A Graph Theory Approach to Portfolio Optimization Part II

Dany Cajas[*]

[First version: November 2023] [Latest version: December 2023]

Abstract

This work presents two approaches that allow us to diversify portfolios based on the graphical representation of the hierarchical relationships among assets. These formulations are modifications of the integer programming and semidefinite programming formulations presented in a previous work that instead of consider the network structure, consider the clusters obtained from the dendrogram structure. We run some examples that show how classic convex models and graph clustering-based asset allocation models do not incorporate properly the information about the clusters in the optimization process, while the addition of constraints on clusters allow us to diversify our portfolio selecting the best assets per cluster.

Keywords: finance, dendrogram, hierarchical clustering, graph theory, portfolio optimization.
JEL Codes: C61, G11
Acknowledgement: I want to thanks and dedicate this work to my parents Daniel and Pilar, and thanks to Catherine, Mappy and Alberto for their support.

[*] E-mail: dcajasn@gmail.com

1 Introduction

Mantegna (1999) showed that dendrograms can be used to represent the hierarchical relationships among financial assets; however, since the development of the hierarchical risk parity (HRP) asset allocation model by López de Prado (2016), the use of dendrograms in finance have gained popularity among students, academics and practitioners. Also, improvements to the HRP model that incorporate a decision criteria to select an optimal number of cluster from the dendrogram have been developed like the hierarchical equal risk contribution (HERC) by Raffinot (2018) and the nested clustered optimization (NCO) by (Prado, 2019). Nevertheless, these graph clustering-based models have to main disadvantages: they are not properly optimization models and they split the wealth among all assets. The first disadvantage means that these models do not have the flexibility to incorporate additional constraints like constraints on maximum risk, minimum return or convex constraints into the optimization process; while the second disadvantage means that these models do not select the best assets from the sample and only split the wealth based on the dendrogram structure.

In a previous work, Cajas (2023) showed two approaches that allow us to incorporate the information from graphs like the minimum spanning tree (MST) and the triangulated maximally filtered graph (TMFG) into the portfolio optimization process through constraints in centrality measures and constraints in the neighborhood of assets in the graph. In this work, we are going to show how taking advantage of the integer programming and semidefinite programming formulation of the neighborhood of assets constraint, we can incorporate information of the clusters from the dendrogram into the optimization process.

In this work, we are going to show how taking advantage of the integer programming and semidefinite programming formulation of the neighborhood of assets constraint proposed by Cajas (2023), we can modified the input matrix of these constraints to incorporate information of the clusters from the dendrogram into the optimization process. First, we are going to explain how the clusters from the dendrogram can be expressed as matrices. Then, we are going to explain how we can add a constraint on the clusters of assets into a classic convex optimization problem through integer programming or semidefinite programming constraints. Finally, we run some examples that show how classic convex models and graph clustering-based asset allocation models do not incorporate the information of clusters from the dendrogram to select best assets, while the simple addition of our constraints on the clusters from the dendrogram in the convex optimization problems allow us to diversify considering assets that do not belong to same cluster.

2 Dendrograms

2.1 Graphical Representations of Dendrograms

A dendrogram is a diagram that represents a tree. It is used in hierarchical clustering to illustrate the hierarchical relationships among variables. Also, it can be used to identify visually the clusters obtained through the hierarchical clustering process[1] and an optimal number of clusters criteria like elbow method or gap statistic.

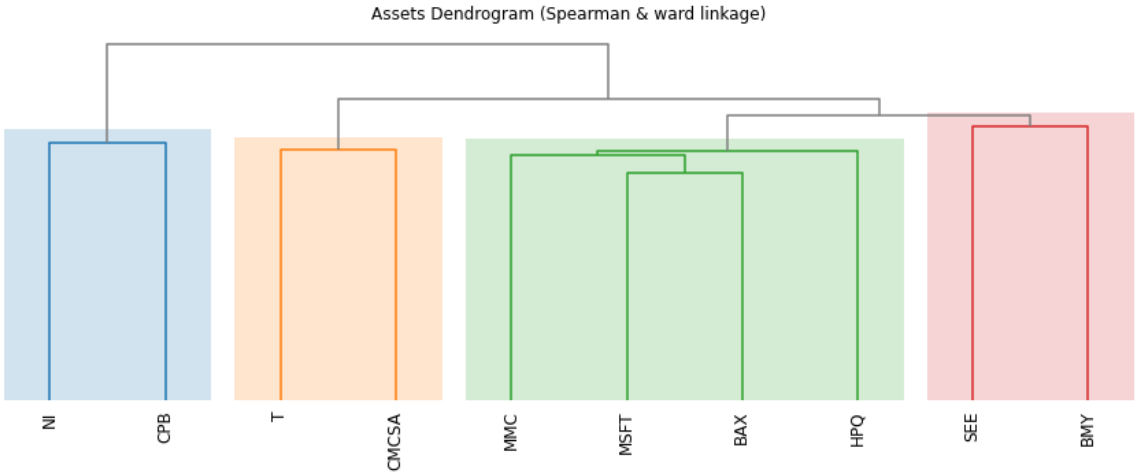

Figure 1: Dendrogram with Optimal Number of Clusters

In figure 1 we can see the dendrogram for a group of ten assets (BAX, BMY, CMCSA, CPB, HPQ, MMC, MSFT, NI, SEE, T) obtained using a hierarchical clustering algorithm based on a distance metric based on spearman correlation and ward linkage; were the optimal number of clusters was determined using the two difference gap statistic[2]. We can see that dendrograms help us to identify visually the clusters of assets that are more related among them. On the other hand, we can plot the clusters of figure 1 as four complete graphs as follows:

