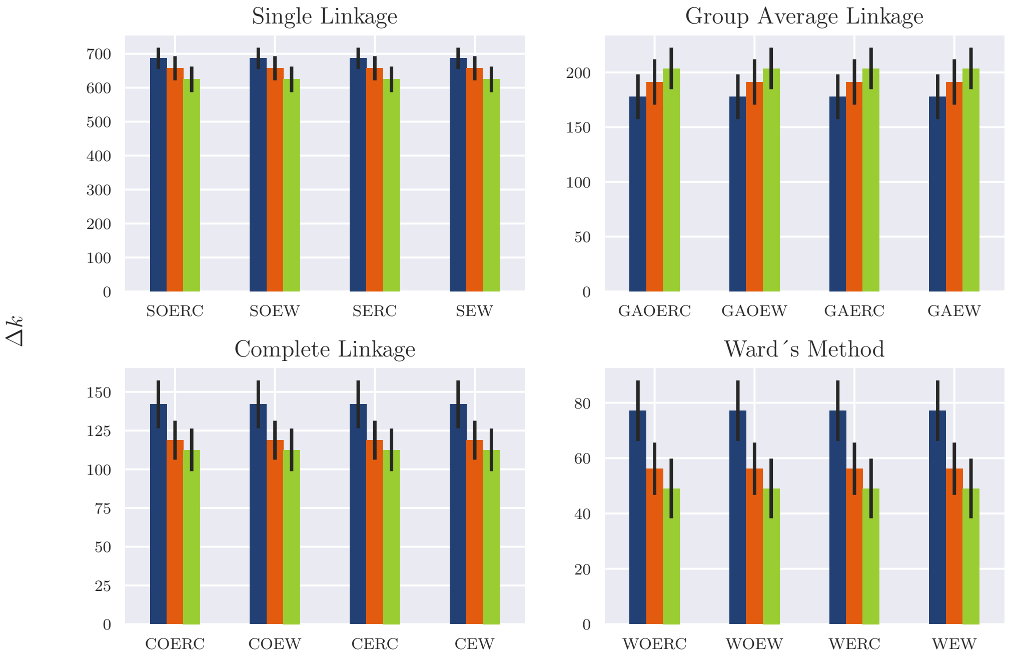
	κ_{Min}	κ_{Max}	μ_κ	σ_κ
SOERC	2.11	419.50	18.05	39.66
SOEW	2.11	419.50	18.05	39.66
SERC	2.11	419.50	18.05	39.66
SEW	2.11	419.50	18.05	39.66
GAOERC	1.62	462.65	21.62	43.65
GAOEW	1.62	462.65	21.62	43.65
GAERC	1.62	462.65	21.62	43.65
GAEW	1.62	462.65	21.62	43.65
COERC	2.03	453.67	30.66	50.54
COEW	2.03	453.67	30.66	50.54
CERC	2.03	453.67	30.66	50.54
CEW	2.03	453.67	30.66	50.54
WOERC	9.16	534.71	73.94	85.82
WOEW	9.16	534.71	73.94	85.82
WERC	9.16	534.71	73.94	85.82
WEW	9.16	534.71	73.94	85.82

(d) Condition number κ all allocation algorithms in Period 4, corresponding to the period between 17/01/2014 and 03/01/2022.

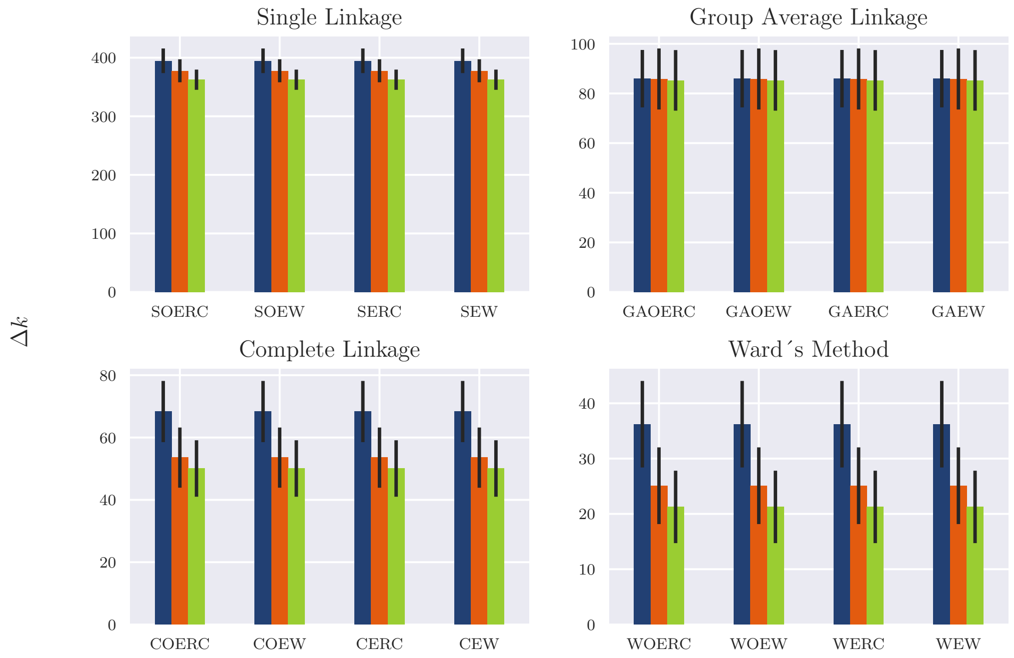
Table 6.1: Statistics for condition number κ for all allocation algorithms during the different time periods.

6.4 Number of Clusters

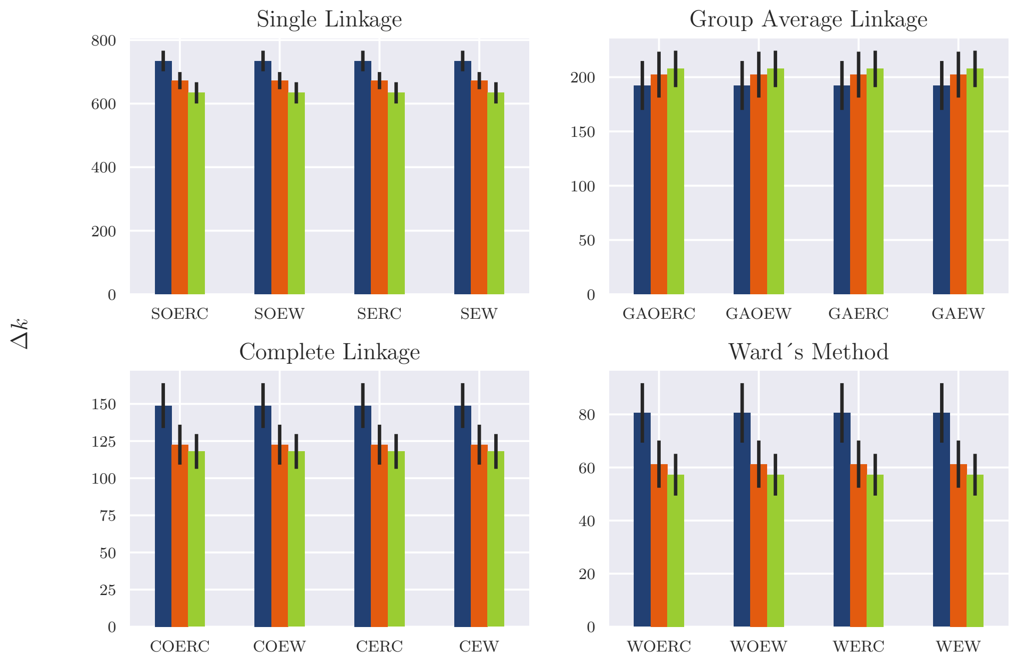
Figure 6.3 shows that there exists a clear discrepancy between the results from the different linkage criteria. Single Linkage portfolios create a large number of clusters, suggesting that it does not reflect the true hierarchical structure very well. These portfolios exhibit small variation, indicating that Single Linkage consistently creates a set of clusters that is far from the optimal number. In addition, for Single Linkage portfolios, the time window n does not reduce Δk considerably. This is in contrast to portfolios based on Ward's method, which produces sets of clusters that are much closer to k^*. Also, for Ward's method the size of the time window n imposes a more significant impact on the produced Δk, where larger n reduces the discrepancy between the final and the optimal number of clusters. This is most clear for the WOERC and WOEW portfolios in the fourth period, where one can see a considerable decrease in Δk when increasing the number of observations n. Allocation algorithms based on Complete and Group Average Linkage performed rather similarly with respect to Δk. On the other hand, a difference between Complete and Group Average Linkage portfolios is that the former criteria produced smaller Δk when using larger time windows n, whereas the opposite is true for Group Average Linkage portfolios. Note, that the reason why Δk is smaller in Period 2 is that this period contains fewer assets compared to the other periods.



