A Graph Theory Approach to Portfolio Optimization

Dany Cajas[*]

[First version: September 2023] [Latest version: October 2023]

Abstract

This work presents two approaches that allow us to diversify portfolios based on the graphical representation of the relationships among assets. We can use the information obtained from graphs like the minimum spanning tree (MST) or the triangulated maximally filtered graph (TMPG) to diversify our portfolio based on the influence of assets in the graph (centrality) and the asset's neighborhood. These formulation are simple and versatile because they consist on additional constraints that we can add to traditional convex portfolio optimization problems. We run some examples that show how classic convex models and graph clustering-based asset allocation models do not incorporate information about the centrality and connections among assets in the optimization process, while the addition of constraints on a centrality measure or in the asset's neighborhood allow us to diversify our portfolio selecting assets in the periphery of the graph or assets that are not directly connected in the graph respectively.

Keywords: finance, MST, TMPG, 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

The use of graphs in asset allocation has gained attention since the development of the hierarchical risk parity (HRP) asset allocation model by López de Prado (2016). There are three main approaches in the use of graphs in asset allocation: peripherality and centrality measure-based approach, mixed integer linear programming (MILP) approach and graph clustering-based approach.

The peripherality and centrality measure-based approach consists in use the graph structure of assets to select stocks based on a centrality measure and then solve an optimization problem like the works of Pozzi et al. (2013) and Li et al. (2019); or modifying the covariance matrix based on the graph structure and then solve a variance problem like the works of Peralta and Zareei (2016). The main advantage of this approach is its simplicity; however, the main disadvantage is that the asset selection based on the graph step and the portfolio optimization steps are made in different stages.

The MILP approach formulates MILP problems that includes integer constraints based on the graph of relationship among assets like the works of Puerto et al. (2020) and Ricca and Scozzari (2024). The advantage of this approach is that incorporate the graph information in the portfolio optimization step; however, the main disadvantage is that MILP does not scale well for medium and large scale problems.

The graph clustering-based approach is the most popular approach, it includes mainly three different models: the HRP (López de Prado, 2016), the hierarchical equal risk contribution (HERC) (Raffinot, 2018) and the nested clustered optimization (NCO) (Prado, 2019). These models take advantage of the hierarchical relationships that exists among assets to create an asset allocation that diversify the risk considering this hierarchical relationships. The main advantage of this approach is that these models incorporate the hierarchical structure based on a dendrogram in the optimization step; however, their main disadvantage is that they do not have the flexibility to incorporate additional constraints like constraints on maximum risk, minimum return or convex constraints because they are not properly optimization models.

This work proposes two approaches that allow us to incorporate information obtained from graphs into the optimization process. The first approach consist on add information about the influence of assets in a graph through a linear constraint in an average centrality measure of the portfolio. The second approach consist on add information about the connections among assets in the graphs through a constraint in the asset's neighborhood that can be modeled using integer programming or semidefinite programming. The advantage of both approaches is that can be easily incorporated into classical return risk trade-off portfolio optimization models. First, we are going to explain some basic concepts of graph theory, how to represent graphs using an adjacency matrix and how this matrix representation allow us to measure the centrality and weight invested in connected assets in a graph. Next, we are going to explain the two most used networks in financial markets: the minimum spanning tree (MST) and the triangulated maximally filtered graph (TMFG). Then, we are going to explain how we can add a constraint on the average centrality measure of a portfolio into a classic convex optimization problem as a linear constraint. After that, we are going to explain how we can add a constraint on the neighborhood of assets into a classic convex optimization problem using integer or semidefinite programming. Finally, we run some examples that show how classic convex models and graph clustering-based asset allocation models do not incorporate the information from the centrality and neighborhood of assets in the graph, while the simple addition of our constraints on the average centrality measure or neighborhood in the convex optimization problems allow us to diversify considering assets in the periphery of the graph or assets that are not directly connected in the graph respectively.

2 Basics on Graph Theory

2.1 Graphical Representations of Graphs

A graph is defined as the pair $G = (V, E)$, where V is a set of elements called vertices and E is a set of paired vertices called edges that represent the links between vertices.

1. **Undirected Graph**: is a graph in which the edges do not point in any direction, this means that the edge are bidirectional.

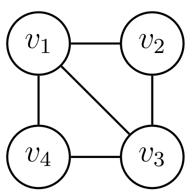

Figure 1: Undirected Graph

2. **Connected Graph:** is a graph in which there is always a path from a vertex to any other vertex.

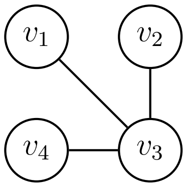

Figure 2: Connected Graph

3. **Complete Graph:** is an undirected graph in which every pair of distinct vertices is connected by a unique edge.

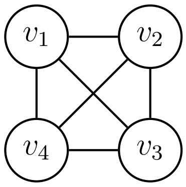

Figure 3: Complete Graph

4. **Weighted Graph:** is a graph in which every edge has a weight.

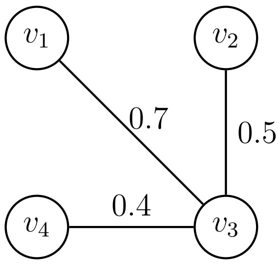

Figure 4: Weighted Graph

5. **Spanning tree:** is a subgraph of an undirected connected graph, which includes all the vertices of the graph with a minimum possible number of edges. The edges may or may not have weights assigned to them. The total number of spanning trees with n vertices that can be created from a complete unweighted graph is equal to n^{n-2}.

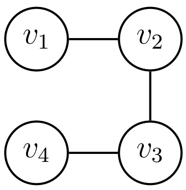

Figure 5: Spanning tree

2.2 Matrix Representation of Graphs

The graphical representation is not too helpful to pose optimization problems. To incorporate the information of graphs into optimization problems we need a matrix representation of a graph. An adjacency matrix A is a matrix whose rows and columns are the vertices of the graph, and the numbers in the matrix are the number of edges that connect vertex i to vertex j. For example, if we have the graph $G = (V, E)$ where $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and $E = \{(v_1, v_2), (v_1, v_5), (v_2, v_3), (v_2, v_4), (v_3, v_4), (v_4, v_5), (v_4, v_6), (v_6, v_6)\}$, we can represent it in graphical and matrix form as follows:

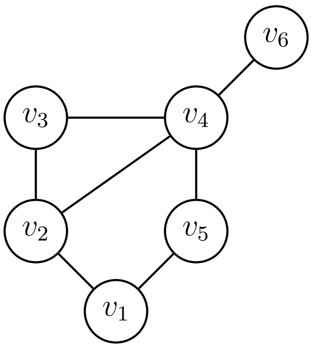

Figure 6: Labeled Graph

$$
\begin{array}{c}
\\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

Figure 7: Adjacency Matrix A

One important property of the adjacency matrix A is that the element (i, j) of the matrix A^k [1] give us the number of walks or connections of size k between node i and j. Based on this property, we can define for a graph without loops of n nodes, the connection matrix B_k of non closed walks of length k as follows:

$$\mathbf{B}_k = \mathbf{1}_{x>1}(A^k + I_n) - I_n \tag{1}$$

where $\mathbf{1}_{x>1}(.)$ is the element wise indicator function and I_n is the identity matrix of size n. The matrix B_k tells us if two different nodes i and j are connected through a walk of size k.