[1] For more information about the hierarchical clustering process see Müllner (2011)

[2] Yue et al. (2008)

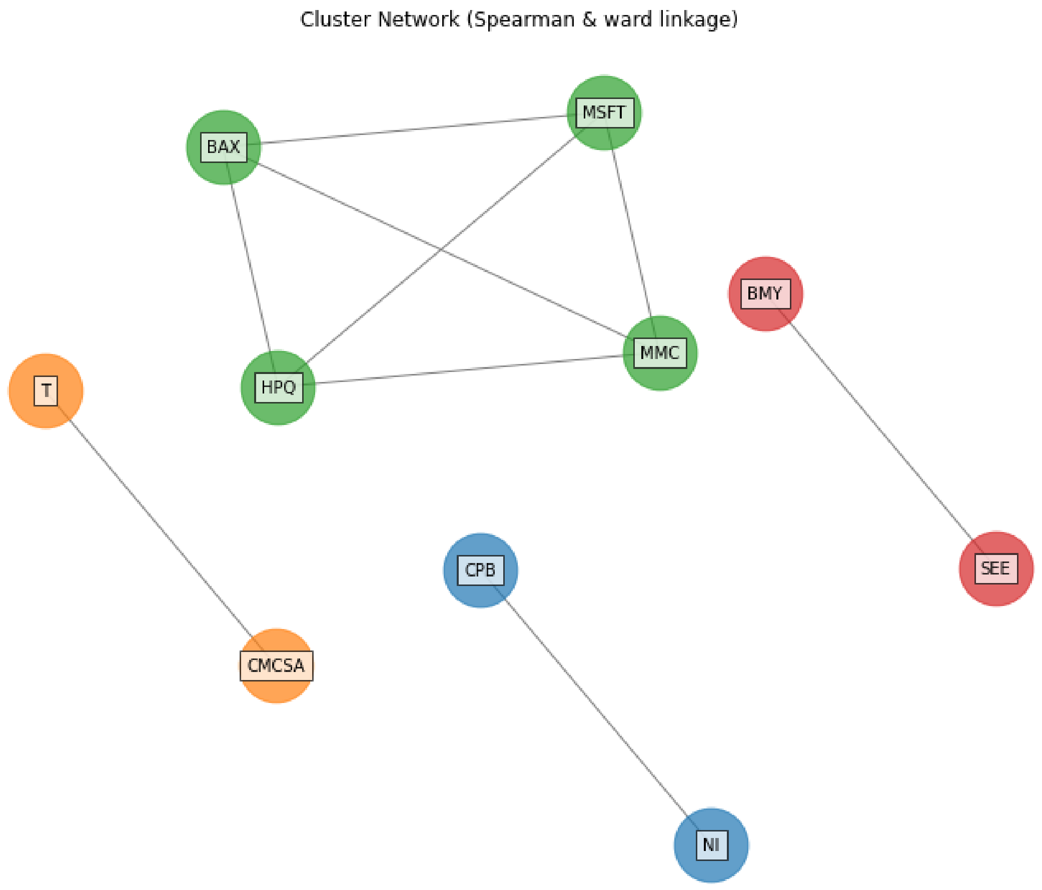

Figure 2: Clusters Network

In figure 2 we can see that the clusters that we visualize on dendrograms can be interpreted as undirected complete graphs, because in each cluster all assets have a similar behavior and they are related.

2.2 Matrix Representation of Dendrograms

The graphical representation allow us to have an initial idea of how assets are related; however we cannot incorporate that information into the portfolio optimization process until we transform these graphs into a matrix form that allow us to build constraints.

The first matrix representation, that we call ***label matrix*** L, is based on the idea that each row represent a cluster (label), each column represents an asset (variable) and in the intersection of each row and column we write a one if the assets belong to that cluster or zero if not. This kind of matrix are widely used in finance with traditional labels like industry, country, sector, etc; to build linear constraints related to limits on weights for example a concentration (max total weight) limit by industry[3].

$$\begin{array}{l|cccccccccc|}
 & \text{NI} & \text{CPB} & \text{T} & \text{CMCSA} & \text{MMC} & \text{MSFT} & \text{BAX} & \text{HPQ} & \text{SEE} & \text{BMY} \\
\text{Cluster 1} & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{Cluster 2} & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{Cluster 3} & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
\text{Cluster 4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}$$

Figure 3: Label Matrix L

In figure 3 we can see the label matrix associated to clusters from dendrogram of figure 1. The second matrix representation, that we called ***adjacency label matrix*** L_A, is an adjacency matrix that represents the cluster network associated to the label matrix L. Mathematically it is defined as:

$$L_A = L'L - I_n \tag{1}$$

where L is the label matrix, I_n is the identity matrix of size n and n is the number of assets.

$$\begin{array}{c|cccccccccc|}
 & \text{NI} & \text{CPB} & \text{T} & \text{CMCSA} & \text{MMC} & \text{MSFT} & \text{BAX} & \text{HPQ} & \text{SEE} & \text{BMY} \\
\text{NI} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{CPB} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{T} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{CMCSA} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{MMC} & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
\text{MSFT} & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
\text{BAX} & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
\text{HPQ} & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
\text{SEE} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\text{BMY} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}$$

Figure 4: Adjacency Label Matrix L_A

In figure 4 we can see the adjacency label matrix associated to figure 2. This matrix has a diagonal block structure where each block represents the adjacency matrix of the

[3] It is easily to see that the product Lx of the label matrix L and the vector of weights x gives the total weight per label.

undirected complete graph of the cluster.