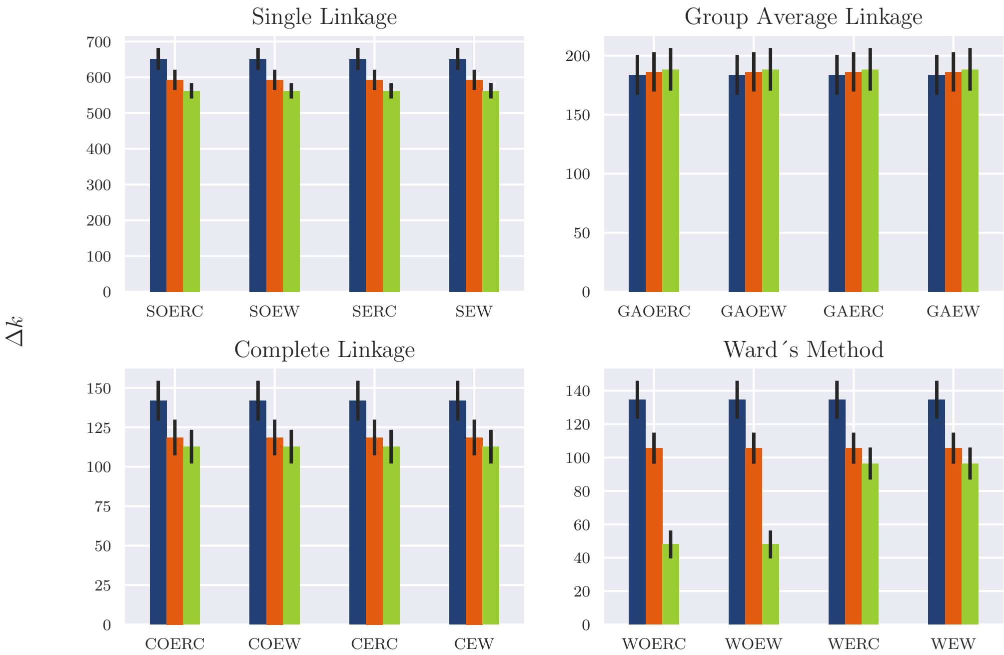
(a) Average Δk and $s_{\Delta k}$ for different portfolio construction methods in Period 1, corresponding to the period between 01/03/1990 and 13/02/1998.



(b) Average Δk and $s_{\Delta k}$ for different portfolio construction methods in Period 2, corresponding to the period between 16/02/1998 and 31/01/2006.



(c) Average Δk and $s_{\Delta k}$ for different portfolio construction methods in Period 3, corresponding to the period between 01/02/2006 and 16/01/2014.

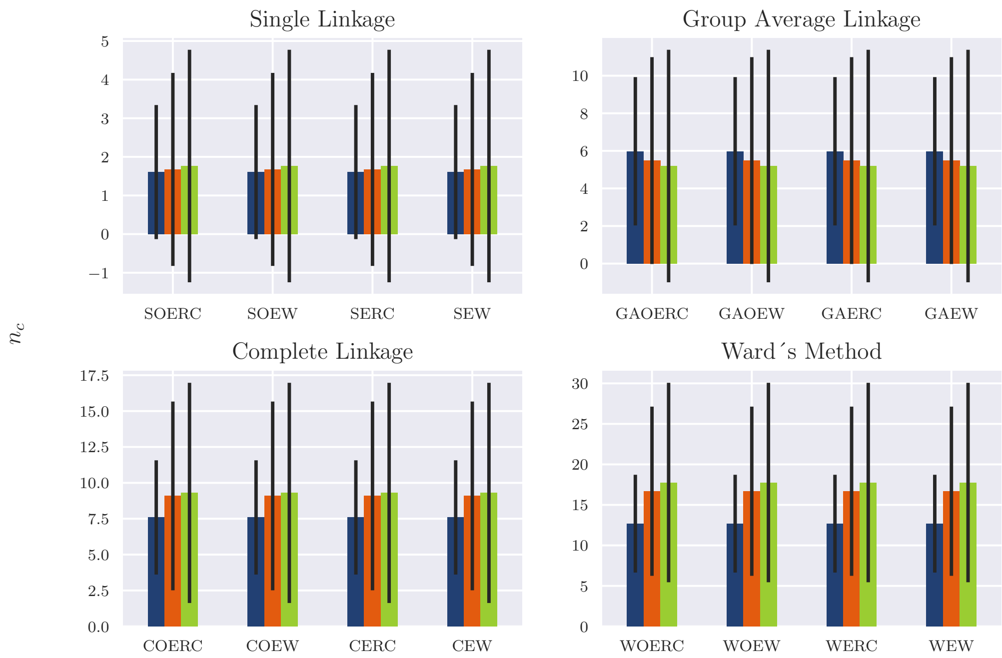


(d) Average Δk and $s_{\Delta k}$ for different portfolio construction methods in Period 4, corresponding to the period between 17/01/2014 and 03/01/2022.

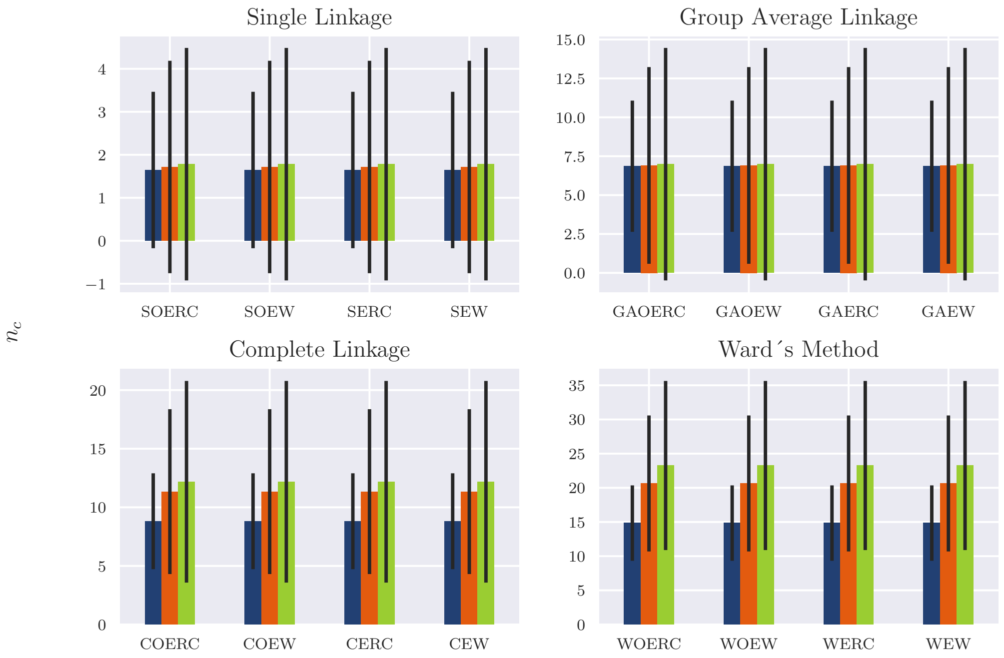
Figure 6.3: Average Δk for the different portfolio construction methods during different time periods. The black line represents the standard deviation of Δk for every portfolio construction method. The blue, orange and green bars represents the results using time windows of size 100-, 300- and 500 days, respectively.

6.5 Cluster Size

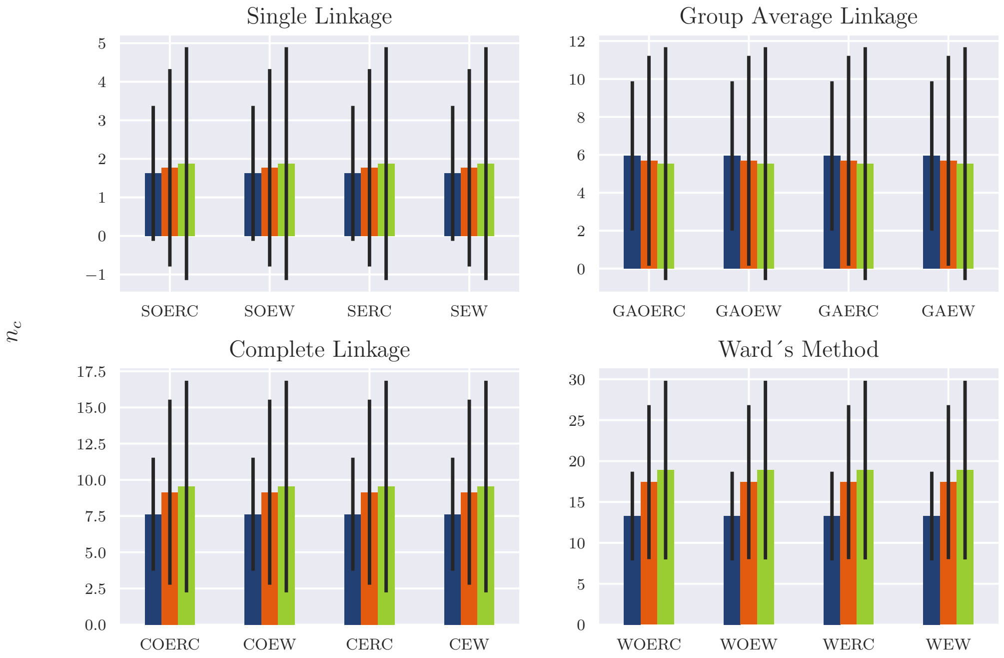
Figure 6.4 shows that as previously suspected, Single Linkage portfolios produce remarkably small clusters with very few assets. On the contrary, Ward's method creates clusters of larger size and where Complete and Group Average Linkage constitutes a middle ground for cluster size. For example, Ward's method portfolios produce approximately 9.8x, 1.9x, and 2.8x larger clusters on average than Single Linkage, Complete Linkage, and Average Linkage, respectively. Also, notice that for allocation algorithms using Ward's method and Complete Linkage, the average cluster size n_C increases as more data points are included, while the cluster size for allocation algorithms using Single and Group Average seems to be unaffected by the increase of n. Additionally, allocation algorithms incorporating these latter linkage criteria exhibit a larger variability regarding average cluster size compared to Ward's method and Complete Linkage. Finally, Figure 6.4 exhibit a natural relationship between Δk and n_C, namely that large Δk imply smaller clusters on average.