[1] $A^k = \underbrace{A\,A\ldots A}_{k \text{ times}}$

$$\begin{array}{c|c} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} & \begin{bmatrix} 2 & 0 & 1 & 2 & 0 & 0 \\ 0 & 3 & 1 & 1 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 \\ 2 & 1 & 1 & 4 & 0 & 0 \\ 0 & 2 & 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Figure 8: Matrix A^2

$$\begin{array}{c|c} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Figure 9: Connection Matrix $\mathbf{B}_2$

In figure 8 the matrix A^2 tell us the number of walks of length two that connect nodes v_i and v_j; and in figure 9 the connection matrix B_2 tells us if two different nodes v_i and v_j are connected through at least one walk of length two. We can generalize the idea for nodes that are connected through walks of different lengths whose maximum length is l, formally it is expressed as:

$$\mathbf{B}_{1,l} = \mathbf{1}_{x>1} \left(\sum_{k=1}^{l} \mathbf{B}_k \right) \tag{2}$$

2.3 Centrality Measures of Graphs

The matrix representation of a graph through the adjacency matrix tells us how nodes are linked, however we need some measures that tell us if a node is in the periphery or in the center of a graph. To answer this question we use the centrality measures[2], these kind of measures give us an idea of the influence of a node in the network. Some popular centrality measures are:

2.3.1 Node's Degree

The node's degree is the number of edges that are directly connected to a node. We can calculate the node's degree vector $\mathbf{D}_n$, that is a vector that contains all the node's degree, using the following formula:

$$\mathbf{D}_n = A\,\mathbf{1}_n \tag{3}$$

[2] A more extensive explanation of centrality measures can be found in Estrada (2011)

where $\mathbf{1}_n$ is a column vector of ones of size $n \times 1$. For the adjacency matrix of figure 6, the node's degree vector is calculated as follows:

$$
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array}
\begin{array}{c}
\begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 2
\end{bmatrix}
\end{array}
\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
=
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array}
\begin{bmatrix} 2 \\ 3 \\ 2 \\ 4 \\ 2 \\ 3 \end{bmatrix}
$$

Figure 10: Degree Vector

2.3.2 Eigenvector Centrality

The eigenvector centrality of node i measures the influence of this node beyond its neighbors, this means that considers connections that are composed for more than one edge (walks of lenght longer than one). We can calculate the eigenvector centrality vector $\mathbf{EC}_n$, that is a vector that contains all the node's eigenvector centralities, using the following formula:

$$\mathbf{EC}_n = \frac{1}{\lambda_{\max}} A \, q_{\max} \tag{4}$$

where $\lambda_{\max}$ and $q_{\max}$ are the maximum eigenvalue and associated eigenvector of adjacency matrix A.

2.3.3 Subgraph centrality

The subgraph centrality measures the importance of a node i in a graph by considering its participation in all closed walks starting (and ending) at it. We can calculate the subgraph centrality vector $\mathbf{EE}_n$, that is a vector that contains all the node's subgraph centralities, using the following formula:

$$
\begin{aligned}
\mathbf{EE}_n &= \operatorname{diag}\left(e^A\right) \\
\mathbf{EE}_n &= \operatorname{diag}\left(\sum_{i=0}^{\infty} \frac{A^i}{i!}\right)
\end{aligned} \tag{5}
$$

where diag(.) is the operator that extract the diagonal of a matrix, e^A is the matrix exponential of adjacency matrix A. The sum of all elements of subgraph centrality vector is the Estrada Index, that was originally designed as a way of characterizing the degree of folding of proteins.

2.4 Graph Measures for Portfolios

2.4.1 Average Centrality Measures

In the case of an investment portfolio, we can define an average centrality measure of a portfolio using the following formula:

$$\mathbf{CM}(x) = \mathbf{C}_n' x \tag{6}$$

where x is the column vector of weights of the portfolio of size $n \times 1$ and $\mathbf{C}_n$ is a centrality measure vector. This formula has the advantage that allows us to measure how diversify is a portfolio based on the graph relationships of its assets.

2.4.2 Percentage Invested in Connected Assets

In the case of an investment portfolio, we can define the percentage invested in connected assets of a portfolio using the following formula:

$$\mathbf{CA}(x) = \frac{\mathbf{1}_n \left(\mathbf{B}_{1,l} \odot |xx'| \right) \mathbf{1}_n'}{\mathbf{1}_n |xx'| \mathbf{1}_n'} \tag{7}$$

where x is the column vector of weights of the portfolio of size $n \times 1$, $\odot$ is the Hadamard product or element wise product, and $|.|$ is the element wise absolute value. This formula allows us to measure the percentage invested in assets that are connected directly or through walks specified through the connection matrix $\mathbf{B}_{1,l}$. The idea is that when we invest in assets that are not connected, the absolute value of the product of their weights $|x_i x_j| = 0$ because one of the assets must have a weight equal zero, this means that $x_i = 0$ or $x_j = 0$.

3 Graphs in Financial Markets

3.1 Minimum Spanning Tree

The minimum spanning tree (MST) is a spanning tree of an undirected connected weighted graph in which the sum of the weight of the edges is as minimum as possible. Mantegna (1999) was the first that suggest that the relationships among assets can be represented using a MST. To construct a MST from a codependence matrix we have to transform the codependence matrix into a distance matrix[3]. Then based on the distance matrix we can calculate the MST using one of the algorithms available[4].