2.3 Dendrogram Measures for Portfolios

2.3.1 Percentage Invested in Related Assets

In the case of an investment portfolio, we can define the percentage invested in related assets in the same way that Cajas (2023) define the percentage invested connected assets of a portfolio but replacing the connection matrix $\mathbf{B}_{1,l}$ with the adjacency label matrix L_A as follows:

$$\mathbf{RA}(x) = \frac{\mathbf{1}_n \left(L_A |xx'| \right) \mathbf{1}_n'}{\mathbf{1}_n |xx'| \mathbf{1}_n'} \tag{2}$$

where x is the column vector of weights of the portfolio of size $n \times 1$ and $|.|$ is the element wise absolute value. This formula allows us to measure the percentage invested in assets that belong to the same cluster. The idea is that when we invest in assets that are in the same clusters, the absolute value of the product of weights $|x_i x_j| = 0$ because one of the assets must have a weight equal zero, this means that $x_i = 0$ or $x_j = 0$.

This indicator gives us an idea of the diversification of the portfolio. The idea here is that portfolios that invest more in assets that are in same clusters are concentrating risk in assets that have a similar behavior. This means that is preferred to invest in assets that are in different clusters.

3 Clusters Constraint

3.1 Mixed Integer Programming Approach

Ricca and Scozzari (2024) proposed an integer programming constraint that allows us to invest in assets that are not neighbors, this means that are not directly linked through an edge. Cajas (2023) generalized this idea for walks of size lower or equal to l using the connection $\mathbf{B}_{1,l}$ and a mixed integer programming (MIP) formulation. Based on this approach we can incorporate the information from clusters replacing the connection matrix $\mathbf{B}_{1,l}$ with the label matrix L as follows:

$$
\begin{aligned}
\underset{x}{\text{opt}} \quad & \phi(x) \\
\text{s.t.} \quad & Ly \leq 1 \\
& x_i \leq b_u y_i \; \forall \; i = 1, \ldots, n \\
& x_i \geq b_l y_i \; \forall \; i = 1, \ldots, n \\
& x \in \mathcal{X}
\end{aligned}
\tag{3}
$$

where $y \in \{0, 1\}$ is binary variable of size $n \times 1$ that indicates if an asset is considered in the portfolio and b_l and b_u are the lower and upper bounds of variable x. The idea of this model is that the constraint $Ly \leq 1$ guarantee that we invest in one asset per cluster, this means select the best asset in the cluster. We can generalize this constraint to $Ly \leq k$ where k is the maximum number of assets per cluster or even define a maximum number of asset per clusters $L_i y \leq k_i$, where L_i is the row i of matrix L and k_i is the maximum number of assets of cluster i. On the other hand, the constraint $Ly \leq 1$ works like a cardinality constraint where the maximum number of assets is the number of clusters.

The main advantages of this model is that can be applied to any convex risk measure. The main disadvantages of this model is that relies in a mixed integer programming (MIP) formulation, making it difficult to solve when we increase the number of assets.

3.2 Semidefinite Programming Approach

In this section we are going to use the semidefinite programming (SDP) formulation of the neighborhood constraint proposed by Cajas (2023) to implement a constraint that allow us to invest in one asset per cluster. To implement this constraint we are going to replace the connection matrix $\mathbf{B}_{1,l}$ with the adjacency label matrix L_A. The idea is that if we invest only in one asset in the cluster, the product of weights of the assets that belong to a cluster approximates to zero. We can pose the minimization of variance with a constraint to invest in assets that are not in the same cluster as follows:

$$
\begin{aligned}
\min_{x,\,X} \quad & \operatorname{Tr}(\Sigma X) \\
\text{s.t.} \quad & \begin{bmatrix} X & x \\ x' & 1 \end{bmatrix} \succeq 0 \\
& X = X' \\
& L_A \odot X = 0 \\
& x \in \mathcal{X}
\end{aligned}
\tag{4}
$$

where X is an auxiliary variable that approximate the product of asset's weights xx' and Σ is the covariance matrix. This idea can be applied to risk measures whose optimization models can be expressed as semidefinite programming problem using the constraint $\begin{bmatrix} X & x \\ x' & 1 \end{bmatrix} \succeq 0$ like kurtosis. In the case of risk measures whose portfolio optimization models can not be expressed using this constraint, we can reduce the amount invested in assets that are related adding a penalty function $\lambda \operatorname{Tr}(X)$ in the objective function as follows:

$$
\begin{aligned}
\operatorname*{opt}_{x} \quad & \phi(x) + \lambda \operatorname{Tr}(X) \\
\text{s.t.} \quad & \begin{bmatrix} X & x \\ x' & 1 \end{bmatrix} \succeq 0 \\
& X = X' \\
& L_A \odot X = 0 \\
& x \in \mathcal{X}
\end{aligned}
\tag{5}
$$

where λ is a penalty factor. The advantage of this approach compared to the MIP approach is that gives us an approximate solution even when the MIP model cannot find a solution. The main disadvantage of the semidefinite approach is that we cannot generalize this constraint to invest in more than one asset per cluster like in the case of the MIP approach. Finally, this constraint also works like an approximate cardinality constraint where the maximum number of assets is the number of clusters.