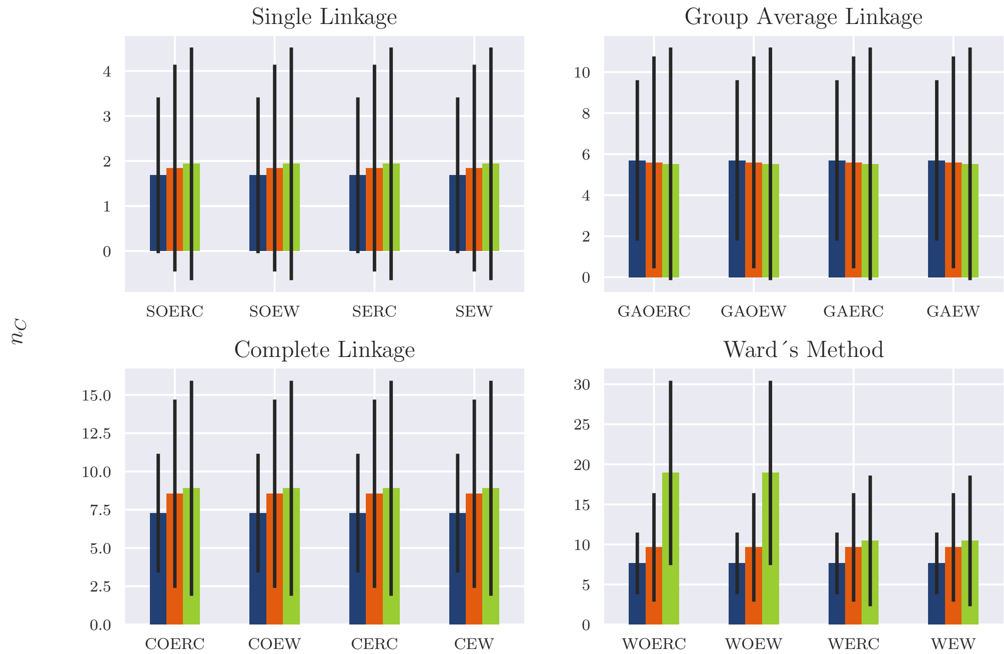
(a) Average cluster size n_C and s_{n_C} for different portfolio construction methods in Period 1, corresponding to the period between 01/03/1990 and 13/02/1998.



(b) Average cluster size n_C and s_{n_C} for different portfolio construction methods in Period 2, corresponding to the period between 16/02/1998 and 31/01/2006.



(c) Average cluster size n_C and s_{n_C} for different portfolio construction methods in Period 3, corresponding to the period between 01/02/2006 and 16/01/2014.



(d) Average cluster size n_C and s_{n_C} for different portfolio construction methods in Period 4, corresponding to the period between 17/01/2014 and 03/01/2022.

Figure 6.4: Average cluster size n_C for the different portfolio construction methods. The black line represents the standard deviation of s_{n_C} for every portfolio construction method. The blue, orange and green bars represents the results using time windows of size 100-, 300- and 500 days, respectively.

6.6 Summary of the Results

To summarize the results from the previous sections in this chapter, Table 6.2 illustrates the ranking of the evaluated allocation methods with respect to each investigated performance criteria. As shown, methods constructed using ERC yield lower Tracking Error compared to methods constructed using EW. In general, the opposite is true regarding Turnover. The average difference between the final number of clusters and the optimal number of clusters $\overline{\Delta k}$ seems to be most affected by the implemented linkage criteria. The same is true for $\overline{\mu_\kappa}$, however, the ranking is inverted. As seen in the results, the relationship between $\overline{\Delta k}$ and $\overline{n_C}$ holds for all allocation methods.

	TE	$\overline{TO}$	$\overline{\Delta k}$	$\overline{\mu_\kappa}$	$\overline{n_C}$
WEW	0.027	0.202	51.44	94.75	**17.06**
WOEW	0.027	0.203	53.2	94.75	**17.06**
WOERC	0.018	0.265	53.2	94.75	**17.06**
WERC	0.018	0.26	53.2	94.75	**17.06**
COEW	0.029	0.137	108.9	39.24	9.11
COERC	0.017	0.199	108.9	39.24	9.11
CEW	0.030	0.137	108.9	39.24	9.11
CERC	0.018	0.199	108.9	39.24	9.11
GAEW	0.030	0.110	158.70	29.27	5.95
GAOEW	0.029	0.114	165.86	29.27	5.95
GAERC	0.019	0.172	165.86	29.27	5.95
GAOERC	0.018	0.175	165.86	29.27	5.95
SEW	0.029	**0.042**	579.14	**21.06**	1.74
SOEW	0.026	0.068	579.14	**21.06**	1.74
SERC	0.016	0.192	579.14	**21.06**	1.74
SOERC	**0.013**	0.217	579.14	**21.06**	1.74

Table 6.2: Comparison of all of the evaluated allocation methods based on different performance measures. The bold numbers represent the best result for each performance measure. Note that the comparison is based on the averages over all time periods and time windows.

7
Discussion

This chapter discusses the empirical results obtained in Chapter 6 in order to provide insights into the performance of the hierarchical portfolio allocation algorithms. This discussion includes a comparison between the different portfolio construction methods as well as a broader analysis of the practical implications, strengths and weaknesses of the different hierarchical portfolio allocation algorithms. Finally, this chapter includes suggestions for future research.

The hierarchical portfolio allocation algorithms developed in this thesis is an extension and adaptation of the works by Markowitz (1952), López de Prado (2018), and Raffinot (2017; 2018). The methods in these papers have all been developed and shown prominence in a passive setting, but have not been examined in and adapted to an active management framework. Based on the results from this thesis, it can be argued that the proposed hierarchical portfolio allocation algorithm, which operates in an active management framework, is able to capture the existing hierarchical structure between assets, similarly to HRP, HCAA, and HERC (López de Prado 2018; Raffinot 2017; Raffinot 2018). In addition, based on the results of this thesis, the hierarchical portfolio allocation algorithms are robust and flexible since they yield consistent results over time and is able to adapt to different market events and conditions. The algorithms performed similarly over different time periods, suggesting that they can be used to create favorable sets of bets, irrespective of the market climate. The proposed hierarchical portfolio allocation algorithms enables the active investor to capture the hierarchical structure of different assets in a high-dimensional setting, while reducing inherent estimation errors resulting in bets that reflect the returns of a related benchmark effectively.

Single Linkage yielded the lowest Tracking Error amongst the set of linkage criteria, suggesting that when using this linkage method the allocation algorithm replicates the benchmark returns quite well. One of the reasons why this is the case may be that the average cluster size is less than two when Single Linkage is used, implying that many of the clusters contain only one asset, resulting in bets that are equal to zero for these singleton clusters since there is no other asset to hedge against. This entails that many of the portfolio weights are identical to the weights of the benchmark. Hence, few of the total bets are forced to equal one, creating a scenario where there are only a few bets which need to be hedged. In addition, there are also fewer assets that the forced bet can be hedged against. By extension, the small Tracking Error obtained from the Single Linkage algorithm suggests that the algorithm creates clusters of assets with almost perfect correlation, and therefore there does not exist a need for a large number of assets to hedge bets against.

The number of assets within a cluster is related to the condition number of the sample covariance matrix. Hence, since the average cluster size is smaller when employing Single Linkage compared to the remaining linkage criteria, the condition number for these clusters becomes smaller as well. This implies that the obtained sample covariance matrices from the clusters are more well-conditioned, which naturally leads to a more stable bet selection process. Also, the Turnover is smaller when Single Linkage is incorporated. Since the clusters are small and consist of highly correlated assets, a low Turnover becomes a natural result as long as the correlation between assets is stable over time. What also has to be mentioned is the fact that the difference between the final number of clusters and the optimal number of clusters suggested by GSI (Hastie et. al., 2001) is very high for allocation methods that employ Single Linkage. By extension, this implies that the obtained clusters do not generalize very well and overfit the data. However, allocation methods constructed using Single Linkage still result in a significantly smaller Tracking Error compared to linkage methods where the differences between the final number of clusters and the optimal number of clusters are smaller.