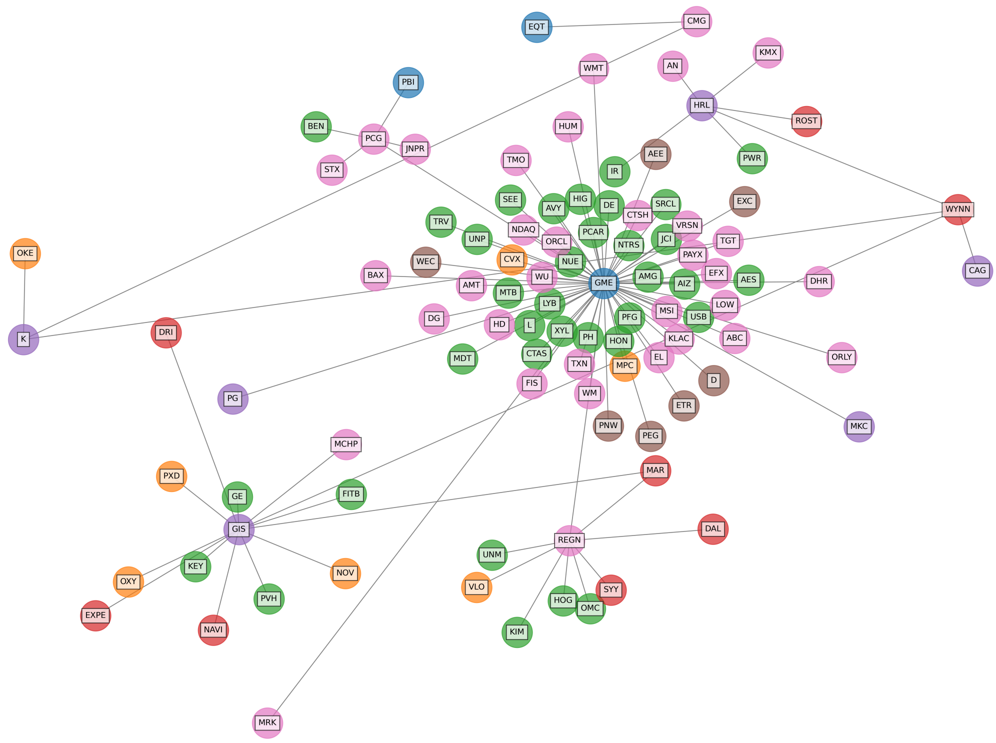

Figure 11: Minimum Spanning Tree of 100 Assets from S&P500

The MST provides useful information about the relationships among assets. Nodes in the border or periphery of the MST have only one edge (degree one) and this number increases for nodes that are nearest to the center of the MST (degree higher than one). The asset with lower degree are less related to others while assets with higher degree

[3] We can use the formulas $d = \sqrt{0.5(1-\rho)}$ or $d = \sqrt{1-|\rho|}$, where ρ is a codependence measure

[4] Networkx Python library has a function to calculate the MST from an adjacency matrix.

are assets that are more related to others. We can create a more diversified portfolio if more assets in the portfolio comes from the periphery of the MST.

3.2 Triangulated Maximally Filtered Graph

Triangulated Maximally Filtered Graph (TMFG) is a network-filtering method proposed by Massara et al. (2017) that provides an approximate solution to the Weighted Maximal Planar Graph problem. TMFG consists in building a triangulation that maximizes a score function associated with the amount of information retained by the network. To construct a TMFG from a codependence matrix we have to transform the codependence matrix into a distance matrix. Then, based on the codependence and distance matrix we can calculate the TMFG using the algorithm proposed by Massara et al. (2017)[5].

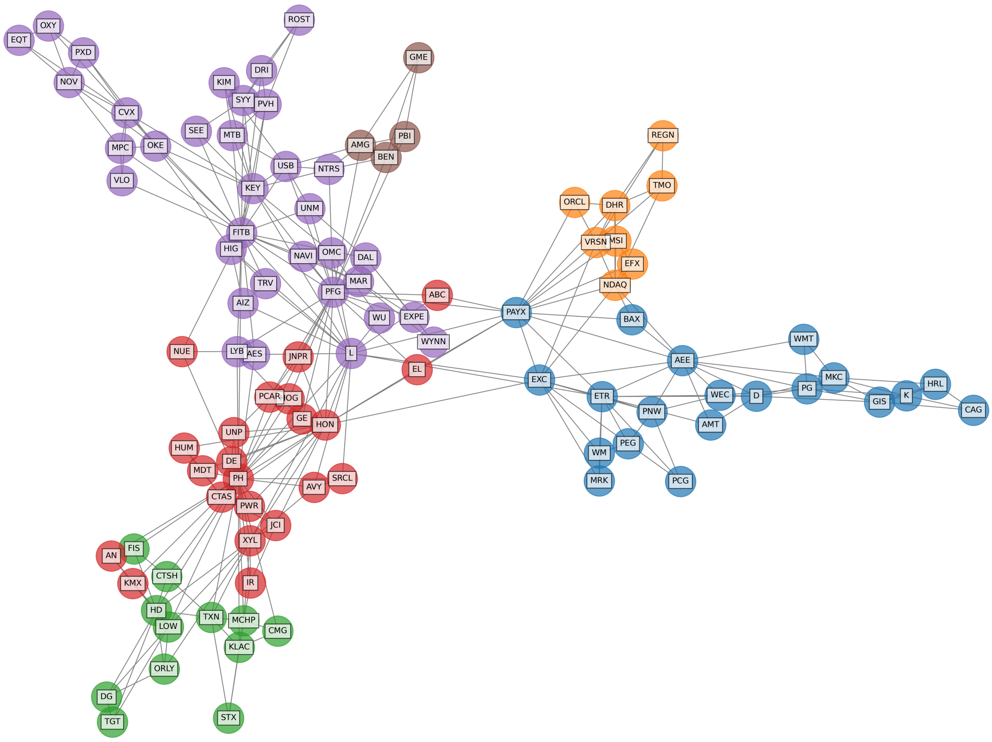

Figure 12: Triangulated Maximally Filtered Graph of 100 Assets from S&P500

The TMFG provides useful information about the relationships among assets be-

[5]Riskfolio-Lib Python library has a function to calculate the TMFG from a codependence and distance matrix.

cause it incorporate the information of the codependence and distance matrix. Nodes in the border or periphery of the TMFG have three edges (degree three) and this number increases for nodes that are nearest to the center of the TMFG (degree higher than three). The asset with lower degree are less related to others while assets with higher degree are assets that are more related to others. We can create a more diversified portfolio if more assets in the portfolio comes from the periphery of the TMFG as shown in Pozzi et al. (2013).