4 Numerical Examples

We select 30 assets (i.e., stocks JCI, TGT, CMCSA, CPB, MO, T, APA, MMC, JPM, ZION, PSA, BAX, BMY, AAPL, PCAR, BA, TMO, TXT, DE, MSFT, HPQ, SEE, VZ, CNP, NI, JNJ, PFE, AMZN, GE and GOOG) from the S&P 500 (NYSE) and download daily adjusted closed prices from Yahoo Finance for the period from January 1, 2019 to December 30, 2022. Then, we calculated daily returns building a returns matrix of size $T = 1006$ and $N = 100$. To calculate the portfolios we use Python 3.10, CVXPY[4] and MOSEK[5] solver.

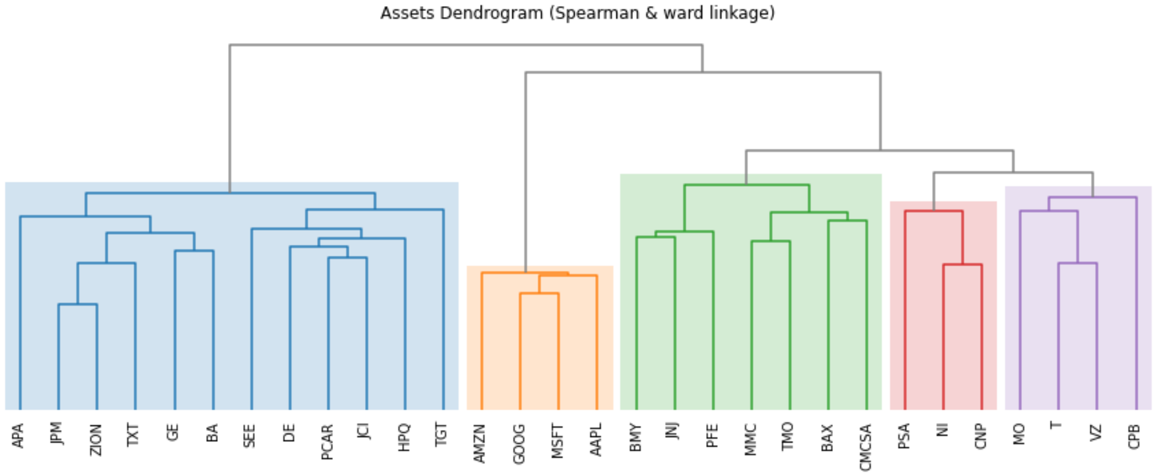

Figure 5: Dendrogram of Selected Assets

In figure 5 we can identify 5 clusters using a distance based on spearman correlation and ward linkage. Using this dendrogram we are going to build the label matrix L and adjacency label matrix L_A that we are going to use in our formulations.

4.1 Minimum Variance Optimization

In this section we are going to compare the minimum variance portfolio with the minimum variance that considers the clusters constraints using the MIP and SDP formulations.

[4] Diamond and Boyd (2016) and Agrawal et al. (2018)

[5] ApS (2023)