The difference in Tracking Error is small between methods incorporating Complete Linkage, Group Average Linkage, and Ward's Method. However, the use of Complete Linkage and Group Average Linkage results in a smaller Turnover compared to Ward's method. This suggests that the investors do not have to rebalance the portfolio as often while still obtaining a similar Tracking Error if Complete Linkage or Group Average Linkage is used. This can be explained by the size of clusters produced by the different linkage criteria, as it affects the size of condition number significantly. Ward's method produces on average the largest set of clusters, followed by Complete and Group Average Linkage. Consequently, the size of the condition number follows the same order, explaining why Ward's method produces bets that alternate more frequently compared to Complete and Group Average Linkage.

By incorporating the optimal leaf ordering algorithm in a subset of the different allocation algorithms, it is clear that its effect on the performance was marginal at best. The ordering of the leaves in the hierarchical structure does not seem to have an impact on the different performance measures over time. One plausible explanation for this phenomenon is that, on average, very similar hierarchical structures are constructed, irrespective if optimal leaf ordering is implemented or not. Naturally, if the constructed structures are similar or identical, it is expected that the hierarchical structures produce indistinguishable results. This by extension, entails that the ordering of the input data is near-optimal or optimal, creating a scenario where the algorithm proposed by Bar-Joseph et al. (2001) is non-crucial for the results. However, this might not have been the case for all of the other 2^{p-1} orders that the input data could have had. In addition, it is important to highlight the size of the data in this thesis which has an averaging effect on the results. If one conducted a more granular analysis with a smaller dataset, the effects from optimal leaf ordering could be more distinguishable.

The results display the importance of choosing the appropriate between-cluster allocation method. In this thesis, the results from implementing equal risk contribution (Maillard et. al., 2010) and equal weighting were investigated. Evidently, when considering Tracking Error, equal risk contribution performed substantially better than equal weighting across all linkage methods and time windows. To understand and explain this substantial outperformance, one can view the benchmark as a weighted sum of assets with varying levels of inherent risk. Naturally, the benchmark encompasses assets that are riskier than other assets in the benchmark, and if one assumes that risky assets exhibit large correlations amongst themselves than to less risky assets, one can view the benchmark as an average of a set of clusters containing assets that are similar in risk profile. This is very closely related to the notion of equal risk contribution. Therefore, if the clusters from the hierarchical clustering are similar to the groupings of assets in the benchmark from a risk perspective, it is reasonable to suggest that the equal risk contribution method constructs bets that reflect the benchmark very well.

This is in contrast to equal weighting, which naively allocates weights between the clusters. If one assumes that risky assets drive the direction of the benchmark, the obtained results from allocation algorithms using equal weighting indicate that this between-cluster allocation method allocates insufficient amounts of weight to these driving assets. Conversely, this holds also if the opposite scenario is true, i.e. when assets of lower risk dictate the direction of the benchmark. The conclusion is that portfolios that incorporate equal weighting in this thesis either allocate too much or too little capital to the clusters of assets that determine the direction of the benchmark, resulting in greater Tracking Error compared to allocation methods that incorporate equal risk contribution.

However, if one assesses equal risk contribution and equal weighting based on Turnover, it is evident that equal weighting creates considerably more stable bets over time compared to allocation algorithms constructed using equal risk contribution. Primarily, the equal weighting allocation method does not require the estimation of the sample covariance matrix for every cluster, since it scales cluster bets evenly. On the contrary, equal risk contribution requires the sample covariance matrix to determine the risk contribution from each cluster. Hence, the induced errors from the estimation of the covariance matrix may create very sensitive and numerically unstable bets that alternate frequently over time. This is in line with the arguments of DeMiguel et. al. (2009), who argue that portfolio optimization methods that are dependent on the estimated covariance matrix yield unstable portfolio weights due to the inferred estimation errors.

Another interesting aspect of the results is that the time window used to compute the sample correlation and covariance matrices does not seem to have a significant impact on the obtained Tracking Error for any of the allocation algorithms. As suspected, the allocation methods perform on average better when more data is used as input, but the difference is lower than expected. Even though the condition numbers show an inverted relationship with the length of the time window, they do not differ considerably. This implies that the stability of the obtained covariance matrices does not depend significantly on the time window. Hence, it is natural to assume that the Tracking Error does not depend on it either. One may assume that the small difference in condition number for different time windows is a result of the cluster selection process. The clusters are partitioned if their associated covariance matrix is not invertible, leading to a smaller ratio between the number of assets and number of observations. If many partitions are made, which has been seen in the results, the length of the time window becomes insignificant since the number of assets already is small.

As previously mentioned, the number of assets in a cluster seems to have an impact on the Tracking Error, and an even more significant impact on the Turnover. The reason for this is clear, since the obtained sample covariance matrix of a cluster is usually more well-conditioned when the concentration ratio is small, as previously outlined in Section 2.3. Hence, a small number of assets decreases the concentration ratio when the number of observations grows, yielding more numerically stable sample covariance matrices that in turn affect Tracking Error and Turnover positively, which is supported in the obtained results.

The primary objective of the evaluated allocation methods is to group assets that exhibit high correlations amongst themselves into clusters. Then, these clusters are partitioned into sub-clusters containing assets with an even greater intracluster correlation. This process is performed recursively until the corresponding sample covariance matrix is invertible. As discussed in Section 2.3, the correlation among assets within a portfolio causes signal-induced instability in the corresponding covariance matrix. Hence, the procedure of grouping highly correlated assets to obtain an invertible covariance matrix is problematic since signal-induced instability can not be decreased by increasing the number of observations when constructing the covariance matrix. This might explain the large discrepancy between the optimal number of clusters suggested by GSI and the final number of clusters.

Finally, the number of small clusters or singleton sets does not seem to be an issue when the objective is to minimize the Tracking Error. Although, if additional constraints are added to the portfolio optimization problem, one might need to find clusters of larger size. Another issue that occurs partly due to singular covariance matrices is the construction of singleton clusters. Since one is not able to place a bet in singleton clusters, this drastically decreases the range of bet options for the active portfolio manager. In order to increase the options, and also to increase the cluster size in general, the evaluated hierarchical portfolio algorithms need to be developed further.

7.1 Future Research

There exist several interesting areas of research in the intersection of hierarchical clustering, portfolio optimization, and active management that can be explored further. A few of these are discussed in this section. First and foremost, the main issue with the purposed hierarchical portfolio allocation algorithm is that the number of obtained clusters is disproportionate in relation to the optimal number of clusters suggested by GSI. We see mainly two ways to handle this. Firstly, an interesting area of research is the combination of shrinkage (Ledoit and Wolf, 2004) and hierarchical clustering, where different variations of shrinkage methods can be used to shrink the covariance matrices of different clusters to obtain more well-conditioned estimates.

Secondly, a relevant part of future research is to develop allocation algorithms where the sample covariance matrix does not need to be inverted. This has already been achieved in a passive setting, for example HRP, HCAA, and HERC (López de Prado 2018; Raffinot 2017; Raffinot 2018). Hence, it should also be feasible to accomplish this in an active setting. The instability and the small cluster size that some of the allocation algorithms experience mainly depend on the singularity of the covariance matrix. Therefore, it is of great importance to evaluate if the hierarchical structure of the assets can be used to a greater extent, instead of solely being used for inversion of the covariance matrix in order to place bets in an active management framework.

Moreover, the scope of different between-cluster allocation methods can be broadened to investigate the scaling of bets, as there existed a distinct discrepancy in the results of this thesis. For example, Maximum Diversification (Choueifaty and Coignard, 2008), as well as other variations of Equal Risk Contribution should be evaluated. Also, in this thesis, the bet associated with the asset which showed the best performance the last week was forced to equal one. The subject of bet-hedging within clusters is a very interesting area and can be extended further by incorporating other, more sophisticated methods to conduct the hedging of bets. This particular subject has been studied extensively in other types of investment settings. However, the hedging strategy may be of great importance and should be evaluated further in an active management setting where the hierarchical structures of the assets are taken into account. Further, the results regarding hedging strategies may depend on the choice of linkage criterion, and hence, this is something that needs to be evaluated further.

Lastly, other performance measures beyond those provided in this thesis can be included to widen the notion of performance. For example, not all active funds may define risk as the variance of the excess return. An interesting area to explore further is how the incorporation of hierarchical structures affects the value at risk within active funds. Since the objective of this thesis was to minimize the variance of the excess return, the expected excess return has been left out of the discussion. Hence, this is another interesting subject of future research that may be explored further.

8
Conclusion

The aim of this thesis was to construct hierarchical portfolio allocation algorithms based on the works of Markowitz (1952), López de Prado (2018), and Raffinot (2017; 2018) which are adapted to an active management framework. The hierarchical portfolio allocation algorithms performs satisfactorily in an active management setting where the objective is to minimize the variance of the excess return and were capable of yielding consistent results over different time periods. From an active management perspective, the proposed algorithms are well-adapted since they are able to replicate benchmark returns without reproducing the associated benchmark weights. The proposed hierarchical portfolio allocation algorithms allows an active investor to capture the hierarchical structure of various assets in a high-dimensional setting while minimizing inherent estimation errors, resulting in bets that effectively reflect a related benchmark.

To contribute to the existing literature, several different portfolio allocation methods were constructed and evaluated. Specifically, allocation methods incorporating Single Linkage outperformed other types of methods with respect to Tracking Error. These methods construct smaller clusters of highly correlated assets, which alleviates the process of hedging bets, leading to small Tracking Errors. In addition, the usage of Equal Risk Contribution for between-cluster allocation decreases the Tracking Error significantly. This leads to the conclusion that the combination of these methods yields the best results for replication of the benchmark returns.