4 Centrality Measure Constraint

We can add information from a graph that allow us to diversify a portfolio considering the influence of assets in the graph, just adding a constraint on an average centrality measure of a portfolio. We can posed this constraint as follows:

$$\begin{aligned} \underset{x}{\text{opt}} \quad & \phi(x) \\ \text{s.t.} \quad & \mathbf{C}_n' x = \bar{c} \\ & x \in \mathcal{X} \end{aligned} \tag{8}$$

where $\phi(x)$ is an objective function, opt is max when $\phi(x)$ is concave and min when $\phi(x)$ is convex, x is the column vector of portfolio weights, $\mathbf{C}_n$ is the centrality measure vector, $\bar{c}$ is the desired average centrality measure of the portfolio and $\mathcal{X}$ is a convex set. To calculate the $\mathbf{C}_n$ we use the unweighted adjacency matrix of the MST or TMFG, this is the adjacency matrix that only considers the number of edges between two vertices.

The problem 8 is a general return risk trade-off optimization problem, this means that we can consider the classic objective functions based on convex risk measures like the maximization of return, minimization of risk, maximization of the risk adjusted return ratio or maximization of a risk averse utility function. Also, the convex set $\mathcal{X}$ is a general set that includes long and short position constraints, linear constraints, constraints on convex risk measures, tracking error constraints among others.

5 Neighborhood Constraint

5.1 Mixed Integer Programming Approach

Ricca and Scozzari (2024) proposed an integer constraint that allows us to invest in assets that are not neighbors, this means that are not directly linked through an edge. We can generalize this idea using the connection matrix $\mathbf{B}_{1,l}$ to assets that are not linked through walks of lengths that are lower or equal than l ($l \geq 1$) as follows:

$$\begin{aligned}
\underset{x}{\text{opt}} \quad & \phi(x) \\
\text{s.t.} \quad & (\mathbf{B}_{1,l} + I_n)y \leq 1 \\
& x_i \leq b_u y_i \ \forall \ i = 1, \ldots, n \\
& x_i \geq b_l y_i \ \forall \ i = 1, \ldots, n \\
& x \in \mathcal{X}
\end{aligned} \tag{9}$$

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 Ricca and Scozzari's model is the special case when $\mathbf{B}_{1,1} = A$.

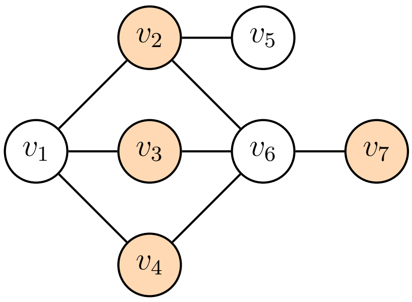

Figure 13: Example of Graph of Assets

To illustrate the idea of this model, if figure 13 is the graph that represent the relationships among 7 assets. If we do not want to invest in assets that are connected by walks of length one, this means $\mathbf{B}_{1,1} = A$; if we select asset v_1 we can not invest in assets $\{v_2, v_3, v_4\}$ and if we select asset v_6 we can not invest in asset v_7. The same idea can be extrapolated to assets that are connected by walks of lengths higher than one using the matrix $\mathbf{B}_{1,l}$.

The main advantages of this model is that can be applied to any convex risk measure and it can be combined easily with a cardinality constraint $\mathbf{1}_n' y \leq K$, where K is the maximum number of assets. The main disadvantages of this model is that relies in a mixed integer programming (MIP) formulation, making difficult to solve when we increase the number of assets; and for matrices $(\mathbf{B}_{1,l} + I_n)$ that are near to the all ones matrix, the problem can not be solved.

5.2 Semidefinite Programming Approach

In this section we proposed a semidefinite programming approach that allows us to invest in assets that are not linked through a walk of length $l \geq 1$. Based on the idea of the percentage invested in connected assets that if we want to invest a lower percentage in assets that are connected the product of their weights approximate to zero, we can pose the minimization of variance with a constraint to invest in assets that are not linked through a walk of length $l \geq 1$ as follows:

$$\begin{aligned} \min_{x,X} \quad & \mathrm{Tr}(\Sigma X) \\ \text{s.t.} \quad & \begin{bmatrix} X & x \\ x' & 1 \end{bmatrix} \succeq 0 \\ & X = X' \\ & \mathbf{B}_{1,l} \odot X = 0 \\ & x \in \mathcal{X} \end{aligned} \tag{10}$$

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 connected adding a penalty function $\lambda \mathrm{Tr}(X)$ in the objective function as follows:

$$\begin{aligned} \underset{x}{\text{opt}} \quad & \phi(x) + \lambda \mathrm{Tr}(X) \\ \text{s.t.} \quad & \begin{bmatrix} X & x \\ x' & 1 \end{bmatrix} \succeq 0 \\ & X = X' \\ & \mathbf{B}_{1,l} \odot X = 0 \\ & x \in \mathcal{X} \end{aligned} \tag{11}$$

where λ is a penalty factor. The advantage of this approach compared to the MIP approach is that gives us an approximate solution for matrices $\mathbf{B}_{1,l}$ that are near to the all ones matrix.

6 Numerical Examples

We select 100 assets (i.e., stocks ABC, AEE, AES, AIZ, AMG, AMT, AN, AVY, BAX, BEN, CAG, CMG, CTAS, CTSH, CVX, D, DAL, DE, DG, DHR, DRI, EFX, EL, EQT, ETR, EXC, EXPE, FIS, FITB, GE, GIS, GME, HD, HIG, HOG, HON, HRL, HUM, IR, JCI, JNPR, K, KEY, KIM, KLAC, KMX, L, LOW, LYB, MAR, MCHP, MDT, MKC, MPC, MRK, MSI, MTB, NAVI, NDAQ, NOV, NTRS, NUE, OKE, OMC, ORCL, ORLY, OXY, PAYX, PBI, PCAR, PCG, PEG, PFG, PG, PH, PNW, PVH, PWR, PXD, REGN, ROST, SEE, SRCL, STX, SYY, TGT, TMO, TRV, TXN, UNM, UNP, USB, VLO, VRSN, WEC, WM, WMT, WU, WYNN and XYL) 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[6] and MOSEK[7] solver.