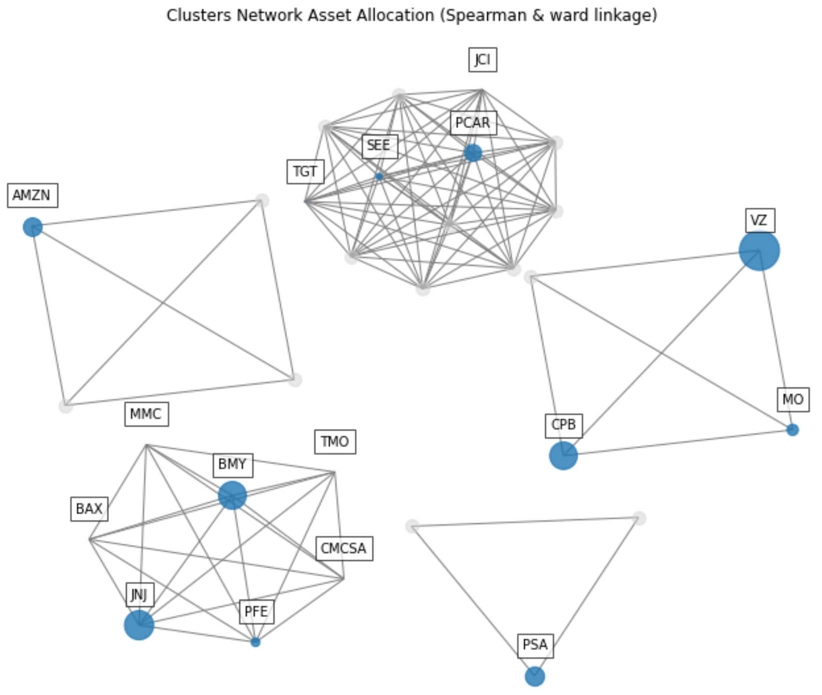

Figure 6: Cluster Network Allocation of Minimum Variance

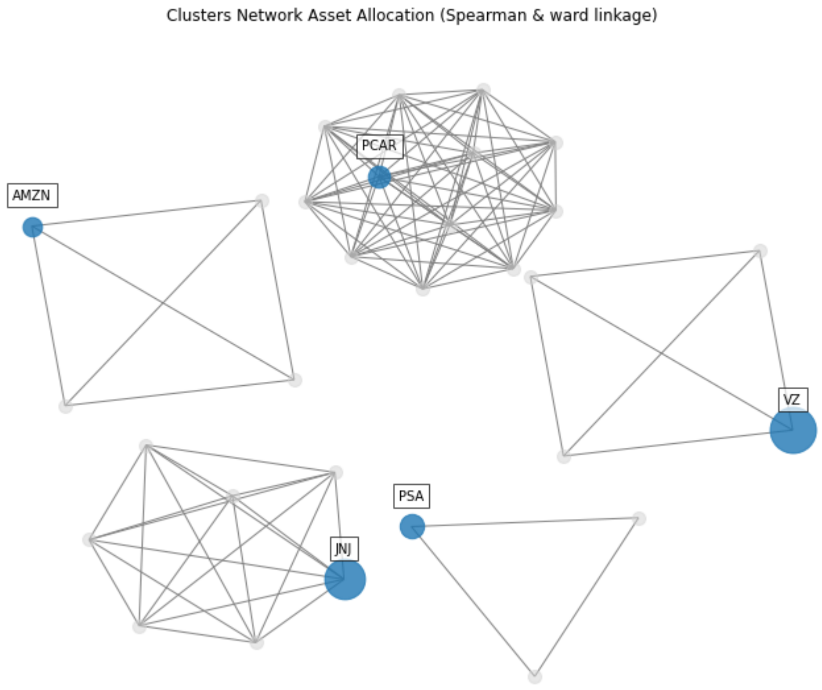

Figure 7: Cluster Network Allocation of Minimum Variance with MIP Constraint

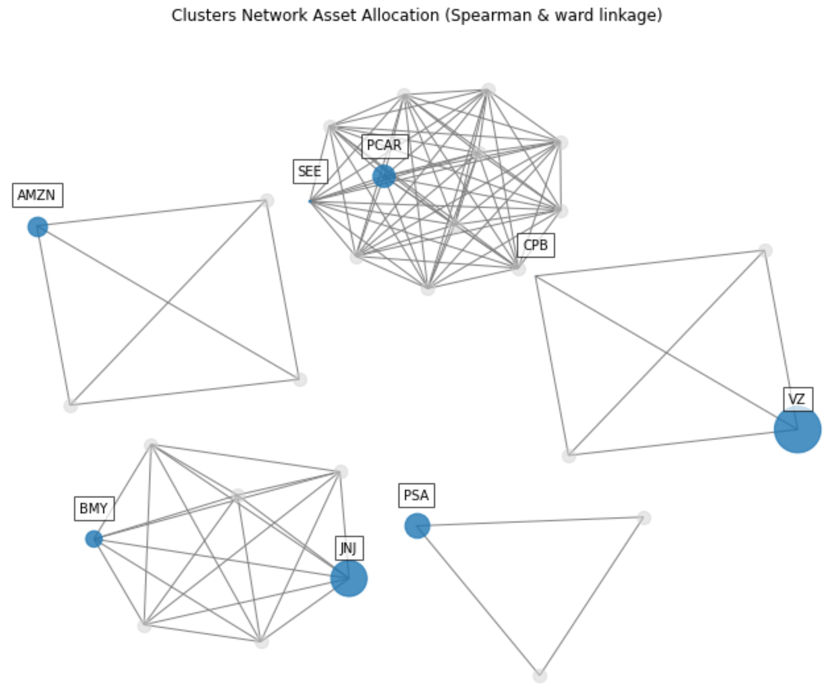

Figure 8: Cluster Network Allocation of Minimum Variance with SDP Constraint

Clusters	Min Variance	MIP	SDP
1	6.13%	9.29%	9.49%
2	6.64%	7.21%	7.27%
3	32.58%	31.46%	30.08%
4	7.03%	11.33%	11.71%
5	47.62%	40.72%	41.44%
$\sigma(x)$	1.00%	1.05%	1.04%
NEC(x)	2.89	3.43	3.45
RA(x)	16.93%	0.00%	2.58%

Table 1: Composition per Portfolio

In figure 6 we can see the asset allocation of the minimum variance portfolio per cluster, the portfolio split the wealth among several assets that belong to all clusters, however as we can see in table 1 it concentrates the weights in two clusters and has a number of effective clusters (NEC)[6] of 2.89. On the other hand, in figures 7 and 8 we can see that the addition of the MIP and SDP constraints to the minimum variance problem reduce the number of assets but as we can see in table 1 both formulations

[6] This indicator is calculated as the inverse of the Herfindahl-Hirschman index of cluster weights.

increase the diversification per cluster, due to the number of effective assets in both portfolios increases to 3.43 and 3.45 respectively. Finally, we can see that the SDP formulation has a closer standard deviation to the minimum variance portfolio but at the cost of a little higher percentage invested in related assets respect to the MIP formulation.

4.2 Minimum RLVaR Optimization

In this section we are going to compare the minimum relativistic value at risk (RLVaR) portfolio with the minimum RLVaR that considers the clusters constraints using the MIP and the SDP penalty formulations.