Depending on the objectives and constraints of the active investor, there might exist advantages in selecting an appropriate linkage criterion that produces fewer, but larger clusters. Consequently, these criteria increase the options of selecting which bets to hedge as well as to reduce overfitting of the portfolio allocation methods. Allocation algorithms that incorporate Ward's method, Group Average Linkage or Complete Linkage generate Tracking Errors that are comparable with Single Linkage, but have the advantage of producing larger clusters. Apparently, there exists a trade-off between cluster size and Turnover, something that has to be taken into account by the active investor.

An interesting aspect of this thesis is the efficiency of Equal Risk Contribution as a between-cluster allocation method for reducing the Tracking Error. The results from this thesis indicate that a benchmark might be viewed as a set of clusters with dissimilar levels of risk that alternate over time, which in turn drives the direction of the benchmark to varying degrees. This thesis shows that the combination of hierarchical clustering and Equal Risk Contribution captures this notion of the benchmark structure very well. Hence, it may be used to allocate capital efficiently with the objective to decrease the portfolio risk with respect to the benchmark.

There exists several areas of research which should be evaluated further in order to construct efficient portfolio allocation methods in an active management framework. Examples of such areas are the combination of hierarchical clustering and different shrinkage methods. Bet selection algorithms that are independent of the inversion of the covariance matrix are highly desirable in an active management setting, and constitute an interesting subject for future research.

In conclusion, it can be established that employing hierarchical clustering in an active management framework with the purpose of replicating the returns of a related benchmark is achievable, and in fact produces satisfactory and robust results. Several different compositions of the proposed hierarchical portfolio allocation algorithm can be considered. It has been demonstrated that the combination of Single Linkage and Equal Risk Contribution may be the most advantageous choice.

Bibliography

[1] Alfelt, G. (2021). Modeling the covariance matrix of financial asset returns (Doctoral dissertation, Department of Mathematics, Stockholm University).

[2] Asness, C. S., Frazzini, A., and Pedersen, L. H. (2012). Leverage aversion and risk parity. Financial Analysts Journal, 68(1), 47-59.

[3] Bar-Joseph, Z., Gifford, D. K., and Jaakkola, T. S. (2001). Fast optimal leaf ordering for hierarchical clustering. Bioinformatics, 17(suppl_1), S22-S29.

[4] Belsley, D. A., Kuh, E., and Welsch, R. E. (1980). The condition number. Regression diagnostics: Identifying influential data and sources of collinearity, 100, 104.

[5] Billio, M., Getmansky, M., Lo, A. W., and Pelizzon, L. (2012). Econometric measures of connectedness and systemic risk in the finance and insurance sectors. Journal of financial economics, 104(3), 535-559.

[6] Choueifaty, Y., and Coignard, Y. (2008). Toward maximum diversification. The Journal of Portfolio Management, 35(1), 40-51.

[7] DeMiguel, V., Garlappi, L., and Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy?. The review of Financial studies, 22(5), 1915-1953.

[8] De Prado, M. L. (2018). Advances in financial machine learning. John Wiley and Sons.

[9] De Prado, M. M. L. (2020). Machine learning for asset managers. Cambridge University Press.

[10] Ernst, P., Thompson, J., and Miao, Y. (2016). Portfolio selection: the power of equal weight. arXiv preprint arXiv:1602.00782.

[11] Fan, J., Fan, Y., and Lv, J. (2008). High dimensional covariance matrix estimation using a factor model. Journal of Econometrics, 147(1), 186-197.

[12] Grinold, R. C. (1989). The fundamental law of active management. Streetwise, The Best of the Journal of Portfolio Management, 161-8.

[13] Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: Springer.

[14] Jagannathan, R., and Ma, T. (2003). Risk reduction in large portfolios: Why imposing the wrong constraints helps. The Journal of Finance, 58(4), 1651-1683.

[15] Jain, A. K., and Dubes, R. C. (1988). Algorithms for clustering data. Prentice-Hall, Inc

[16] Kolm, P. N., Tütüncü, R., and Fabozzi, F. J. (2014). 60 Years of portfolio optimization: Practical challenges and current trends. European Journal of Operational Research, 234(2), 356-371.

[17] Ledoit, O., and Wolf, M. (2004). Honey, I shrunk the sample covariance matrix. The Journal of Portfolio Management, 30(4), 110-119.

[18] Levina, E., Rothman, A., and Zhu, J. (2008). Sparse estimation of large covariance matrices via a nested lasso penalty. The Annals of Applied Statistics, 2(1), 245-263.

[19] Lloyd, S. (1982). Least squares quantization in PCM. IEEE transactions on information theory, 28(2), 129-137. ISO 690

[20] Maillard, S., Roncalli, T., and Teïletche, J. (2010). The properties of equally weighted risk contribution portfolios. The Journal of Portfolio Management, 36(4), 60-70.

[21] Markowitz, H. (1952). Portfolio Selection. Journal of Finance, 7, 7791

[22] Michaud, R. O. (1989). The Markowitz optimization enigma: Is ‘optimized’ optimal?. Financial analysts journal, 45(1), 31-42.

[23] Nielsen, F. (2016). Introduction to HPC with MPI for Data Science. Springer.

[24] Papenbrock, J. (2011). Asset Clusters and Asset Networks in Financial Risk Management and Portfolio Optimization (Doctoral dissertation, Dissertation, Karlsruhe, Karlsruher Institut für Technologie (KIT), 2011).

[25] Perrin, S., and Roncalli, T. (2020). Machine learning optimization algorithms and portfolio allocation. Machine Learning for Asset Management: New Developments and Financial Applications, 261-328.

[26] Qian, E. (2005a). Risk parity portfolios: Efficient portfolios through true diversification. Panagora Asset Management.

[27] Qian, E. E. (2005b). On the financial interpretation of risk contribution: Risk budgets do add up. Available at SSRN 684221.

[28] Raffinot, T. (2017). Hierarchical clustering-based asset allocation. The Journal of Portfolio Management, 44(2), 89-99.

[29] Raffinot, T. (2018). The hierarchical equal risk contribution portfolio. Available at SSRN 3237540.

[30] Simon, H. A. (1962). The architecture of complexity. In Facets of systems science (pp. 457-476). Springer, Boston, MA.

[31] Bodnar, T. Parolya, N. Schmid, W.(2018) Estimation of the global minimum variance portfolio in high dimensions, European Journal of Operational Research, Volume 266, Issue 1, Pages 371-390,

[32] Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. Journal of the American statistical association, 58(301), 236-244.

[33] Yue, S., Wang, X., and Wei, M. (2008). Application of two-order difference to gap statistic. Transactions of Tianjin University, 14(3), 217-221.

A

Appendix A

A.1 Period 1: 01/03/1990 – 13/02/1998

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0112	0.2512	0.1273	686.4599	31.5095	3.2058	663.4582	27.8816	70.0516	1.6059	1.7359
SOEW	0.0285	0.0410	0.0074	686.4599	31.5095	3.2058	663.4582	27.8816	70.0516	1.6059	1.7359
SERC	0.0112	0.2512	0.1273	686.4599	31.5095	3.2058	663.4582	27.8816	70.0516	1.6059	1.7359
SEW	0.0282	0.0408	0.0073	686.4599	31.5095	3.2058	663.4582	27.8816	70.0516	1.6059	1.7359
GAOERC	0.0155	0.1999	0.0934	177.7414	20.4636	2.5793	1019.1992	45.5487	101.8722	5.9783	3.9476
GAOEW	0.0275	0.1148	0.0083	177.7414	20.4636	2.5793	1019.1992	45.5487	101.8722	5.9783	3.9476
GAERC	0.0155	0.1999	0.0934	177.7414	20.4636	2.5793	1019.1992	45.5487	101.8722	5.9783	3.9476
GAEW	0.0299	0.1147	0.0083	177.7414	20.4636	2.5793	1019.1992	45.5487	101.8722	5.9783	3.9476
COERC	0.0158	0.2218	0.1022	141.9650	15.4501	3.2586	945.6940	54.4482	103.9642	7.5905	3.9796
COEW	0.0268	0.1336	0.0088	141.9650	15.4501	3.2586	945.6940	54.4482	103.9642	7.5905	3.9796
CERC	0.0158	0.2218	0.1022	141.9650	15.4501	3.2586	945.6940	54.4482	103.9642	7.5905	3.9796
CEW	0.0301	0.1331	0.0089	141.9650	15.4501	3.2586	945.6940	54.4482	103.9642	7.5905	3.9796
WOERC	0.0164	0.2777	0.1085	77.1547	10.9443	6.1746	1588.6985	134.9067	217.7241	12.6805	6.0366
WOEW	0.0250	0.2005	0.0154	77.1547	10.9443	6.1746	1588.6985	134.9067	217.7241	12.6805	6.0366
WERC	0.0164	0.2777	0.1085	77.1547	10.9443	6.1746	1588.6985	134.9067	217.7241	12.6805	6.0366
WEW	0.0250	0.2005	0.0154	77.1547	10.9443	6.1746	1588.6985	134.9067	217.7241	12.6805	6.0366