6.1 Return Variance Optimization

If we calculate the efficient frontier using the variance as a risk measure and then group the weights of the assets by the node's degree of each asset in the MST and TMFG we get the following charts:

[6] Diamond and Boyd (2016) and Agrawal et al. (2018)
[7] ApS (2023)

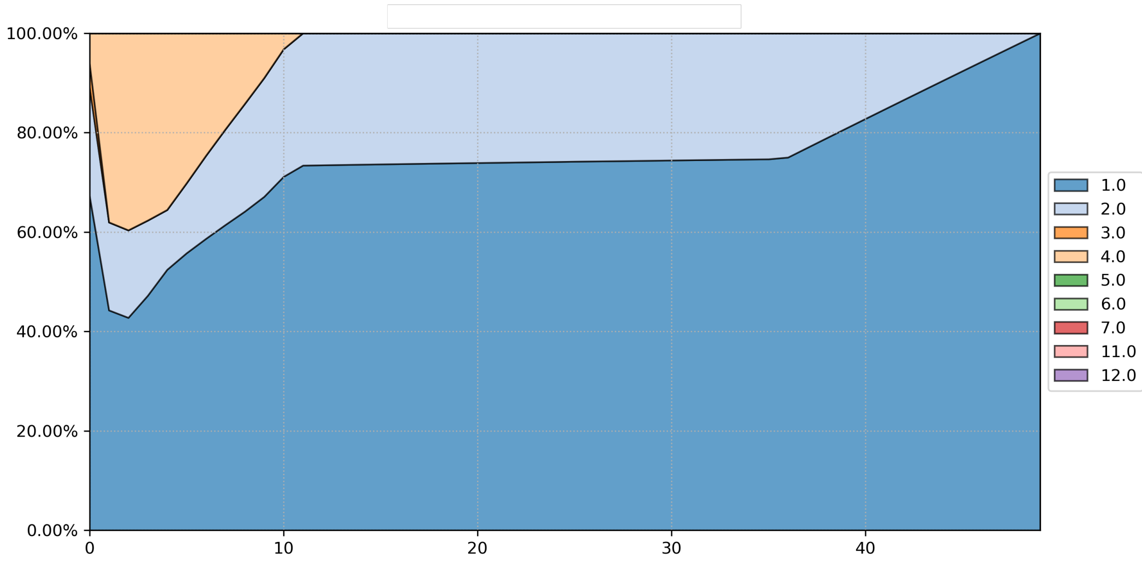

Figure 14: Efficient Frontier Composition per Node's Degree MST

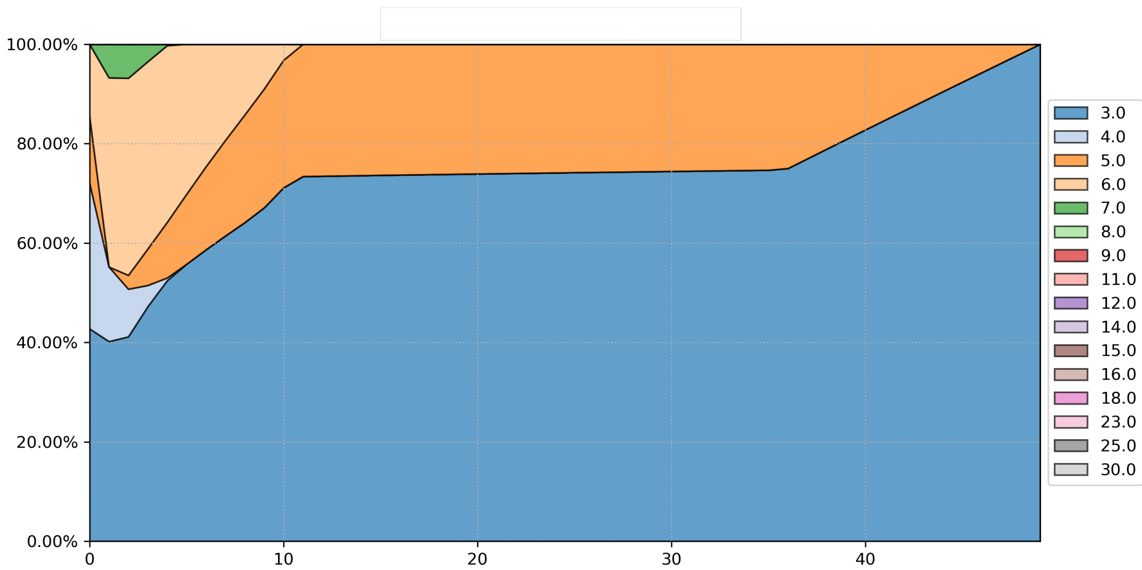

Figure 15: Efficient Frontier Composition per Node's Degree TMFG

In figure 11 and 12 we can see that for the two graphs, the portfolios in the variance efficient frontier that are more risk averse (left of charts) are less concentrated in assets that are in the periphery while when assets are less risk averse (right of charts) are more concentrated in assets in the periphery. If we solve problem 8 using the degree vector based on the MST with the constraints $\bar{c} = 1$ and $\bar{c} = 2$ for the average node's degree; the problems 9 and 10 for the adjacency matrix of the MST; and then we compare these four portfolios with the minimum variance portfolio, we get the following results:

Degree	Degree $\bar{c}=1$	Degree $\bar{c}=2$	MIP	SDP	Min Variance
1	100.00%	49.89%	74.84%	77.26%	67.29%
2	0.00%	23.46%	21.27%	18.43%	21.82%
3	0.00%	6.41%	3.88%	4.30%	4.82%
4	0.00%	17.25%	0.00%	0.00%	6.07%
5	0.00%	2.98%	0.00%	0.00%	0.00%
6	0.00%	0.00%	0.00%	0.00%	0.00%
7	0.00%	0.00%	0.00%	0.00%	0.00%
11	0.00%	0.00%	0.00%	0.00%	0.00%
12	0.00%	0.00%	0.00%	0.00%	0.00%
Std. Dev.	1.047 %	1.023%	1.040%	1.020%	1.014%
$\mathbf{D}_n'x$	1.00	2.00	1.29	1.27	1.50
$\mathbf{CA}(x)$	0.00%	10.10%	0.00%	0.00%	5.86%

Table 1: MST Node's Degree Composition per Portfolio

We can see in table 1 that the portfolio with degree $\bar{c}=1$ allow us to allocate all the budget in assets that are in periphery of the MST. Portfolios with node's degree $\bar{c}=1$, MIP and SDP has a percentage invested in connected assets of 0.00%, this means that these portfolios do not invest in assets that are neighbors in the MST. On the other hand, the minimum variance portfolio do not diversify well based on the graph structure of MST, because it has an average node's degree of 1.5 and a percentage invested in connected assets of 5.86%.