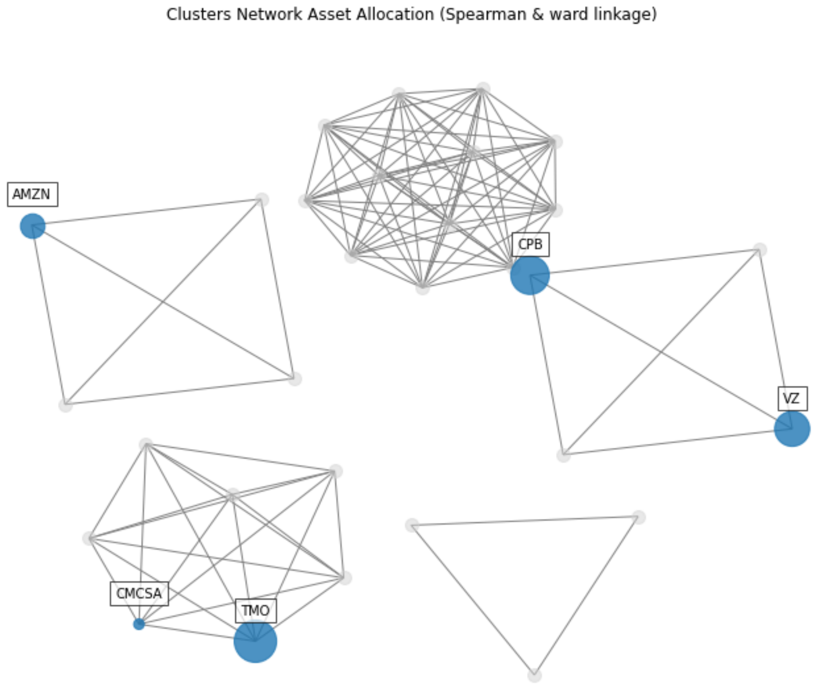

Figure 9: Cluster Network Allocation of Minimum RLVaR

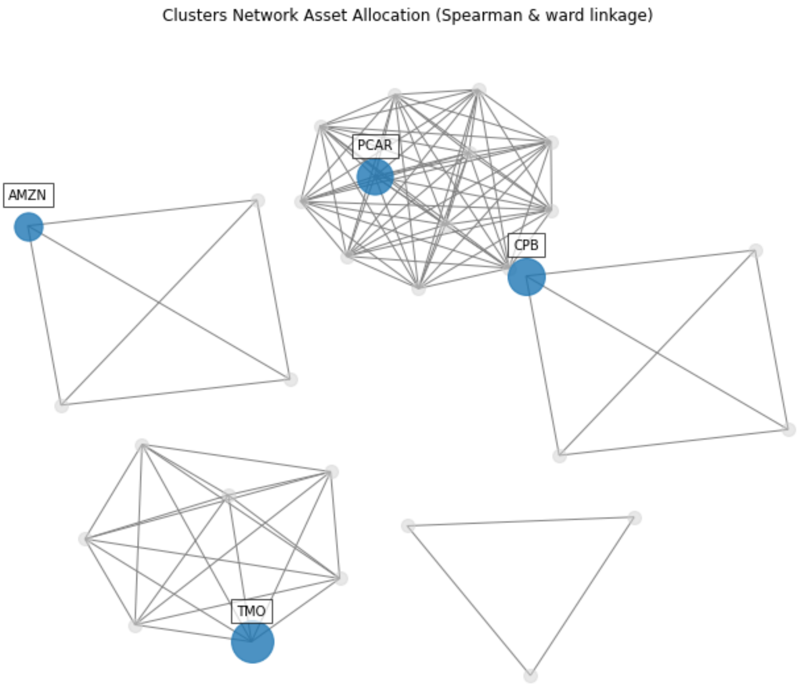

Figure 10: Cluster Network Allocation of Minimum RLVaR with MIP Constraint

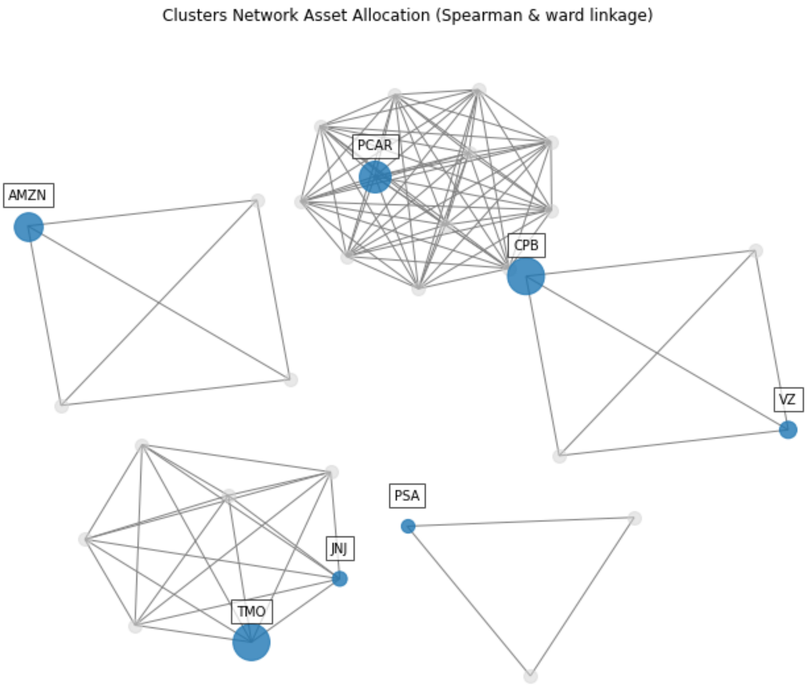

Figure 11: Cluster Network Allocation of Minimum RLVaR with SDP Penalty

Clusters	Min RLVaR	MIP	SDP
1	0.00%	24.82%	19.12%
2	11.50%	15.18%	15.70%
3	36.55%	33.89%	30.00
4	0.00%	0.00%	3.45%
5	51.95%	26.11%	31.73%
$\text{RLVaR}_\kappa(x)$	3.48%	3.79%	3.68%
$\text{NEC}(x)$	2.40	3.74	3.95
$\text{RA}(x)$	14.86%	0.00%	5.08%