Table A.1: Comparison of performance criteria between the different allocation algorithms with rolling time window $n = 100$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0105	0.2827	0.1624	657.4091	35.5705	2.2263	532.4246	22.5256	59.3051	1.6752	2.4989
SOEW	0.0284	0.0379	0.0112	657.4091	35.5705	2.2263	532.4246	22.5256	59.3051	1.6752	2.4989
SERC	0.0105	0.2827	0.1624	657.4091	35.5705	2.2263	532.4246	22.5256	59.3051	1.6752	2.4989
SEW	0.0288	0.0380	0.0111	657.4091	35.5705	2.2263	532.4246	22.5256	59.3051	1.6752	2.4989
GAOERC	0.0139	0.2118	0.1256	191.2319	20.7474	1.7701	767.9036	25.4825	73.4198	5.4827	5.5060
GAOEW	0.0270	0.0983	0.0073	191.2319	20.7474	1.7701	767.9036	25.4825	73.4198	5.4827	5.5060
GAERC	0.0139	0.2118	0.1256	191.2319	20.7474	1.7701	767.9036	25.4825	73.4198	5.4827	5.5060
GAEW	0.0291	0.0987	0.0074	191.2319	20.7474	1.7701	767.9036	25.4825	73.4198	5.4827	5.5060
COERC	0.0148	0.2334	0.1227	118.7878	12.5902	2.2047	722.0696	37.9729	84.2618	9.0893	6.5749
COEW	0.0268	0.1286	0.0088	118.7878	12.5902	2.2047	722.0696	37.9729	84.2618	9.0893	6.5749
CERC	0.0148	0.2334	0.1227	118.7968	12.5853	2.2047	722.0696	37.9730	84.2619	9.0893	6.5749
CEW	0.0291	0.1291	0.0088	118.7968	12.5853	2.2047	722.0696	37.9730	84.2619	9.0893	6.5749
WOERC	0.0145	0.2939	0.1476	56.1682	9.4545	4.9892	1029.0766	84.1672	150.3261	16.6863	10.4451
WOEW	0.0257	0.1860	0.0141	56.1682	9.4545	4.9892	1029.0766	84.1672	150.3261	16.6863	10.4451
WERC	0.0145	0.2939	0.1476	56.1682	9.4545	4.9892	1029.0766	84.1672	150.3261	16.6863	10.4451
WEW	0.0257	0.1860	0.0141	56.1682	9.4545	4.9892	1029.0766	84.1672	150.3261	16.6863	10.4451

Table A.2: Comparison of performance criteria between the different allocation algorithms with rolling time window $n = 300$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0104	0.2805	0.1620	624.6151	37.7266	2.0242	488.5229	22.3009	57.2679	1.7628	3.0103
SOEW	0.0277	0.0359	0.0120	624.6151	37.7266	2.0242	488.5229	22.3009	57.2679	1.7628	3.0103
SERC	0.0104	0.2805	0.1620	624.6151	37.7266	2.0242	488.5229	22.3009	57.2679	1.7628	3.0103
SEW	0.0277	0.0359	0.0120	624.6151	37.7266	2.0242	488.5229	22.3009	57.2679	1.7628	3.0103
GAOERC	0.0139	0.2120	0.1353	203.5961	18.9505	1.5823	692.8282	22.1884	66.9814	5.1929	6.1881
GAOEW	0.0278	0.0938	0.0070	203.5961	18.9505	1.5823	692.8282	22.1884	66.9814	5.1929	6.1881
GAERC	0.0139	0.2120	0.1353	203.5961	18.9505	1.5823	692.8282	22.1884	66.9814	5.1929	6.1881
GAEW	0.0278	0.0938	0.0070	203.5961	18.9505	1.5823	692.8282	22.1884	66.9814	5.1929	6.1881
COERC	0.0143	0.2206	0.1021	112.5604	13.7651	1.9311	702.9099	34.1331	81.0805	9.2992	7.6706
COEW	0.0259	0.1244	0.0081	112.5604	13.7651	1.9311	702.9099	34.1331	81.0805	9.2992	7.6706
CERC	0.0143	0.2206	0.1021	112.5604	13.7651	1.9311	702.9099	34.1331	81.0805	9.2992	7.6706
CEW	0.0280	0.1253	0.0083	112.5604	13.7651	1.9311	702.9099	34.1331	81.0805	9.2992	7.6706
WOERC	0.0145	0.2856	0.1383	49.0541	10.7789	4.4062	888.8324	72.0349	131.2460	17.7602	12.3118
WOEW	0.0261	0.1768	0.0136	49.0541	10.7789	4.4062	888.8324	72.0349	131.2460	17.7602	12.3118
WERC	0.0145	0.2856	0.1383	49.0541	10.7789	4.4062	888.8324	72.0349	131.2460	17.7602	12.3118
WEW	0.0261	0.1768	0.0136	49.0541	10.7789	4.4062	888.8324	72.0349	131.2460	17.7602	12.3118

Table A.3: Comparison of performance criteria between the different allocation algorithms with rolling time window $n = 500$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

A.2 Period 2: 16/02/1998 – 31/01/2006

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0170	0.2687	0.1650	394.7404	20.9999	3.0070	361.4693	21.3822	46.9112	1.6458	1.8193
SOEW	0.0390	0.0532	0.0119	394.7404	20.9999	3.0070	361.4693	21.3822	46.9112	1.6458	1.8193
SERC	0.0182	0.2655	0.1611	394.7404	20.9999	3.0070	361.4693	21.3822	46.9112	1.6458	1.8193
SEW	0.0385	0.0534	0.0119	394.7404	20.9999	3.0070	361.4693	21.3822	46.9112	1.6458	1.8193
GAOERC	0.0257	0.2175	0.0899	85.9584	11.5655	2.6702	599.8954	39.1119	73.4217	6.8569	4.2163
GAOEW	0.0379	0.1567	0.0113	85.9584	11.5655	2.6702	599.8954	39.1119	73.4217	6.8569	4.2163
GAERC	0.0261	0.2129	0.0858	85.9584	11.5655	2.6702	599.8954	39.1119	73.4217	6.8569	4.2163
GAEW	0.0394	0.1565	0.0116	85.9584	11.5655	2.6702	599.8954	39.1119	73.4217	6.8569	4.2163
COERC	0.0258	0.2386	0.0841	68.3367	9.8136	4.0673	605.6669	47.5589	78.7811	8.8119	4.0904
COEW	0.0372	0.1830	0.0118	68.3367	9.8136	4.0673	605.6669	47.5589	78.7811	8.8119	4.0904
CERC	0.0267	0.2377	0.0887	68.3367	9.8136	4.0673	605.6669	47.5589	78.7811	8.8119	4.0904
CEW	0.0393	0.1825	0.0116	68.3367	9.8136	4.0673	605.6669	47.5589	78.7811	8.8119	4.0904
WOERC	0.0263	0.3204	0.0897	36.2165	7.8220	13.6357	1033.3004	119.7025	167.5821	14.8403	5.5144
WOEW	0.0355	0.2750	0.0204	36.2165	7.8220	13.6357	1033.3004	119.7025	167.5821	14.8403	5.5144
WERC	0.0263	0.3204	0.0897	36.2165	7.8220	13.6357	1033.3004	119.7025	167.5821	14.8403	5.5144
WEW	0.0355	0.2750	0.0204	36.2165	7.8220	13.6357	1033.3004	119.7025	167.5821	14.8403	5.5144

Table A.4: Comparison of performance criteria between the different allocation algorithms with rolling time window $n = 100$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0180	0.2569	0.1741	377.7077	19.6117	2.1730	243.0138	15.1698	33.9446	1.7162	2.4703
SOEW	0.0361	0.0459	0.0160	377.7077	19.6117	2.1730	243.0138	15.1698	33.9446	1.7162	2.4703
SERC	0.0178	0.2673	0.1911	377.7077	19.6117	2.1730	243.0138	15.1698	33.9446	1.7162	2.4703
SEW	0.0368	0.0461	0.0160	377.7077	19.6117	2.1730	243.0138	15.1698	33.9446	1.7162	2.4703
GAOERC	0.0250	0.1917	0.0837	85.8307	12.2889	1.7789	400.8863	22.1195	50.0319	6.9070	6.3218
GAOEW	0.0363	0.1388	0.0122	85.8307	12.2889	1.7789	400.8863	22.1195	50.0319	6.9070	6.3218
GAERC	0.0254	0.2028	0.1018	85.8307	12.2889	1.7789	400.8863	22.1195	50.0319	6.9070	6.3218
GAEW	0.0369	0.1398	0.0123	0.0000	0.0000	1.7789	400.8863	22.1195	50.0319	6.9070	6.3218
COERC	0.0241	0.2373	0.0990	53.5542	9.6501	2.9102	448.6915	33.4650	63.8859	11.3388	7.0330
COEW	0.0347	0.1808	0.0138	53.5542	9.6501	2.9102	448.6915	33.4650	63.8859	11.3388	7.0330
CERC	0.0248	0.2410	0.1132	53.5480	9.6579	2.9102	448.6915	33.4644	63.8854	11.3386	7.0326
CEW	0.0350	0.1810	0.0143	53.5480	9.6579	2.9102	448.6915	33.4644	63.8854	11.3386	7.0326
WOERC	0.0250	0.3205	0.1201	25.0948	6.9322	11.0631	638.4841	75.2080	114.4576	20.6368	9.9483
WOEW	0.0344	0.2621	0.0220	25.0948	6.9322	11.0631	638.4841	75.2080	114.4576	20.6368	9.9483
WERC	0.0250	0.3205	0.1201	25.0948	6.9322	11.0631	638.4841	75.2080	114.4576	20.6368	9.9483
WEW	0.0344	0.2621	0.0220	25.0948	6.9322	11.0631	638.4841	75.2080	114.4576	20.6368	9.9483