If we solve problem 8 using the degree vector based on the TMFG with the constraints $\bar{c}=3$ and $\bar{c}=6$ for the average node's degree; the problems 9 and 10 for the adjacency matrix of the TMFG; and then we compare these four portfolios with the minimum variance portfolio, we get the following results:

Degree	Degree $\bar{c} = 3$	Degree $\bar{c} = 6$	MIP	SDP	Min Variance
3	100.00%	33.87%	43.55%	47.50%	42.76%
4	0.00%	23.22%	49.58%	38.02%	29.35%
5	0.00%	11.74%	0.00%	4.38%	13.43%
6	0.00%	15.35%	6.87%	10.10%	14.46%
7	0.00%	1.89%	0.00%	0.00%	0.00%
8	0.00%	0.00%	0.00%	0.00%	0.00%
9	0.00%	0.00%	0.00%	0.00%	0.00%
11	0.00%	0.00%	0.00%	0.00%	0.00%
12	0.00%	0.00%	0.00%	0.00%	0.00%
14	0.00%	1.79%	0.00%	0.00%	0.00%
15	0.00%	0.00%	0.00%	0.00%	0.00%
16	0.00%	10.29%	0.00%	0.00%	0.00%
18	0.00%	0.00%	0.00%	0.00%	0.00%
23	0.00%	0.00%	0.00%	0.00%	0.00%
25	0.00%	0.00%	0.00%	0.00%	0.00%
30	0.00%	1.84%	0.00%	0.00%	0.00%
Std. Dev.	1.084%	1.041%	1.064%	1.031%	1.014%
$\mathbf{D}_n'x$	3.00	6.00	3.70	3.77	4.00
$\mathbf{CA}(x)$	0.00%	11.64%	0.00%	0.93%	9.21%

Table 2: TMFG Node's Degree Composition per Portfolio

We can see in table 2 that the constraint $\bar{c} = 3$ allow us to allocate all the budget in assets that are in periphery of the TMFG. Portfolios with node's degree $\bar{c} = 3$ and MIP have a percentage invested in connected assets of 0.00%, while the SDP portfolio has a value of 0.93%; this means that these portfolios do not invest in assets that are neighbors in the TMFG. On the other hand, the minimum variance portfolio do not diversify well based on the graph structure of TMFG, because it has an average node's degree of 4 and a percentage invested in connected assets of 9.21%.

6.2 Return RLVaR Optimization

If we calculate the efficient frontier using the relativistic value at risk (RLVaR) (Cajas, 2023) as a risk measure and then group the weights of the assets by the node's degree of each asset in the MST and TMFG we get the following charts:

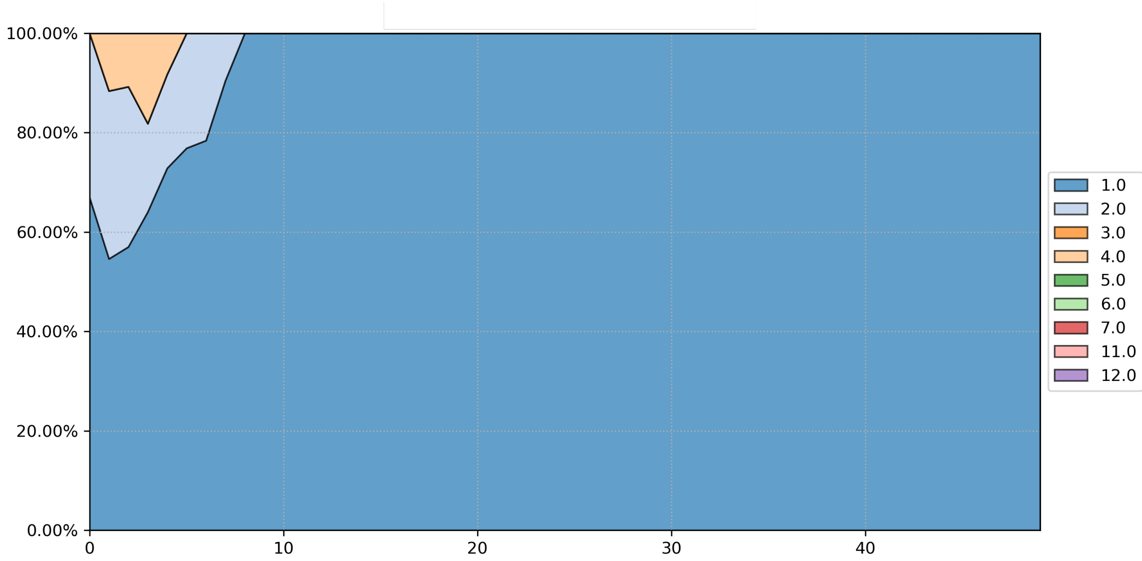

Figure 16: Efficient Frontier Composition per Node's Degree MST

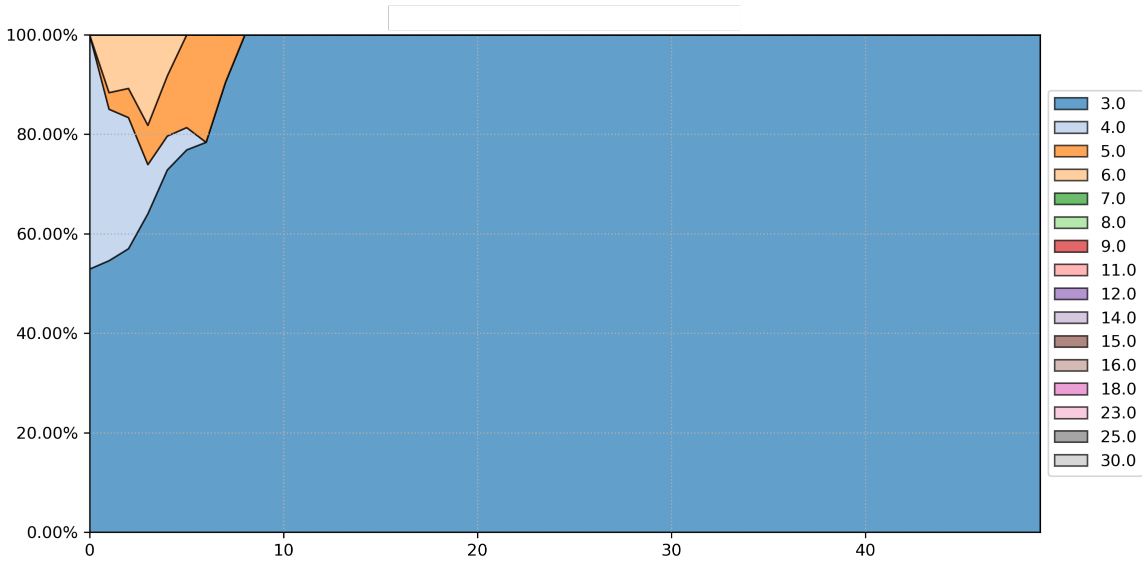

Figure 17: Efficient Frontier Composition per Node's Degree TMFG

In figure 13 and 14 we can see that for the two graphs, the portfolios in the RLVaR efficient frontier that are more risk averse (left of charts) are less concentrated in assets that are in the periphery while when assets are less risk averse (right of charts) are more concentrated in assets in the periphery. If we solve solve 8 using the degree vector based on the MST with the constraints $\bar{c} = 1$ and $\bar{c} = 2$ for the average node's degree; the problems 9 and 11 with $\lambda = 0.05$ for the adjacency matrix of the MST; and then we compare these four portfolios with the minimum variance portfolio, we get the following results:

Degree	Degree $\bar{c} = 1$	Degree $\bar{c} = 2$	MIP	SDP	Min RLVaR
1	100.00%	46.69%	66.37%	81.09%	66.96%
2	0.00%	29.96%	33.63%	18.31%	33.04%
3	0.00%	0.00%	0.00%	0.59%	0.00%
4	0.00%	23.35%	0.00%	0.00%	0.00%
5	0.00%	0.00%	0.00%	0.00%	0.00%
6	0.00%	0.00%	0.00%	0.00%	0.00%
7	0.00%	0.00%	0.00%	0.00%	0.00%
11	0.00%	0.00%	0.00%	0.00%	0.00%
12	0.00%	0.00%	0.00%	0.00%	0.00%
RLVaR	4.00%	3.91%	3.75%	3.87%	3.72%
$\mathbf{D}'_n x$	1.00	2.00	1.34%	1.19%	1.33
$\mathbf{CA}(x)$	0.00%	2.02%	0.00%	0.00%	0.00%

Table 3: MST Node's Degree Composition per Portfolio

We can see in table 3 that the constraint $\bar{c} = 1$ allow us to allocate all the budget in assets that are in periphery of the MST. Portfolios with node's degree $\bar{c} = 1$, MIP and SDP have a percentage invested in connected assets of 0.00%, this means that these portfolios do not invest in assets that are neighbors in the MST. The minimum RLVaR portfolio diversify well based on the graph structure of MST, because it has an average node's degree of 1.33 and a percentage invested in connected assets of 0.00%.

If we solve problem 8 using the degree vector based on the TMFG with the constraints $\bar{c} = 3$ and $\bar{c} = 6$ for the average node's degree; the problems 9 and 11 for the adjacency matrix of the TMFG; and then we compare these four portfolios with the minimum RLVaR portfolio, we get the following results:

Degree	Degree $\bar{c} = 3$	Degree $\bar{c} = 6$	MIP	SDP	Min RLVaR
3	100.00%	23.97%	65.88%	73.49%	52.93%
4	0.00%	61.24%	34.12%	17.76%	47.07%
5	0.00%	6.42%	0.00%	5.15%	0.00%
6	0.00%	0.00%	0.00%	3.60%	0.00%
7	0.00%	0.00%	0.00%	0.00%	0.00%
8	0.00%	0.00%	0.00%	0.00%	0.00%
9	0.00%	0.00%	0.00%	0.00%	0.00%
11	0.00%	0.00%	0.00%	0.00%	0.00%
12	0.00%	0.00%	0.00%	0.00%	0.00%
14	0.00%	0.00%	0.00%	0.00%	0.00%
15	0.00%	0.00%	0.00%	0.00%	0.00%
16	0.00%	0.00%	0.00%	0.00%	0.00%
18	0.00%	0.00%	0.00%	0.00%	0.00%
23	0.00%	0.00%	0.00%	0.00%	0.00%
25	0.00%	0.00%	0.00%	0.00%	0.00%
30	0.00%	8.37%	0.00%	0.00%	0.00%
RLVaR	4.029%	4.110%	3.880%	4.015%	3.722%
$\mathbf{D}_n' x$	3.00	6.00	3.34	3.38	3.47
$\mathbf{CA}(x)$	0.00%	14.37%	0.00%	0.13%	11.66%

Table 4: TMFG Node's Degree Composition per Portfolio

We can see in table 4 that the constraint $\bar{c} = 3$ allow us to allocate all the budget in assets that are in periphery of the TMFG. Portfolios with node's degree $\bar{c} = 3$, MIP and SDP have a percentage invested in connected assets of 0.00%; this means that these portfolios do not invest in assets that are neighbors in the TMFG. On the other hand, the minimum RLVaR portfolio do not diversify well based on the graph structure of TMFG, because it has an average node's degree of 3.47 and a percentage invested in connected assets of 11.66%.

6.3 Graph Clustering-Based Asset Allocation

If we calculate the three main graph clustering-based asset allocation models using the variance as a risk measure and then group the weights of the assets by the node's degree of each asset in the MST and TMFG, we get the following results:

Degree	HRP	HERC	NCO	Degree $\bar{c} = 1$	MIP	SDP
1	60.65%	57.08%	61.86%	100.00%	74.84%	77.26%
2	20.03%	22.50%	21.83%	0.00%	21.27%	18.43%
3	7.79%	9.61%	11.60%	0.00%	3.88%	4.30%
4	7.78%	7.91%	4.71%	0.00%	0.00%	0.00%
5	1.26%	0.57%	0.00%	0.00%	0.00%	0.00%
6	0.28%	0.14%	0.00%	0.00%	0.00%	0.00%
7	0.36%	0.17%	0.00%	0.00%	0.00%	0.00%
11	1.48%	1.84%	0.00%	0.00%	0.00%	0.00%
12	0.42%	0.19%	0.00%	0.00%	0.00%	0.00%
$\mathbf{D}'_n x$	1.87	1.90	1.59	1.00	1.29	1.27
$\mathbf{CA}(x)$	2.36%	3.06%	6.87%	0.00%	0.00%	0.00%

Table 5: MST Node's Degree Composition per Portfolio

Degree	HRP	HERC	NCO	Degree $\bar{c} = 3$	MIP	SDP
3	33.93%	36.40%	47.62%	100.00%	43.55%	47.50%
4	20.94%	20.97%	27.95%	0.00%	49.58%	38.02%
5	12.60%	8.28%	4.73%	0.00%	0.00%	4.38%
6	9.77%	6.98%	6.95%	0.00%	6.87%	10.10%
7	10.88%	15.06%	12.75%	0.00%	0.00%	0.00%
8	0.73%	0.21%	0.00%	0.00%	0.00%	0.00%
9	2.02%	2.92%	0.00%	0.00%	0.00%	0.00%
11	3.14%	3.37%	0.00%	0.00%	0.00%	0.00%
12	1.99%	2.79%	0.00%	0.00%	0.00%	0.00%
14	0.23%	0.13%	0.00%	0.00%	0.00%	0.00%
15	1.48%	1.84%	0.00%	0.00%	0.00%	0.00%
16	0.76%	0.33%	0.00%	0.00%	0.00%	0.00%
18	0.50%	0.23%	0.00%	0.00%	0.00%	0.00%
23	0.24%	0.13%	0.00%	0.00%	0.00%	0.00%
25	0.36%	0.17%	0.00%	0.00%	0.00%	0.00%
30	0.42%	0.19%	0.00%	0.00%	0.00%	0.00%
$\mathbf{D}'_n x$	5.40	5.32	4.09	3.00	3.70	3.77
$\mathbf{CA}(x)$	6.35%	7.82%	12.14%	0.00%	0.00%	0.93%