Table 2: Composition per Portfolio

In figure 9 we can see the asset allocation of the minimum RLVaR portfolio per cluster, the portfolio split the wealth among assets that belong to three clusters, however as we can see in table 2 it concentrates the weights in two clusters (number of effective assets (NEA) is 2.40). On the other hand, in figures 10 and 11 we can see that the addition of the MIP constraint and SDP penalty to the minimum RLVaR problem increase the diversification per cluster as shown in table 2, due to the number of effective assets in both portfolios increases to 3.74 and 3.95 respectively. Finally, we can see that the SDP penalty formulation has a closer RLVaR to the minimum RLVaR portfolio but at the cost of a little higher percentage invested in related assets respect to the MIP formulation.

4.3 Graph Clustering-Based Asset Allocation

In this section we are going to compare the minimum variance portfolio that considers the clusters constraints using the MIP and SDP formulations with the hierarchical with the latest graph clustering-based models: HRP, HERC and NCO.

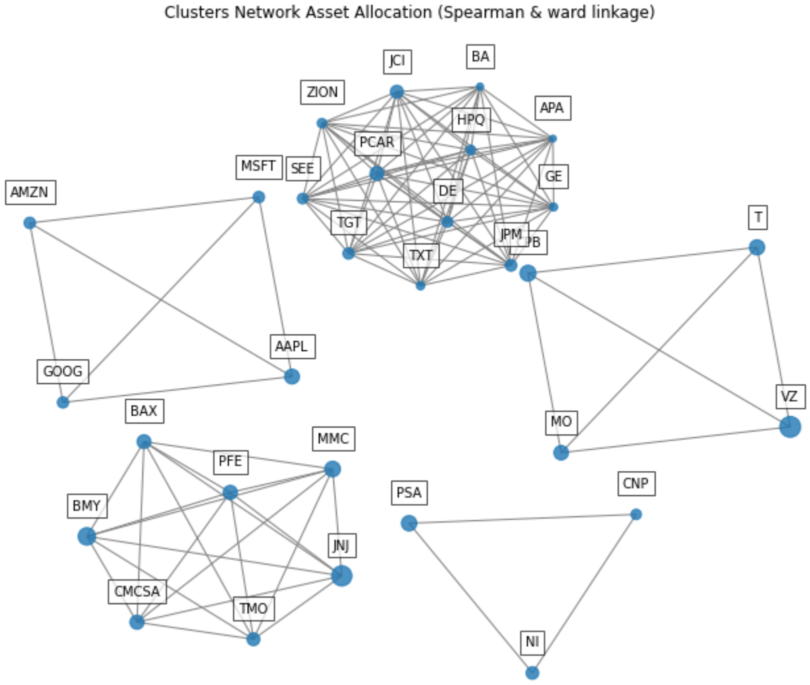

Figure 12: Cluster Network Allocation of HRP

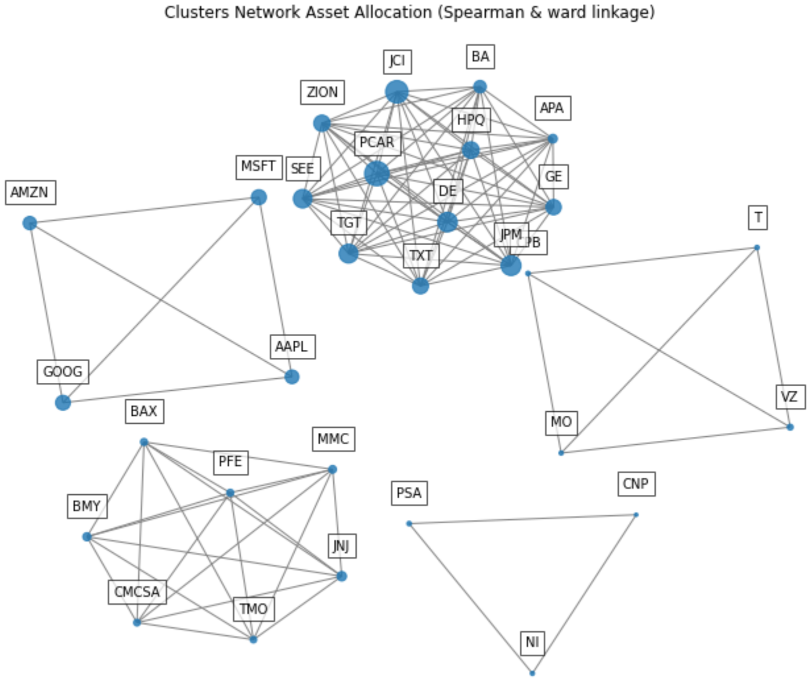

Figure 13: Cluster Network Allocation of HERC

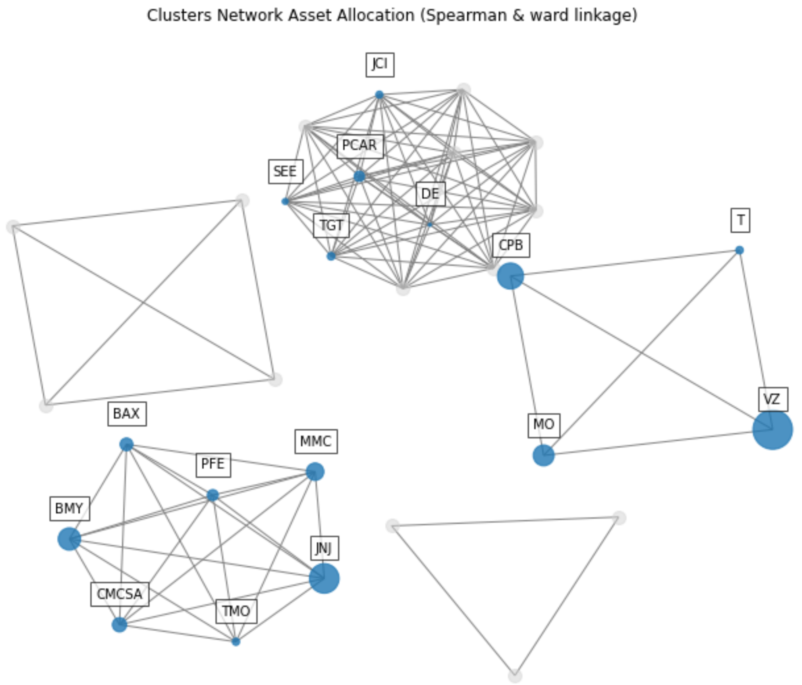

Figure 14: Cluster Network Allocation of NCO

Clusters	HRP	HERC	NCO	MIP	SDP
1	24.07%	73.91%	4.98%	24.82%	19.12%
2	11.66%	15.51%	0.00%	15.18%	15.70%
3	32.87%	7.76%	42.63%	33.89%	30.00%
4	9.67%	0.99%	0.00%	0.00%	3.45%
5	21.72%	1.84	52.39%	26.11%	31.73%
$\sigma(x)$	1.19%	1.57%	1.03%	1.05%	1.04%
NEC(x)	4.23	1.73	2.18	3.74	3.95
RA(x)	19.34%	51.58%	30.05%	0.00%	5.08%

Table 3: Composition per Portfolio

In figure 12 we can see the asset allocation of the HRP portfolio per cluster, the portfolio split the wealth among all assets and clusters, the portfolio diversified well among clusters with a NEC of 4.23; however, it has a RA of 19.34%, this means that around the 19% of the wealth is invested in assets that belong to the same cluster.

In figure 13 we can see the asset allocation of the HERC portfolio per cluster, the portfolio split the wealth among all assets, the portfolio has the lowest diversification among clusters with a NEC of 1.73. Also, it has a RA of 51.58%, this means that around the 52% of the wealth is invested in assets that belong to the same cluster.

In figure 14 we can see the asset allocation of the NCO portfolio per cluster, the portfolio split the wealth among several assets, the portfolio has a low diversification among clusters with a NEC of 2.18. Also, it has a RA of 30.05%, this means that around the 30% of the wealth is invested in assets that belong to the same cluster.