Table A.5: Comparison of performance criteria between the different allocation algorithms with rolling time window $n = 300$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0181	0.2501	0.1746	362.4421	17.2370	1.9736	195.6007	13.5306	28.2188	1.7815	2.7040
SOEW	0.0354	0.0423	0.0169	362.4421	17.2370	1.9736	195.6007	13.5306	28.2188	1.7815	2.7040
SERC	0.0186	0.2543	0.1788	362.4421	17.2370	1.9736	195.6007	13.5306	28.2188	1.7815	2.7040
SEW	0.0358	0.0428	0.0163	362.4421	17.2370	1.9736	195.6007	13.5306	28.2188	1.7815	2.7040
GAOERC	0.0239	0.1968	0.0957	85.2729	12.2003	1.5912	338.5041	19.4642	44.3969	6.9882	7.4724
GAOEW	0.0350	0.1344	0.0138	85.2729	12.2003	1.5912	338.5041	19.4642	44.3969	6.9882	7.4724
GAERC	0.0250	0.2000	0.1074	85.2729	12.2003	1.5912	338.5041	19.4642	44.3969	6.9882	7.4724
GAEW	0.0366	0.1354	0.0138	85.2729	12.2003	1.5912	338.5041	19.4642	44.3969	6.9882	7.4724
COERC	0.0242	0.2353	0.0988	50.0611	9.0555	2.5215	399.3264	30.3977	59.1443	12.1795	8.6026
COEW	0.0346	0.1774	0.0145	50.0611	9.0555	2.5215	399.3264	30.3977	59.1443	12.1795	8.6026
CERC	0.0251	0.2375	0.1094	50.0611	9.0555	2.5215	399.3264	30.3977	59.1443	12.1795	8.6026
CEW	0.0353	0.1782	0.0144	50.0611	9.0555	2.5215	399.3264	30.3977	59.1443	12.1795	8.6026
WOERC	0.0244	0.3197	0.1276	21.2640	6.5474	9.9376	563.7267	68.3723	105.2995	23.2631	12.3704
WOEW	0.0337	0.2556	0.0228	21.2640	6.5474	9.9376	563.7267	68.3723	105.2995	23.2631	12.3704
WERC	0.0244	0.3197	0.1276	21.2640	6.5474	9.9376	563.7267	68.3723	105.2995	23.2631	12.3704
WEW	0.0337	0.2556	0.0228	0.0000	0.0000	9.9376	563.7267	68.3723	105.2995	23.2631	12.3704

Table A.6: Comparison of performance criteria between the different allocation algorithms with rolling time window $n = 500$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

A.3 Period 3: 01/02/2006 – 16/01/2014

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0115	0.1978	0.1539	734.1861	32.3504	3.2322	593.0827	27.0946	62.2827	1.6229	1.7505
SERC	0.0115	0.1978	0.1539	734.1861	32.3504	3.2322	593.0827	27.0946	62.2827	1.6229	1.7505
SOEW	0.0250	0.0408	0.0065	734.1861	32.3504	3.2322	593.0827	27.0946	62.2827	1.6229	1.7505
SEW	0.0250	0.0408	0.0065	734.1861	32.3504	3.2322	593.0827	27.0946	62.2827	1.6229	1.7505
GAOERC	0.0161	0.1676	0.1014	192.3291	22.5604	2.5948	760.7436	41.8450	76.0582	5.9418	3.9370
GAOEW	0.0264	0.1067	0.0074	192.3291	22.5604	2.5948	760.7436	41.8450	76.0582	5.9418	3.9370
GAERC	0.0161	0.1676	0.1014	192.3291	22.5604	2.5948	760.7436	41.8450	76.0582	5.9418	3.9370
GAEW	0.0264	0.1067	0.0074	192.3291	22.5604	2.5948	760.7436	41.8450	76.0582	5.9418	3.9370
COERC	0.0157	0.1827	0.1020	148.8727	15.0907	3.3308	747.4482	50.3431	79.8743	7.6284	3.9010
COEW	0.0262	0.1241	0.0070	148.8727	15.0907	3.3308	747.4482	50.3431	79.8743	7.6284	3.9010
CERC	0.0157	0.1827	0.1020	148.8727	15.0907	3.3308	747.4482	50.3431	79.8743	7.6284	3.9010
CEW	0.0262	0.1241	0.0070	148.8727	15.0907	3.3308	747.4482	50.3431	79.8743	7.6284	3.9010
WOERC	0.0164	0.2368	0.0930	80.5568	11.2095	10.7235	1445.8178	134.2199	189.7557	13.2728	5.4246
WOEW	0.0241	0.1881	0.0124	80.5568	11.2095	10.7235	1445.8178	134.2199	189.7557	13.2728	5.4246
WERC	0.0164	0.2368	0.0930	80.5568	11.2095	10.7235	1445.8178	134.2199	189.7557	13.2728	5.4246
WEW	0.0241	0.1881	0.0124	80.5568	11.2095	10.7235	1445.8178	134.2199	189.7557	13.2728	5.4246

Table A.7: Comparison of performance criteria between the different allocation algorithms for the third period with rolling time window $n = 100$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0109	0.1904	0.1298	672.2856	27.0229	2.3143	340.0514	18.9807	39.6970	1.7674	2.5602
SERC	0.0109	0.1904	0.1298	672.2856	27.0229	2.3143	340.0514	18.9807	39.6970	1.7674	2.5602
SOEW	0.0235	0.0390	0.0077	672.2856	27.0229	2.3143	340.0514	18.9807	39.6970	1.7674	2.5602
SEW	0.0235	0.0390	0.0077	672.2856	27.0229	2.3143	340.0514	18.9807	39.6970	1.7674	2.5602
GAOERC	0.0159	0.1440	0.0638	202.3211	21.1354	1.7588	437.3867	24.0775	47.1725	5.6858	5.5338
GAOEW	0.0258	0.0930	0.0069	202.3211	21.1354	1.7588	437.3867	24.0775	47.1725	5.6858	5.5338
GAERC	0.0159	0.1440	0.0638	202.3211	21.1354	1.7588	437.3867	24.0775	47.1725	5.6858	5.5338
GAEW	0.0258	0.0930	0.0069	202.3211	21.1354	1.7588	437.3867	24.0775	47.1725	5.6858	5.5338
COERC	0.0153	0.1694	0.0698	122.5734	13.4356	2.2700	478.6749	34.5154	56.5311	9.1489	6.3887
COEW	0.0253	0.1188	0.0067	122.5734	13.4356	2.2700	478.6749	34.5154	56.5311	9.1489	6.3887
CERC	0.0153	0.1694	0.0698	122.5734	13.4356	2.2700	478.6749	34.5154	56.5311	9.1489	6.3887
CEW	0.0253	0.1188	0.0067	122.5734	13.4356	2.2700	478.6749	34.5154	56.5311	9.1489	6.3887
WOERC	0.0150	0.2277	0.0972	61.2624	8.8925	7.3936	774.4973	82.1655	113.0407	17.4239	9.4131
WOEW	0.0225	0.1731	0.0109	61.2624	8.8925	7.3936	774.4973	82.1655	113.0407	17.4239	9.4131
WERC	0.0150	0.2277	0.0972	61.2624	8.8925	7.3936	774.4973	82.1655	113.0407	17.4239	9.4131
WEW	0.0225	0.1731	0.0109	61.2624	8.8925	7.3936	774.4973	82.1655	113.0407	17.4239	9.4131