Table 6: TMFG Node's Degree Composition per Portfolio

We can see that for the MST and TMFG, the HRP and HERC allocate weights along all node's degree with an average degree of 5.4 and 5.3 respectively. In the case of the NCO, it has a similar asset allocation than the minimum variance portfolio with an average degree of 4.09. Also, the three graph clustering-based asset allocation models invest a higher proportion of weights in assets that are neighbors in the MST and TMFG. On the other hand, the three portfolios that diversify based on the graph structure have lower values for the average node's degree and percentage invested in connected assets.

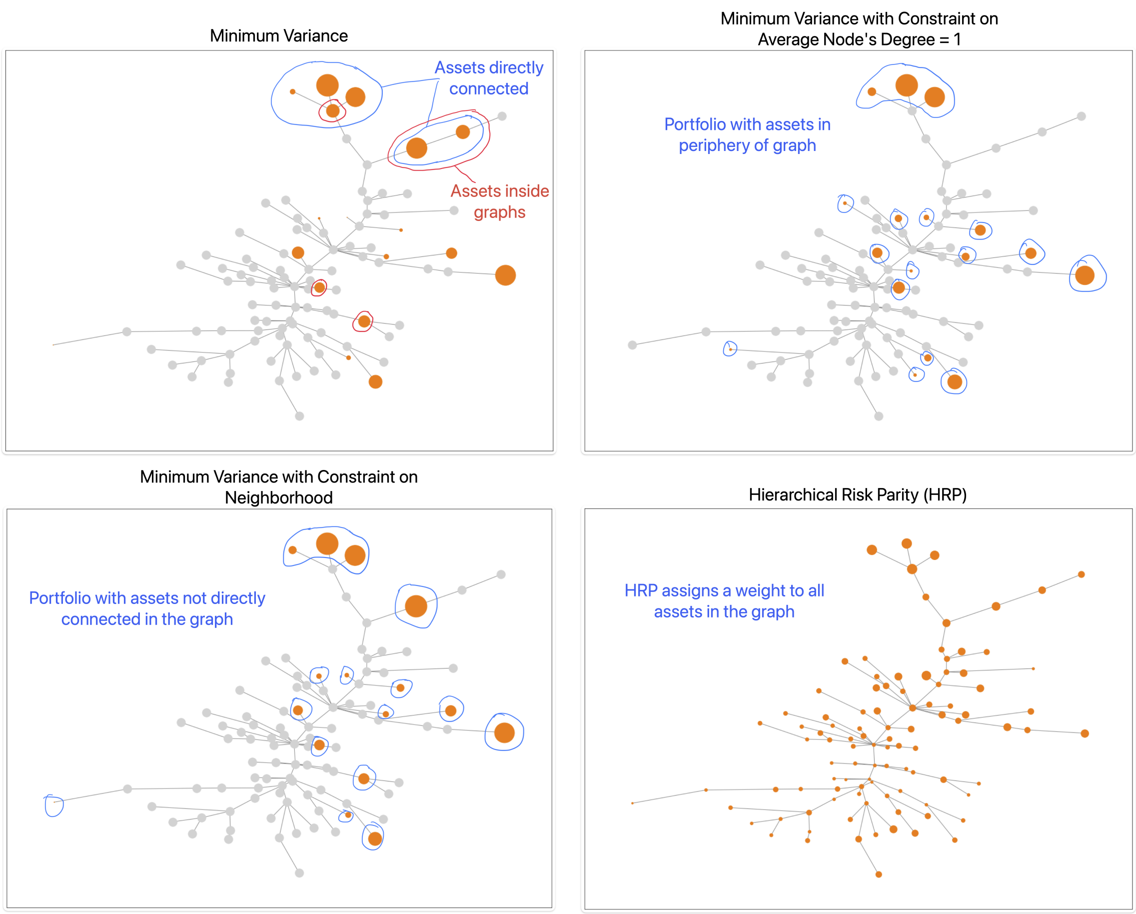

Figure 18: Comparison of Portfolio Models in the MST

7 Conclusions

This work proposes two approaches that allow us to incorporate information from a graph structure into classic return risk trade-off portfolio optimization models. Taking advantage of the matrix representation of graphs like the MST and TMFG through an adjacency matrix, we can add constraints that allow us to diversify the portfolio based on constraints in the average centrality measure of a portfolio or constraints in the neighborhood of assets. We show in some examples that classic convex portfolio optimization problems do not incorporate information from graphs in the diversification process. Also, the most modern graph clustering-based asset allocation models like HRP do not diversify considering the centrality of assets in graphs, or the connections that exist between assets and their neighborhood; because these models assign weights to all assets. Finally, we show that the incorporation of a constraints on the average node's degree in the convex portfolio optimization problems allows us to select assets in the periphery of graph; and adding a constraint on the neighborhood of assets allows us to select assets that are not directly connected in the graph.

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, 3). Portfolio optimization of relativistic value at risk. *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.

Estrada, E. (2011, 10). 1217 Centrality measures. In *The Structure of Complex Networks: Theory and Applications*. Oxford University Press.

Li, Y., X.-F. Jiang, Y. Tian, S.-P. Li, and B. Zheng (2019, February). Portfolio optimization based on network topology. *Physica A: Statistical Mechanics and its Applications 515*, 671–681.

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.

Massara, G. P., T. D. Matteo, and T. Aste (2017). Network filtering for big data: Triangulated maximally filtered graph. *J. Complex Networks 5*, 161–178.

Peralta, G. and A. Zareei (2016, September). A network approach to portfolio selection. *Journal of Empirical Finance 38*, 157–180.

Pozzi, F., T. D. Matteo, and T. Aste (2013, April). Spread of risk across financial markets: better to invest in the peripheries. *Scientific Reports 3*(1).

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

Puerto, J., M. Rodríguez-Madrena, and A. Scozzari (2020, May). Clustering and portfolio selection problems: A unified framework. *Computers & Operations Research 117*, 104891.

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.