Finally, we can notice that the graph clustering-based models do not take advantage of the clustering structure obtained through the dendrogram because instead of select the best assets from each cluster to improve the diversification per cluster, they split the weights among all assets of the clusters, as we can see in figures 12, 13 and 14. This behavior implies that the portfolios obtained using these models have a number of assets that increase in the same proportion than the number of assets of each cluster.

5 Conclusions

This work proposes two approaches that allow us to incorporate information from a dendrogram structure into classic return risk trade-off portfolio optimization models. Taking advantage of the matrix representation of the clusters obtained from dendrograms through a label matrix and adjacency label matrix, we can add constraints that allow us to diversify the portfolio based on constraints in the clusters. We show in some examples that classic convex portfolio optimization problems do not incorporate information from clusters in the diversification process. Also, the latest graph clustering-based asset allocation models like HRP do not diversify by selecting the best assets per cluster; because these models assign weights to all assets that belong to a selected cluster. Finally, we show that the incorporation of constraints on the clusters in the convex portfolio optimization problems allows us to increase the diversification by cluster of our portfolio and also reduce the portfolio cardinality by selecting the best assets per cluster.

References

Agrawal, A., R. Verschueren, S. Diamond, and S. Boyd (2018). A rewriting system for convex optimization problems. *Journal of Control and Decision 5*(1), 42–60.

ApS, M. (2023). *MOSEK Optimizer API for Python 10.1.8*.

Cajas, D. (2023, 10). A graph theory approach to portfolio optimization. *SSRN Electronic Journal*.

Diamond, S. and S. Boyd (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research 17*(83), 1–5.

López de Prado, M. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management 42*(4), 59–69.

Mantegna, R. (1999, sep). Hierarchical structure in financial markets. *The European Physical Journal B 11*(1), 193–197.

Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms.

Prado, M. (2019, 01). A robust estimator of the efficient frontier. *SSRN Electronic Journal*.

Raffinot, T. (2018, 08). The hierarchical equal risk contribution portfolio.

Ricca, F. and A. Scozzari (2024, January). Portfolio optimization through a network approach: Network assortative mixing and portfolio diversification. *European Journal of Operational Research 312*(2), 700–717.

Yue, S., X. Wang, and M. Wei (2008, 06). Application of two-order difference to gap statistic. *Transactions of Tianjin University 14*, 217–221.