Table A.8: Comparison of performance criteria between the different allocation algorithms with rolling time window $n = 300$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0115	0.1842	0.1343	633.8779	33.4172	2.1282	288.5553	17.4937	35.0073	1.8756	3.0200
SOEW	0.0229	0.0401	0.0084	633.8779	33.4172	2.1282	288.5553	17.4937	35.0073	1.8756	3.0200
SERC	0.0115	0.1842	0.1343	633.8779	33.4172	2.1282	288.5553	17.4937	35.0073	1.8756	3.0200
SEW	0.0229	0.0401	0.0084	633.8779	33.4172	2.1282	288.5553	17.4937	35.0073	1.8756	3.0200
GAOERC	0.0165	0.1397	0.0608	207.5840	16.7790	1.5861	343.2316	21.4727	40.5384	5.5364	6.1382
GAERC	0.0165	0.1397	0.0608	207.5840	16.7790	1.5861	343.2316	21.4727	40.5384	5.5364	6.1382
GAOEW	0.0263	0.0894	0.0067	207.5840	16.7790	1.5861	343.2316	21.4727	40.5384	5.5364	6.1382
GAEW	0.0263	0.0894	0.0067	207.5840	16.7790	1.5861	343.2316	21.4727	40.5384	5.5364	6.1382
COERC	0.0157	0.1640	0.0722	118.0032	11.7101	1.9995	455.6906	31.6768	54.3037	9.5373	7.3063
COEW	0.0253	0.1159	0.0069	118.0032	11.7101	1.9995	455.6906	31.6768	54.3037	9.5373	7.3063
CERC	0.0157	0.1640	0.0722	118.0032	11.7101	1.9995	455.6906	31.6768	54.3037	9.5373	7.3063
CEW	0.0253	0.1159	0.0069	118.0032	11.7101	1.9995	455.6906	31.6768	54.3037	9.5373	7.3063
WOERC	0.0154	0.2175	0.0901	57.271	7.863	6.7821	603.8029	70.9113	90.7895	18.8961	10.9201
WOEW	0.0233	0.1663	0.0113	57.271	7.863	6.7821	603.8029	70.9113	90.7895	18.8961	10.9201
WERC	0.0154	0.2175	0.0901	57.271	7.863	6.7821	603.8029	70.9113	90.7895	18.8961	10.9201
WEW	0.0233	0.1663	0.0113	57.271	7.863	6.7821	603.8029	70.9113	90.7895	18.8961	10.9201

Table A.9: Comparison of performance criteria between the different allocation algorithms with rolling time window $n = 500$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

A.4 Period 4: 17/01/2014 – 03/01/2022

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0141	0.1530	0.0303	651.1369	30.6713	3.2169	800.4378	28.4708	73.7613	1.6812	1.7314
SOEW	0.0276	0.0431	0.0063	651.1369	30.6713	3.2169	800.4378	28.4708	73.7613	1.6812	1.7314
SERC	0.0141	0.1530	0.0303	651.1369	30.6713	3.2169	800.4378	28.4708	73.7613	1.6812	1.7314
SEW	0.0276	0.0431	0.0063	651.1369	30.6713	3.2169	800.4378	28.4708	73.7613	1.6812	1.7314
GAOERC	0.0166	0.1487	0.0260	183.7444	16.8835	2.6249	841.5940	43.5099	80.9470	5.6947	3.9104
GAOEW	0.0293	0.1092	0.0068	183.7444	16.8835	2.6249	841.5940	43.5099	80.9470	5.6947	3.9104
GAERC	0.0166	0.1487	0.0260	183.7444	16.8835	2.6249	841.5940	43.5099	80.9470	5.6947	3.9104
GAEW	0.0293	0.1092	0.0068	183.7444	16.8835	2.6249	841.5940	43.5099	80.9470	5.6947	3.9104
COERC	0.0162	0.1628	0.0270	141.8778	12.6325	3.2614	764.2937	51.2694	80.0615	7.2760	3.8788
COEW	0.0286	0.1266	0.0066	141.8778	12.6325	3.2614	764.2937	51.2694	80.0615	7.2760	3.8788
CERC	0.0162	0.1628	0.0270	141.8768	12.6331	3.2614	764.2937	51.2694	80.0615	7.2760	3.8788
CEW	0.0286	0.1266	0.0066	141.8768	12.6331	3.2614	764.2937	51.2694	80.0615	7.2760	3.8788
WOERC	0.0160	0.1624	0.0280	134.6055	11.3658	4.3987	746.8262	55.0209	79.2414	7.6259	3.8550
WOEW	0.0272	0.1275	0.0068	134.6055	11.3658	4.3987	746.8262	55.0209	79.2414	7.6259	3.8550
WERC	0.0160	0.1624	0.0280	134.6055	11.3658	4.3987	746.8262	55.0209	79.2414	7.6259	3.8550
WEW	0.0272	0.1275	0.0068	134.6055	11.3658	4.3987	746.8262	55.0209	79.2414	7.6259	3.8550

Table A.10: Comparison of performance criteria between the different allocation algorithms with rolling time window $n = 100$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0134	0.1442	0.0300	592.7077	28.3389	2.2704	486.7430	19.8481	46.3313	1.8414	2.2979
SOEW	0.0261	0.0420	0.0080	592.7077	28.3389	2.2704	486.7430	19.8481	46.3313	1.8414	2.2979
SERC	0.0134	0.1442	0.0300	592.7077	28.3389	2.2704	486.7430	19.8481	46.3313	1.8414	2.2979
SEW	0.0261	0.0420	0.0080	592.7077	28.3389	2.2704	486.7430	19.8481	46.3313	1.8414	2.2979
GAOERC	0.0168	0.1398	0.0252	186.3188	16.6996	1.8372	509.9514	24.7458	48.6880	5.5987	5.1641
GAOEW	0.0305	0.0948	0.0069	186.3188	16.6996	1.8372	509.9514	24.7458	48.6880	5.5987	5.1641
GAERC	0.0168	0.1398	0.0252	186.3188	16.6996	1.8372	509.9514	24.7458	48.6880	5.5987	5.1641
GAEW	0.0305	0.0948	0.0069	186.3188	16.6996	1.8372	509.9514	24.7458	48.6880	5.5987	5.1641
COERC	0.0158	0.1606	0.0279	118.5542	11.3215	2.2918	512.7307	34.4650	56.6143	8.5432	6.1515
COEW	0.0288	0.1197	0.0069	118.5542	11.3215	2.2918	512.7307	34.4650	56.6143	8.5432	6.1515
CERC	0.0158	0.1606	0.0279	118.5542	11.3215	2.2918	512.7307	34.4650	56.6143	8.5432	6.1515
CEW	0.0288	0.1197	0.0069	118.5542	11.3215	2.2918	512.7307	34.4650	56.6143	8.5432	6.1515
WOERC	0.0148	0.1640	0.0326	105.5847	9.3217	3.6294	472.2443	38.2956	53.7298	9.6286	6.7752
WOEW	0.0268	0.1233	0.0076	105.5847	9.3217	3.6294	472.2443	38.2956	53.7298	9.6286	6.7752
WERC	0.0148	0.1640	0.0326	105.5847	9.3217	3.6294	472.2443	38.2956	53.7298	9.6286	6.7752
WEW	0.0148	0.1640	0.0326	105.5847	9.3217	3.6294	472.2443	38.2956	53.7298	9.6286	6.7752

Table A.11: Comparison of performance criteria between the different allocation algorithms with rolling time window $n = 300$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

	TE	TO		Δk		κ				n_C	
		μ	σ	μ	σ	Min	Max	μ	σ	μ	σ
SOERC	0.0139	0.1409	0.0270	562.1450	21.6246	2.1134	419.4956	18.0460	39.6559	1.9362	2.5852
SOEW	0.0256	0.0410	0.0084	562.1450	21.6246	2.1134	419.4956	18.0460	39.6559	1.9362	2.5852
SERC	0.0139	0.1409	0.0270	562.1450	21.6246	2.1134	419.4956	18.0460	39.6559	1.9362	2.5852
SEW	0.0256	0.0410	0.0084	562.1450	21.6246	2.1134	419.4956	18.0460	39.6559	1.9362	2.5852
GAOERC	0.0166	0.1378	0.0246	188.4027	18.0820	1.6171	462.6501	21.6197	43.6479	5.5270	5.6719
GAOEW	0.0299	0.0911	0.0077	188.4027	18.0820	1.6171	462.6501	21.6197	43.6479	5.5270	5.6719
GAERC	0.0166	0.1378	0.0246	188.4027	18.0820	1.6171	462.6501	21.6197	43.6479	5.5270	5.6719
GAEW	0.0299	0.0911	0.0077	188.4027	18.0820	1.6171	462.6501	21.6197	43.6479	5.5270	5.6719
COERC	0.0163	0.1587	0.0288	112.7322	10.6603	2.0265	453.6685	30.6576	50.5414	8.9000	7.0242
COEW	0.0289	0.1162	0.0075	112.7322	10.6603	2.0265	453.6685	30.6576	50.5414	8.9000	7.0242
CERC	0.0163	0.1587	0.0288	112.7322	10.6603	2.0265	453.6685	30.6576	50.5414	8.9000	7.0242
CEW	0.0289	0.1162	0.0075	112.7322	10.6603	2.0265	453.6685	30.6576	50.5414	8.9000	7.0242
WOERC	0.0161	0.2224	0.0418	47.9644	8.3617	9.1626	534.7059	73.9350	85.8157	18.9239	11.5091
WOEW	0.0261	0.1734	0.0130	47.9644	8.3617	9.1626	534.7059	73.9350	85.8157	18.9239	11.5091
WERC	0.0148	0.1661	0.0358	96.4014	9.6168	3.7209	438.9038	35.0804	49.8256	10.4390	8.1613
WEW	0.0270	0.1217	0.0085	96.4014	9.6168	3.7209	438.9038	35.0804	49.8256	10.4390	8.1613

Table A.12: Comparison of performance criteria between the different allocation algorithms with rolling time window $n = 500$. Note that the κ and n_C values are averages. Δk represents the difference between k_{final} and k^*

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

AVANCEZ
1829

CHALMERS
UNIVERSITY OF TECHNOLOGY